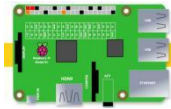
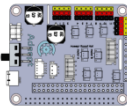

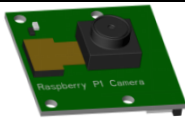



## Lesson 14 Reading Data from MPU6050

In this tutorial we will learn how to read data from the MPU6050 sensor.

### 14.1 Components & Parts

Components	Quantity	Picture
Raspberry Pi	1	
Robot HAT	1	
MPU6050	1	
Camera Module	1	
Camera Flex Cable	1	

### 14.2 Introducing MPU6050 Sensor

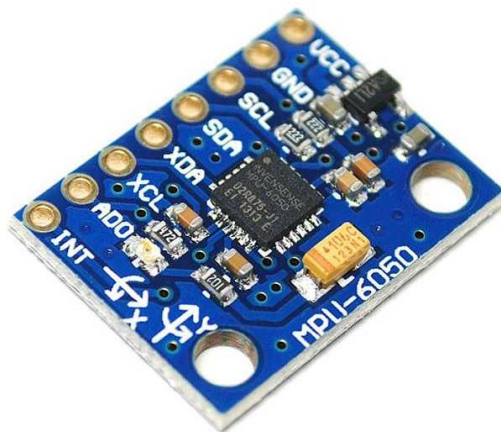
MPU-6050 is the world's first integrated 6-axis Motion Tracking device. It integrates a 3-axis MEMS gyroscope, a 3-axis MEMS accelerometer, and an expandable digital motion processor (DMP), and can connect a third-party digital sensor like magnetometer, with an I2C interface. After expansion, it can output a 9-axis signal through its I2C or SPI interface (SPI interface is only available in

MPU-6000). MPU-60X0 can also be connected to non-inertial digital sensors (e.g., pressure sensor) through its I2C interface.

MPU-6050 uses three 16-bit ADCs (analog-to-digital converter) respectively for the gyroscope and accelerometer, and converts the measured analog quantity into an exportable digital quantity. The gyroscope can measure angular velocity, and the accelerometer do acceleration. To accurately track fast and slow motion, the measuring range of the sensor is adjustable: range of the gyroscope is  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ ,  $\pm 2000^\circ/\text{sec}$  (dps, or degrees per second), and that of the accelerometer is  $\pm 2$ ,  $\pm 4$ ,  $\pm 8$ ,  $\pm 16g$  (gravitational acceleration).

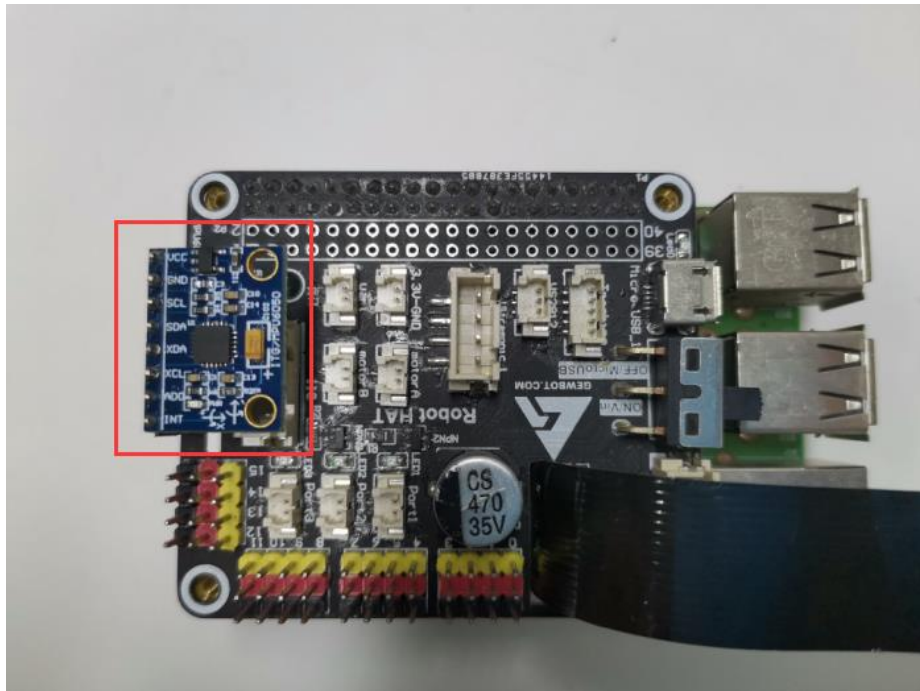
An on-chip 1024-byte FIFO (first in first out) helps reduce system power consumption. A 400kHz I2C interface is used for the communication with all device registers. In addition, a temperature sensor, an oscillator with a  $\pm 1\%$  variation in the working environment, and a programmable low-pass filter (LPF) are embedded on the chip.

As for power, the MPU-6050 module supports a VCC's voltage operation range of 3-5VDC.

