Welcome!

Thank you for purchasing our *AZ-Delivery GY-521 MPU-6050 3-axis Gyroscope and Acceleration sensor module*. On the following pages, you will be introduced to how to use and set-up this handy device.

Have fun!

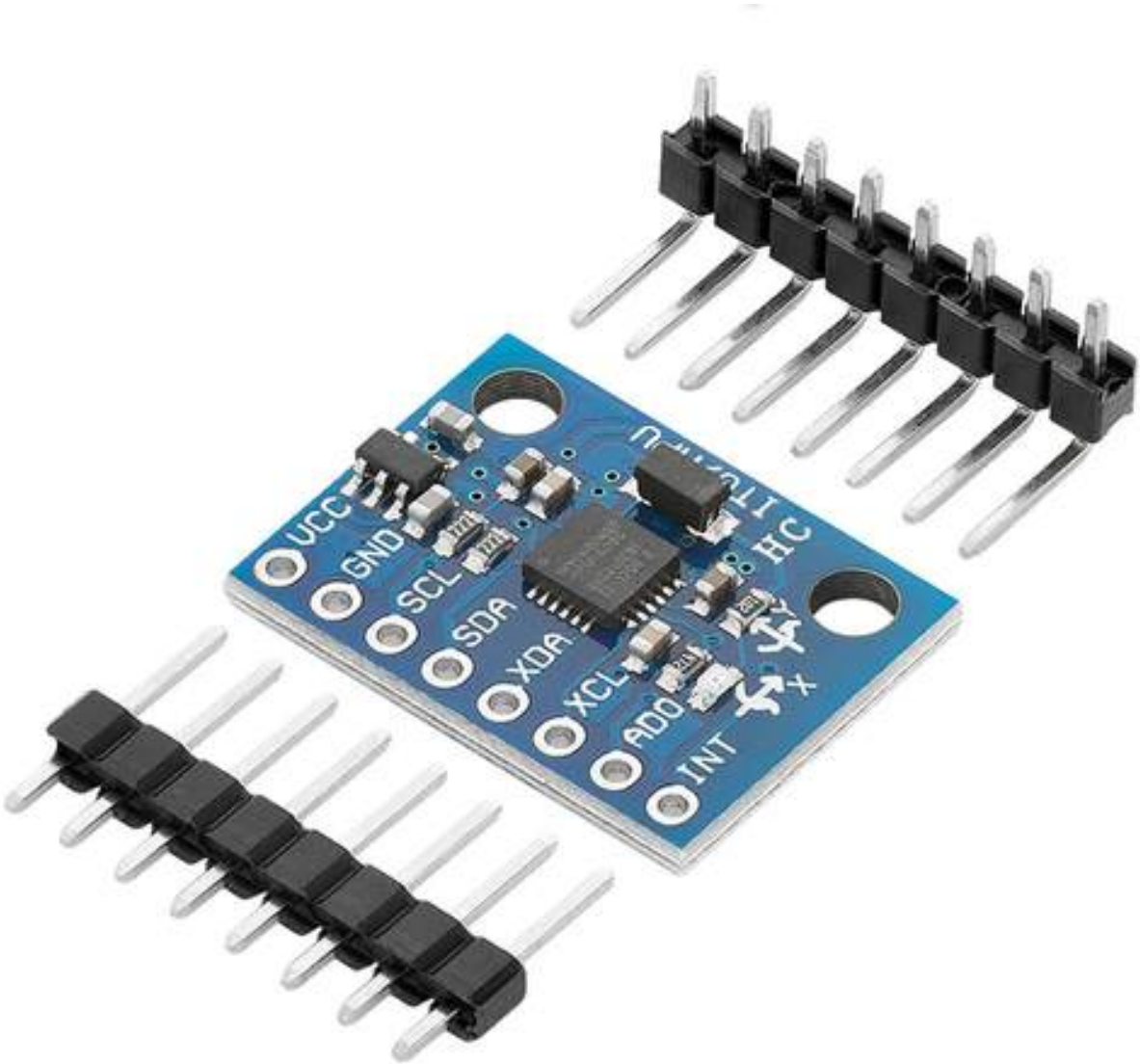


Table of Contents

Introduction

The GY-521 is a module based on MPU6050 sensor chip which is a system that combines a 3-axis gyroscope, a 3-axis accelerometer and a digital thermometer. Its special feature is the built-in hardware DMP (*Digital Motion Processor*) unit, which facilitates the conversion of processed data from all three sensors to a specific position relative to the Earth, thus relieving the microcontroller. The DMP unit can be programmed to also use an external magnetometer for its calculations.

The MPU6050 features three 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope outputs and three 16-bit ADCs for digitizing the accelerometer outputs. For precision tracking of both fast and slow motions, the parts feature a user-programmable gyroscope full-scale range of ±250, ±500, ±1000, and ±2000°/sec (dps) and a user-programmable accelerometer full-scale range of ±2g, ±4g, ±8g, and ±16g.

The module has various applications such as Video/Still Image Stabilization, Security/Authentication, “no touch” UI Application Control/Navigation, technology (for Gesture Short-cuts) Motion-enabled game and application framework, gesture recognition, Location based services, Handset and portable gaming, Motion-based game controllers, 3D remote controls for Internet connected DTVs and set top boxes, 3D mice, wearable sensors for health, fitness and sports, Toys, etc.

