
AdafruitMPU6050 Library Documentation

Release 1.0

Bryan Siepert

Feb 15, 2020

Contents

1	Dependencies	3
2	Installing from PyPI	5
3	Usage Example	7
4	Contributing	9
5	Documentation	11
6	Table of Contents	13
6.1	Simple test	13
6.2	Plotter Example	13
6.3	Sleep Example	14
6.4	adafruit_mpu6050	15
6.4.1	Implementation Notes	15
7	Indices and tables	17
	Python Module Index	19
	Index	21

CircuitPython helper library for the MPU6050 6-DoF Accelerometer and Gyroscope

CHAPTER 1

Dependencies

This driver depends on:

- [Adafruit CircuitPython](#)
- [Bus Device](#)
- [Register](#)

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the [Adafruit library and driver bundle](#).

Installing from PyPI

On supported GNU/Linux systems like the Raspberry Pi, you can install the driver locally [from PyPI](#). To install for current user:

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

To install system-wide (this may be required in some cases):

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

To install in a virtual environment in your current project:

```
mkdir project-name && cd project-name
python3 -m venv .env
source .env/bin/activate
pip3 install adafruit-circuitpython-mpu6050
```


CHAPTER 3

Usage Example

```
import time
import board
import busio
import adafruit_mpu6050

i2c = busio.I2C(board.SCL, board.SDA)
mpu = adafruit_mpu6050.MPU6050(i2c)

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)
```


CHAPTER 4

Contributing

Contributions are welcome! Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.

CHAPTER 5

Documentation

For information on building library documentation, please check out [this guide](#).

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/mpu6050_simpletest.py

```
1 import time
2 import board
3 import busio
4 import adafruit_mpu6050
5
6 i2c = busio.I2C(board.SCL, board.SDA)
7 mpu = adafruit_mpu6050.MPU6050(i2c)
8
9 while True:
10     print("Acceleration: X:%.2f, Y: %.2f, Z: %.2f m/s^2"%(mpu.acceleration))
11     print("Gyro X:%.2f, Y: %.2f, Z: %.2f degrees/s"%(mpu.gyro))
12     print("Temperature: %.2f C"%mpu.temperature)
13     print("")
14     time.sleep(1)
```

6.2 Plotter Example

See the effects of changing the gyroscope and accelerometer range by viewing the data in a serial plotter

Listing 2: examples/mpu6050_plotter_example.py

```
1 import time
2 import board
3 import busio
4 import adafruit_mpu6050
```

(continues on next page)

(continued from previous page)

```

5
6 i2c = busio.I2C(board.SCL, board.SDA)
7 mpu = adafruit_mpu6050.MPU6050(i2c)
8 mpu.accelerometer_range = adafruit_mpu6050.Range.RANGE_2_G
9 mpu.gyro_range = adafruit_mpu6050.GyroRange.RANGE_250_DPS
10 while True:
11     # this prints out all the values like a tuple which Mu's plotter prefer
12     print("%.2f, %.2f, %.2f"%(mpu.acceleration), end=", ")
13     print("%.2f, %.2f, %.2f"%(mpu.gyro))
14     time.sleep(0.010)

```

6.3 Sleep Example

Observe how the cycle and sleep modes effect measurements by viewing the data in a serial plotter

Listing 3: examples/mpu6050_sleep_example.py

```

1 import time
2 import board
3 import busio
4 import adafruit_mpu6050
5
6 i2c = busio.I2C(board.SCL, board.SDA)
7 mpu = adafruit_mpu6050.MPU6050(i2c)
8
9 # This example is meant to be used with the serial plotter which makes
10 # it easier to see how the readings change with different settings.
11 # Make sure to poke and prod the sensor while the demo is running to
12 # generate some intersting data!
13
14 while True:
15     # first show some 'normal' readings
16
17     mpu.sleep = False
18     mpu.cycle = False
19
20     for count in range(0, 100):
21         print(mpu.acceleration)
22         time.sleep(0.010)
23
24     # Next, set a slow cycle rate so the effect can be seen clearly.
25     mpu.cycle_Rate = adafruit_mpu6050.Rate.CYCLE_5_HZ
26     # ensure that we're not sleeping or cycle won't work
27     mpu.sleep = False
28     # Finally, enable cycle mode
29     mpu.cycle = True
30
31     for count in range(0, 100):
32         print(mpu.acceleration)
33         time.sleep(0.010)
34
35     # Finally enable sleep mode. Note that while we can still fetch
36     # data from the measurement registers, the measurements are not
37     # updated. In sleep mode the accelerometer and gyroscope are

```

(continues on next page)