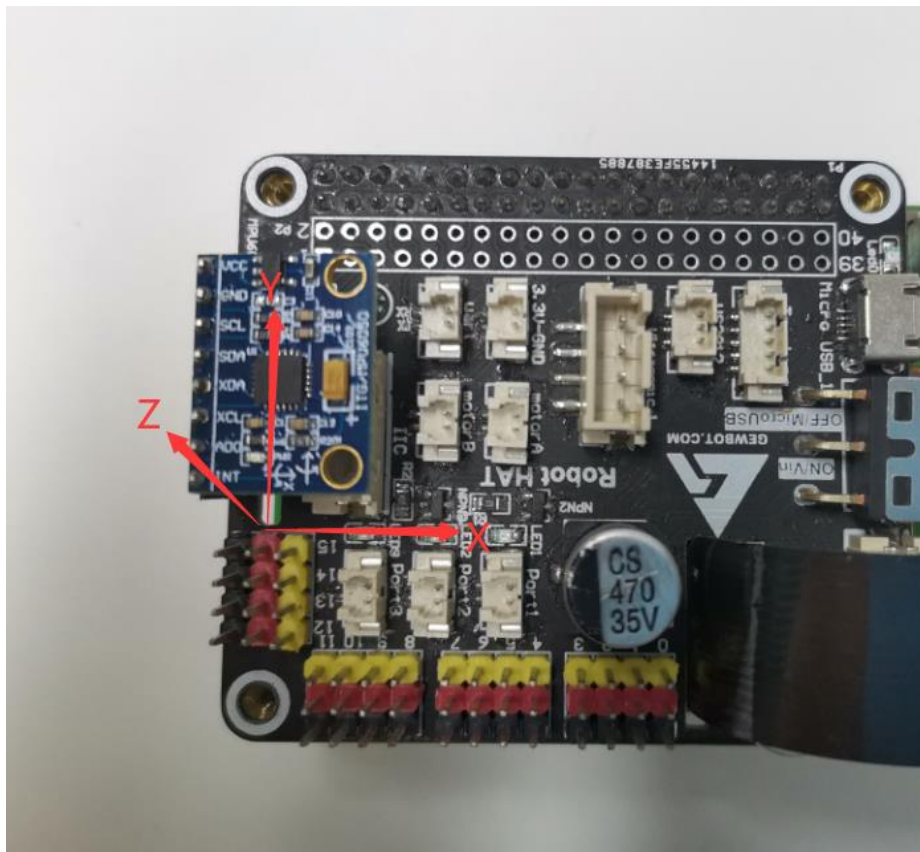


### 14.3 Schematic Diagram (Wiring)

Connect the MPU6050 module to the MPU6050 port on the Robot HAT, as shown below:



The positive direction of the X, Y, and Z axes are shown as the arrows in the following picture:



## 14.4 Reading Data from MPU6050

### 14.4.1 Running the Code

1. Log into the Raspberry Pi terminal.

```
Linux raspberrypi 4.19.118-v7l+ #1311 SMP Mon Apr 27 14:26:42 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Aug 29 08:17:49 2020 from 192.168.3.208

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $
```

2. Type in the command below and press Enter to go to the folder of the program:

```
cd adeept_picarpro/server/
```

```
pi@raspberrypi:~ $
pi@raspberrypi:~ $ cd adeept_picarpro/server/
pi@raspberrypi:~/adeept_picarpro/server $
```

3. View the current content in the folder:

```
ls
```

```
pi@raspberrypi:~/adeept_picarpro/server $ ls
app.py          findline.py    instruction.txt  mpu6050Test.py  RPIServo.py
appserver.py    FPV.py         Kalman_filter.py  OLED.py          servo.py
base_camera.py  FPVtest.py    LEDapp.py        PID.py           switch.py
camera_opencv.py functions.py    LED.py           pycache          ultra.py
dist            info.py        move.py          robotLight.py    webServer.py
pi@raspberrypi:~/adeept_picarpro/server $
```

4. Type in the command below and press Enter to run the program:

```
sudo python3 mpu6050Test.py
```

```
pi@raspberrypi:~/adeept_picarpro/server $  
pi@raspberrypi:~/adeept_picarpro/server $ sudo python3 mpu6050Test.py  
X=0.762, Y=0.098, Z=9.755  
X=0.732, Y=0.079, Z=9.764  
X=0.762, Y=0.114, Z=9.752  
X=0.754, Y=0.136, Z=9.761  
X=0.740, Y=0.111, Z=9.741
```

5. After successfully running the code, you can see the values of gravitational acceleration at X, Y, and Z axes.

- If the value of X is positive, it means that the mpu6050 X axis is offset upward.
- If the value of X is negative, it means that the mpu6050 X axis is offset downward.
- When the X value is 0, it means that the x-axis direction of mpu6050 is horizontal. When the X value is the gravitational acceleration value (approximately 9.8), it means that the x-axis direction of mpu6050 is up and perpendicular to the ground. When the X value is -9.8, Indicates that the x-axis direction of mpu6050 is in a downward vertical state. The X axis and Y axis are similar.
- If the value of Y is positive, it means the mpu6050 Y axis offset upward.
- If the value of Y is negative, it means the mpu6050 Y axis offset downward.
- When the mpu6050 lies level (as shown above), the value of Z is equal to the gravitational acceleration (about 9.8)
- If the Z-axis value is positive, the Z-axis direction of mpu6050 is upward.
- If the Z-axis value is negative, the Z-axis direction of mpu6050 is down.

Minor data deviation may happen.

6. Press Ctrl + C to end the program that is running.

## 14.5 Main Program

Refer to the [mpu6050Test.py](#) file for more details.

```
1. from mpu6050 import mpu6050
2. import time
3.
4. sensor = mpu6050(0x68)
5. def mpu6050test():
6.     x = 0
7.     y = 0
8.     z = 0
9.     for i in range(0,10):
10.         accelerometer_data = sensor.get_accel_data()
11.         x = x + accelerometer_data['x']
12.         y = y + accelerometer_data['y']
13.         z = z + accelerometer_data['z']
14.     print('X=%.3f, Y=%.3f, Z=%.3f'%(x/10.0,y/10.0,z/10.0))
15.     time.sleep(0.3)
16.
17. if __name__ == "__main__":
18.     try:
19.         while True:
20.             mpu6050test()
21.     except:
22.         pass
```

The function reads the initial data of MPU6050's 6 axes and accumulate them. After the loop is completed, obtain the average value to reduce deviation.