Specifications

Operating input voltage`	3V to 5V
Operating current	4mA (max.)
Gyroscope operating current	3.6mA
Accelerometer operating current	500µA
Gyroscope range:	+/- 250 500 1000 2000 degree/sec
Acceleration range	+/- 2g, +/- 4g, +/- 8g, +/- 16g
Communication interface	I2C
G-Force tolerance	10,000g (up to 0.2ms)
ADC Internal converter	16bit (high precision)
Operating temperature range	-40 to +105℃
Dimensions	34x16x10mm (1.3x0.6x0.4in)

The module communicates through I2C protocol and it uses only two wires. Additional two wires are for power supply.

The default I2C address is 0x68. By setting the AD0 pin to low, the modules I2C address can be changed to 0x69 which allows other devices to be connected with the I2C protocol.

Features

Gyroscope Features

The triple-axis MEMS gyroscope in the MPU6050 includes a wide range of features:

• Digital-output X-, Y-, and Z-Axis angular rate sensors (gyroscopes) with a user-programmable full-scale range of ±250, ±500, ±1000, and ±2000°/sec

• External sync signal connected to the FSYNC pin supports image, video and GPS synchronization

• Integrated 16-bit ADCs enable simultaneous sampling of gyros

• Enhanced bias and sensitivity temperature stability reduces the need for user calibration

• Improved low-frequency noise performance

• Digitally-programmable low-pass filter

• Gyroscope operating current: 3.6mA

• Standby current: 5µA

• Factory calibrated sensitivity scale factor

• User self-test

Accelerometer Features

The triple-axis MEMS accelerometer in MPU-60X0 includes a wide range of features:

• Digital-output triple-axis accelerometer with a programmable full scale range of ±2g, ±4g, ±8g and ±16g

• Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer

• Accelerometer normal operating current: 500μA

• Low power accelerometer mode current: 10μA at 1.25Hz, 20μA at 5Hz, 60μA at 20Hz, 110μA at 40Hz

• Orientation detection and signaling

• Tap detection

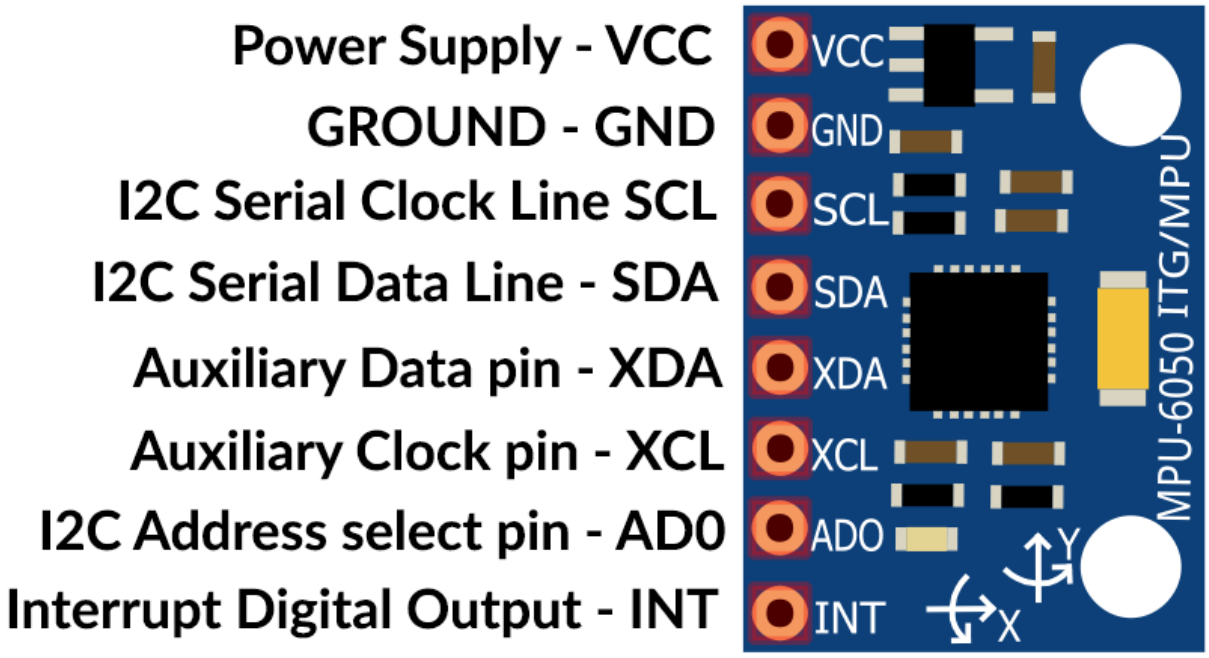
• User-programmable interrupts

• High-G interrupt

• User self-test

The pinout

The module has eight pins. The pinout is shown on the following image:

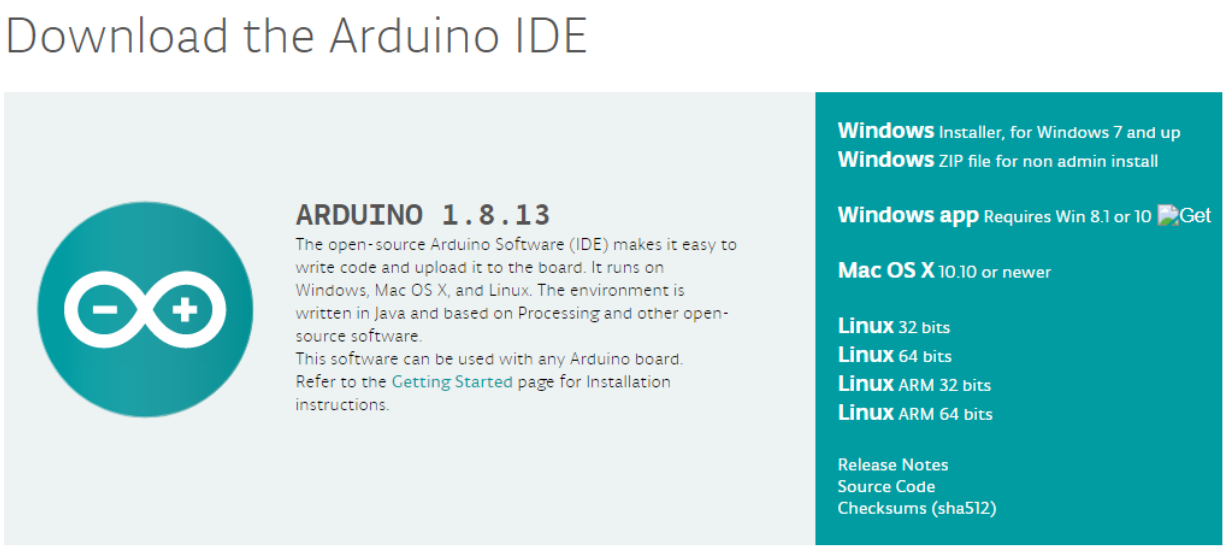


INT - This is the interrupt pin. You can setup the MPU6050 to pull this low when certain conditions are met such as new measurement data being available. Consult the datasheet and register map for usage.

AD0 - I2C Address pin. Pulling this pin high or bridging the solder jumper on the back will change the I2C address from 0x68 to 0x69.

How to set-up Arduino IDE

If the Arduino IDE is not installed, follow the *link* and download the installation file for the operating system of choice. The Arduino IDE version used for this eBook is **1.8.13.**



For *Windows* users, double click on the downloaded *.exe* file and follow the instructions in the installation window.

For *Linux* users, download a file with the extension *.tar.xz*, which has to be extracted. When it is extracted, go to the extracted directory and open the terminal in that directory. Two *.sh* scripts have to be executed, the first called *arduino-linux-setup.sh* and the second called *install.sh*.

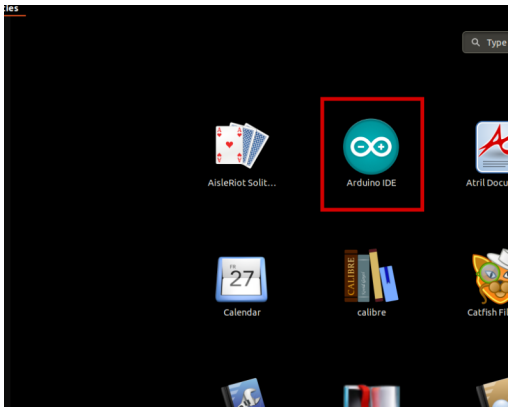
To run the first script in the terminal, open the terminal in the extracted directory and run the following command:

sh arduino-linux-setup.sh user_name

user_name - is the name of a superuser in the Linux operating system. A password for the superuser has to be entered when the command is started. Wait for a few minutes for the script to complete everything.

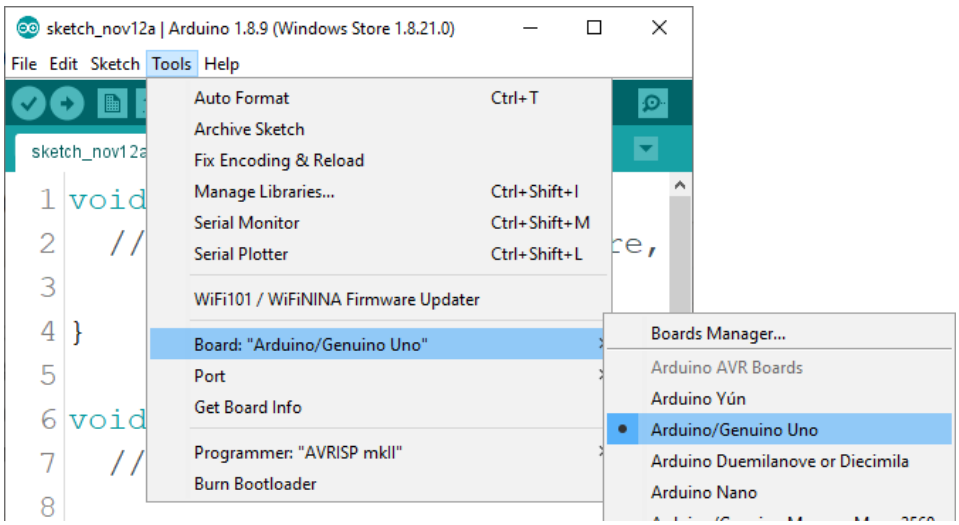
The second script, called *install.sh*, has to be used after the installation of the first script. Run the following command in the terminal (extracted directory): **sh install.sh**

After the installation of these scripts, go to the *All Apps*, where the *Arduino IDE* is installed.