(continued from previous page)

```

38     # deactivated to save power, so measurements are halted.
39
40     mpu.cycle = False
41     mpu.sleep = True
42
43     for count in range(0, 100):
44         print(mpu.acceleration)
45         time.sleep(0.010)

```

6.4 adafruit_mpu6050

CircuitPython helper library for the MPU6050 6-DoF Accelerometer and Gyroscope

- Author(s): Bryan Siepert

6.4.1 Implementation Notes

Hardware: * Adafruit's MPU6050 Breakout: <https://adafruit.com/products/3886>

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://github.com/adafruit/circuitpython/releases>
- Adafruit's Bus Device library: https://github.com/adafruit/Adafruit_CircuitPython_BusDevice
- Adafruit's Register library: https://github.com/adafruit/Adafruit_CircuitPython_Register

class `adafruit_mpu6050.Bandwidth`
 Allowed values for `filter_bandwidth`.

- `Bandwidth.BAND_260_HZ`
- `Bandwidth.BAND_184_HZ`
- `Bandwidth.BAND_94_HZ`
- `Bandwidth.BAND_44_HZ`
- `Bandwidth.BAND_21_HZ`
- `Bandwidth.BAND_10_HZ`
- `Bandwidth.BAND_5_HZ`

class `adafruit_mpu6050.GyroRange`
 Allowed values for `gyro_range`.

- `GyroRange.RANGE_250_DPS`
- `GyroRange.RANGE_500_DPS`
- `GyroRange.RANGE_1000_DPS`
- `GyroRange.RANGE_2000_DPS`

class `adafruit_mpu6050.MPU6050` (*i2c_bus, address=104*)
 Driver for the MPU6050 6-DoF accelerometer and gyroscope.

Parameters

- `i2c_bus` (*I2C*) – The I2C bus the MPU6050 is connected to.

- **address** – The I2C slave address of the sensor

acceleration

Acceleration X, Y, and Z axis data in m/s^2

accelerometer_range

The measurement range of all accelerometer axes. Must be a *Range*

cycle

Enable or disable periodic measurement at a rate set by *cycle_rate*. If the sensor was in sleep mode, it will be waken up to cycle

cycle_rate

The rate that measurements are taken while in *cycle* mode. Must be a *Rate*

filter_bandwidth

The bandwidth of the gyroscope Digital Low Pass Filter. Must be a *GyroRange*

gyro

Gyroscope X, Y, and Z axis data in $^\circ/s$

gyro_range

The measurement range of all gyroscope axes. Must be a *GyroRange*

reset ()

Reinitialize the sensor

sample_rate_divisor

The sample rate divisor. See the datasheet for additional detail

sleep

Shuts down the accelerometers and gyroscopes, saving power. No new data will be recorded until the sensor is taken out of sleep by setting to *False*

temperature

The current temperature in $^\circ C$

class adafruit_mpu6050.**Range**

Allowed values for *accelerometer_range*.

- Range.RANGE_2_G
- Range.RANGE_4_G
- Range.RANGE_8_G
- Range.RANGE_16_G

class adafruit_mpu6050.**Rate**

Allowed values for *cycle_rate*.

- Rate.CYCLE_1_25_HZ
- Rate.CYCLE_5_HZ
- Rate.CYCLE_20_HZ
- Rate.CYCLE_40_HZ

- Adafruit's MPU6050 Breakout: <https://adafruit.com/products/3886>

CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`

a

`adafruit_mpu6050`, 15

A

acceleration (*adafruit_mpu6050.MPU6050 attribute*), 16
accelerometer_range (*adafruit_mpu6050.MPU6050 attribute*), 16
adafruit_mpu6050 (*module*), 15

B

Bandwidth (*class in adafruit_mpu6050*), 15

C

cycle (*adafruit_mpu6050.MPU6050 attribute*), 16
cycle_rate (*adafruit_mpu6050.MPU6050 attribute*), 16

F

filter_bandwidth (*adafruit_mpu6050.MPU6050 attribute*), 16

G

gyro (*adafruit_mpu6050.MPU6050 attribute*), 16
gyro_range (*adafruit_mpu6050.MPU6050 attribute*), 16
GyroRange (*class in adafruit_mpu6050*), 15

M

MPU6050 (*class in adafruit_mpu6050*), 15

R

Range (*class in adafruit_mpu6050*), 16
Rate (*class in adafruit_mpu6050*), 16
reset () (*adafruit_mpu6050.MPU6050 method*), 16

S

sample_rate_divisor (*adafruit_mpu6050.MPU6050 attribute*), 16
sleep (*adafruit_mpu6050.MPU6050 attribute*), 16

T

temperature (*adafruit_mpu6050.MPU6050 attribute*), 16