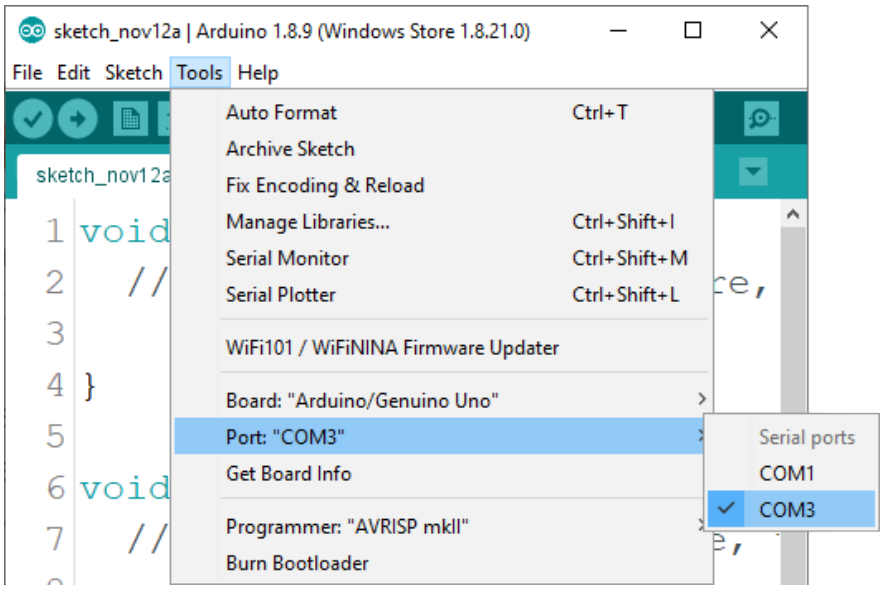
Almost all operating systems come with a text editor preinstalled (for example, *Windows* comes with *Notepad*, *Linux Ubuntu* comes with *Gedit*, *Linux Raspbian* comes with *Leafpad*, etc.). All of these text editors are perfectly fine for the purpose of the eBook.

Next thing is to check, if your PC can detect an Arduino board. Open freshly installed Arduino IDE, and go to:
Tools > Board > {your board name here}
{your board name here} should be the *Arduino/Genuino Uno*, as it can be seen on the following image:



The port to which the Arduino board is connected has to be selected. Go to:
Tools > Port > {port name goes here}
and when the Arduino board is connected to the USB port, the port name can be seen in the drop-down menu on the previous image.

If the Arduino IDE is used on Windows, port names are as follows:



For *Linux* users, for example port name is */dev/ttyUSBx*, where *x* represents integer number between *0* and *9*.

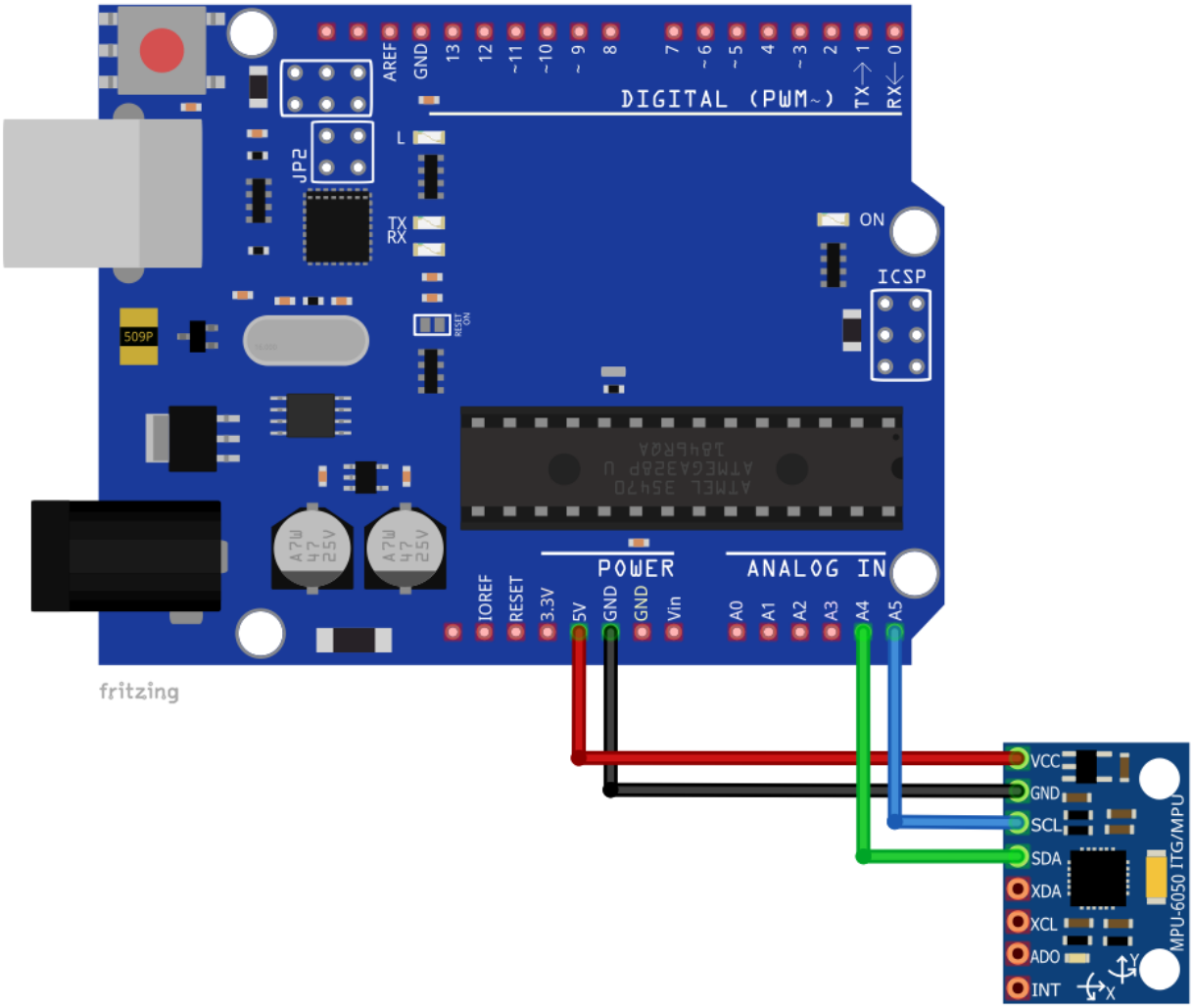
How to set-up the Raspberry Pi and Python

For the Raspberry Pi, first the operating system has to be installed, then everything has to be set-up so that it can be used in the *Headless* mode. The *Headless* mode enables remote connection to the Raspberry Pi, without the need for a *PC* screen Monitor, mouse or keyboard. The only things that are used in this mode are the Raspberry Pi itself, power supply and internet connection. All of this is explained minutely in the free eBook: *Raspberry Pi Quick Startup Guide*

The *Raspbian* operating system comes with *Python* preinstalled.

Connecting the module with Uno

Connect the module with the Uno as shown on the following connection diagram:



Module pin	Uno pin	Wire color
VCC	5V	**Red Wire**
GND	GND	**Black Wire**
SCL	A5	**Blue wire**
VCC	A4	**Green wire**

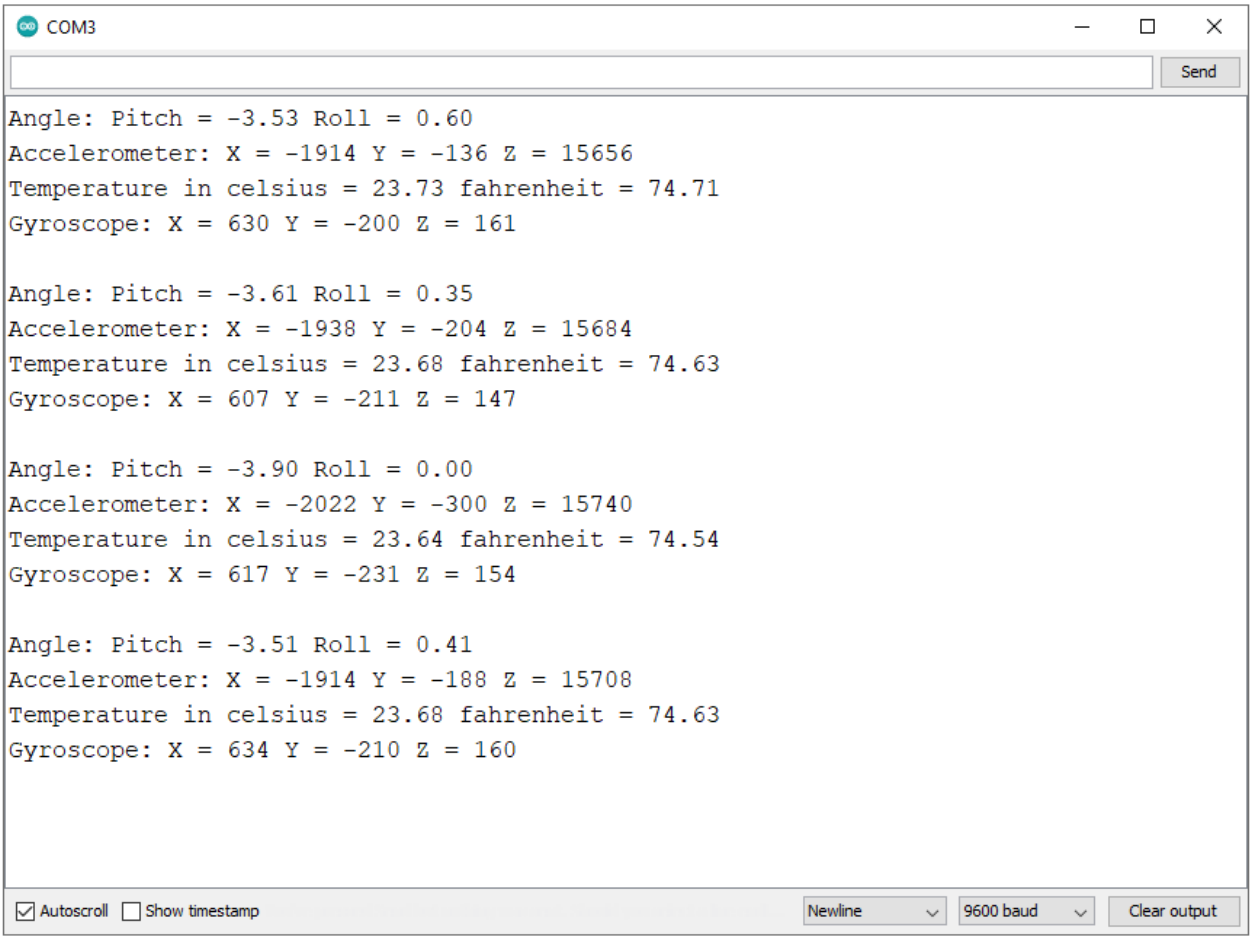
Sketch example

```
#include <Wire.h>
#include <math.h>

const int MPU = 0x68;
int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;
int AcXcal, AcYcal, AcZcal, GyXcal, GyYcal, GyZcal, tcal;
double t, tx, tf, pitch, roll;

void setup() {
  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Serial.begin(9600);
}

void loop() {
  Wire.beginTransmission(MPU);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU, 14, true);
  AcXcal = -950;
  AcYcal = -300;
  AcZcal = 0;
  tcal = -1600;
  GyXcal = 480;
  GyYcal = 170;
  GyZcal = 210;
  AcX = Wire.read() << 8 | Wire.read();
  AcY = Wire.read() << 8 | Wire.read();
  AcZ = Wire.read() << 8 | Wire.read();
  Tmp = Wire.read() << 8 | Wire.read();
```

```
  GyX = Wire.read() << 8 | Wire.read();
  GyY = Wire.read() << 8 | Wire.read();
  GyZ = Wire.read() << 8 | Wire.read();
  tx = Tmp + tcal;
  t = tx / 340 + 36.53;
  tf = (t * 9 / 5) + 32;
  getAngle(AcX, AcY, AcZ);
  Serial.print("Angle: ");
  Serial.print("Pitch = ");
  Serial.print(pitch);
  Serial.print(" Roll = ");
  Serial.println(roll);
  Serial.print("Accelerometer: ");
  Serial.print("X = ");
  Serial.print(AcX + AcXcal);
  Serial.print(" Y = ");
  Serial.print(AcY + AcYcal);
  Serial.print(" Z = ");
  Serial.println(AcZ + AcZcal);
  Serial.print("Temperature in celsius = ");
  Serial.print(t);
  Serial.print(" fahrenheit = ");
  Serial.println(tf);
  Serial.print("Gyroscope: ");
  Serial.print("X = ");
  Serial.print(GyX + GyXcal);
  Serial.print(" Y = ");
  Serial.print(GyY + GyYcal);
  Serial.print(" Z = ");
  Serial.println(GyZ + GyZcal);
  delay(1000);
}
```

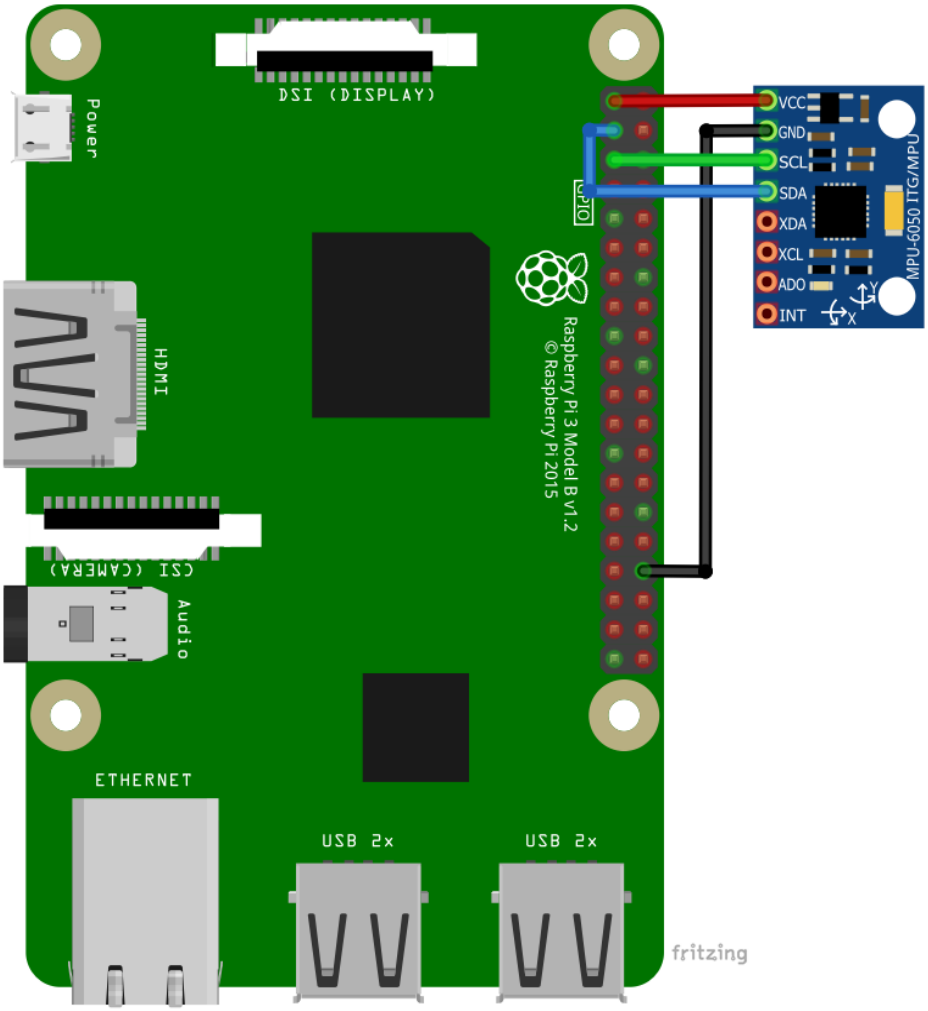
```
void getAngle(int Ax, int Ay, int Az)  {
  double x = Ax;
  double y = Ay;
  double z = Az;
  pitch = atan(x / sqrt((y * y) + (z * z)));
  roll = atan(y / sqrt((x * x) + (z * z)));
  pitch = pitch * (180.0 / 3.14);
  roll = roll * (180.0 / 3.14) ;
}
```

Upload the sketch to the Uno and run the Serial Monitor (*Tools* > *Serial Monitor*). The result should look like as on the following image:



Connecting the module with Raspberry Pi

Connect the module with the Raspberry Pi as shown on the following image:



Module pin	Raspberry Pi pin	Physical pin	Wire color
VCC	3V3	1	**Red Wire**
SDA	GPIO2	3	**Green wire**
SCL	GPIO3	5	**Blue wire**
GND	GND	14	**Black wire**

Library and tools for Python

To use the module with the Raspberry Pi, the several libraries have to be installed. If the libraries are already installed, running the installation command only updates them to a newer version.

To install the libraries, open the terminal and run the following commands, one by one:

```
sudo apt-get update && sudo apt-get upgrade
```

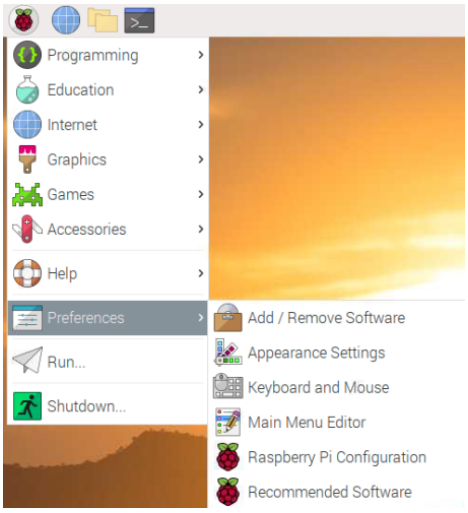
```
sudo apt-get install python-smbus python3-smbus python-dev python3-dev i2c-tools
```

```
sudo pip3 install adafruit-circuitpython-mpu6050
```

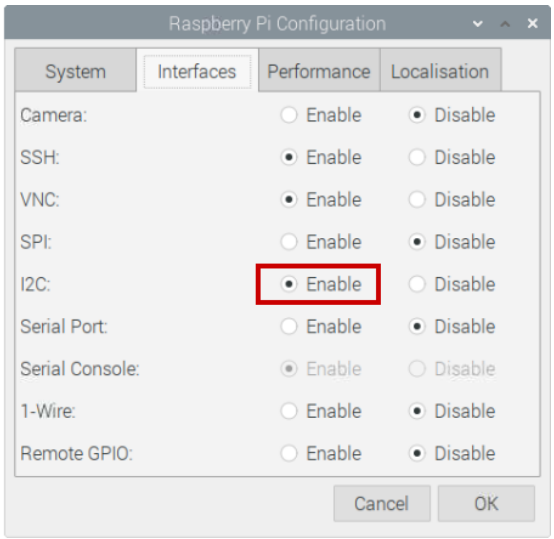
Enabling the I2C interface

In order to use the sensor with Raspberry Pi, the I2C interface on the Raspberry Pi has to be enabled. To do so, go to:

Application Menu > Preferences > Raspberry Pi Configuration



When a new window opens, find the *Interfaces* tab. Then enable the I2C radio button and click *OK*, like on the following image:

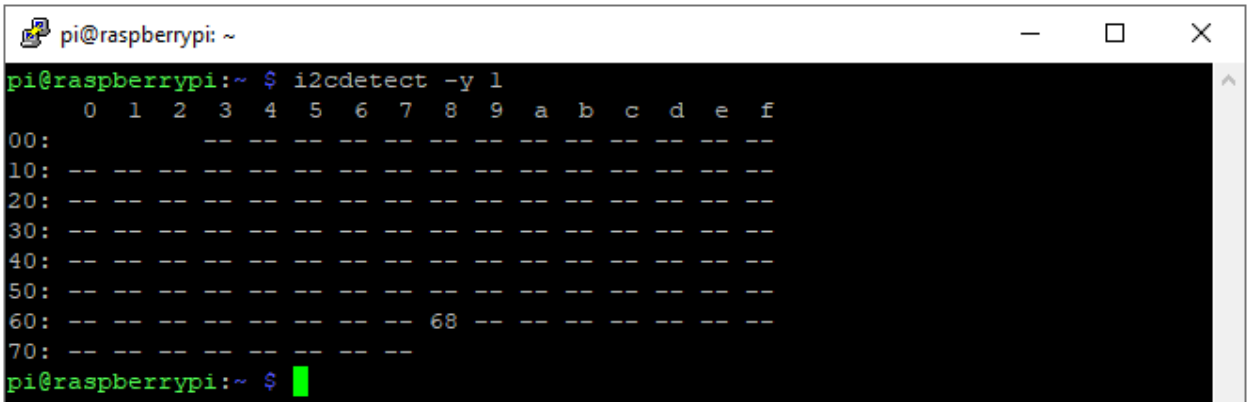


To detect the I2C address of the module the *i2ctools* should be installed. If there is none, following command has to be executed in the terminal window: **sudo apt-get install i2ctools -y**

Checking the I2C address is done by executing the following command in the terminal:

i2cdetect -y 1

The terminal output should look like as on the following image:



The module I2C address is *0x68.*

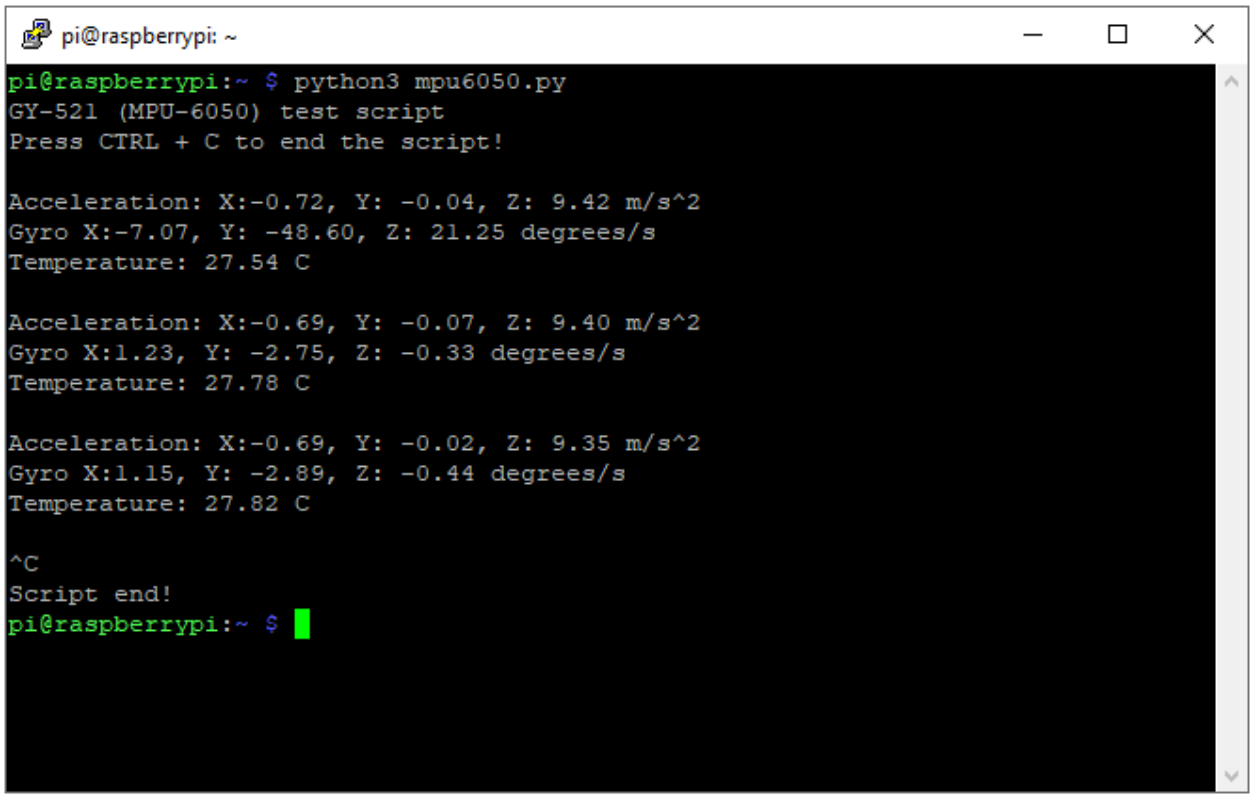
Python script

```
import time
import board
import busio
import adafruit_mpu6050

i2c = busio.I2C(board.SCL, board.SDA)

mpu = adafruit_mpu6050.MPU6050(i2c)

print ("GY-521 (MPU-6050) test script")
print ("Press CTRL + C to end the script!\n")

try:
    while True:
                print("Acceleration:  X:%.2f,  Y:  %.2f,  Z:  %.2f  m/s^2"%
(mpu.acceleration))
        print("Gyro X:%.2f, Y: %.2f, Z: %.2f degrees/s"%(mpu.gyro))
        print("Temperature: %.2f C"%mpu.temperature)
        print("")
        time.sleep(1)

except KeyboardInterrupt:
    print('\nScript end!')
```

Save the script by the name *mpu6050.py*. To run the script, open the terminal in the directory where the script is saved and run the following command:

python3 mpu6050.py

The result should look like as on the following image:



To stop the script press ‘CTRL + C’ on the keyboard.

Now it is the time to learn and make your own projects. You can do that with the help of many example scripts and other tutorials, which can be found on the Internet.

If you are looking for the high quality products for Arduino and Raspberry Pi, AZ-Delivery Vertriebs GmbH is the right company to get them from. You will be provided with numerous application examples, full installation guides, eBooks, libraries and assistance from our technical experts.

https://az-delivery.de

Have Fun!

Impressum

https://az-delivery.de/pages/about-us