

---

# **AdafruitMMA8451 Library Documentation**

***Release 1.0***

**Tony DiCola**

**Aug 23, 2019**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Usage Example</b>	<b>5</b>
<b>3</b>	<b>Contributing</b>	<b>7</b>
<b>4</b>	<b>Building locally</b>	<b>9</b>
4.1	Zip release files . . . . .	9
4.2	Sphinx documentation . . . . .	9
<b>5</b>	<b>Table of Contents</b>	<b>11</b>
5.1	Simple test . . . . .	11
5.2	adafruit_mma8451 . . . . .	12
<b>6</b>	<b>Indices and tables</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>
	<b>Index</b>	<b>19</b>



CircuitPython module for the MMA8451 3 axis accelerometer.



# CHAPTER 1

---

## Dependencies

---

This driver depends on:

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

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





## CHAPTER 2

---

### Usage Example

---

See `examples/simpletest.py` for a demo of the usage.



## CHAPTER 3

---

### Contributing

---

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



### 4.1 Zip release files

To build this library locally you'll need to install the `circuitpython-build-tools` package.

```
python3 -m venv .env
source .env/bin/activate
pip install circuitpython-build-tools
```

Once installed, make sure you are in the virtual environment:

```
source .env/bin/activate
```

Then run the build:

```
circuitpython-build-bundles --filename_prefix adafruit-circuitpython-mma8451 --
↳library_location .
```

### 4.2 Sphinx documentation

Sphinx is used to build the documentation based on rST files and comments in the code. First, install dependencies (feel free to reuse the virtual environment from above):

```
python3 -m venv .env
source .env/bin/activate
pip install Sphinx sphinx-rtd-theme
```

Now, once you have the virtual environment activated:

```
cd docs
sphinx-build -E -W -b html . _build/html
```

This will output the documentation to `docs/_build/html`. Open the `index.html` in your browser to view them. It will also (due to `-W`) error out on any warning like Travis will. This is a good way to locally verify it will pass.

## 5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/mma8451\_simpletest.py

```
1  # Simple demo of reading the MMA8451 orientation every second.
2  # Author: Tony DiCola
3  import time
4
5  import board
6  import busio
7
8  import adafruit_mma8451
9
10
11  # Initialize I2C bus.
12  i2c = busio.I2C(board.SCL, board.SDA)
13
14  # Initialize MMA8451 module.
15  sensor = adafruit_mma8451.MMA8451(i2c)
16  # Optionally change the address if it's not the default:
17  #sensor = adafruit_mma8451.MMA8451(i2c, address=0x1C)
18
19  # Optionally change the range from its default of +/-4G:
20  #sensor.range = adafruit_mma8451.RANGE_2G # +/- 2G
21  #sensor.range = adafruit_mma8451.RANGE_4G # +/- 4G (default)
22  #sensor.range = adafruit_mma8451.RANGE_8G # +/- 8G
23
24  # Optionally change the data rate from its default of 800hz:
25  #sensor.data_rate = adafruit_mma8451.DATARATE_800HZ # 800Hz (default)
26  #sensor.data_rate = adafruit_mma8451.DATARATE_400HZ # 400Hz
27  #sensor.data_rate = adafruit_mma8451.DATARATE_200HZ # 200Hz
```

(continues on next page)

(continued from previous page)

```

28 #sensor.data_rate = adafruit_mma8451.DATARATE_100HZ # 100Hz
29 #sensor.data_rate = adafruit_mma8451.DATARATE_50HZ # 50Hz
30 #sensor.data_rate = adafruit_mma8451.DATARATE_12_5HZ # 12.5Hz
31 #sensor.data_rate = adafruit_mma8451.DATARATE_6_25HZ # 6.25Hz
32 #sensor.data_rate = adafruit_mma8451.DATARATE_1_56HZ # 1.56Hz
33
34 # Main loop to print the acceleration and orientation every second.
35 while True:
36     x, y, z = sensor.acceleration
37     print('Acceleration: x={0:0.3f}m/s^2 y={1:0.3f}m/s^2 z={2:0.3f}m/s^2'.format(x, y,
→ z))
38     orientation = sensor.orientation
39     # Orientation is one of these values:
40     # - PL_PUF: Portrait, up, front
41     # - PL_PUB: Portrait, up, back
42     # - PL_PDF: Portrait, down, front
43     # - PL_PDB: Portrait, down, back
44     # - PL_LRF: Landscape, right, front
45     # - PL_LRB: Landscape, right, back
46     # - PL_LLF: Landscape, left, front
47     # - PL_LLB: Landscape, left, back
48     print('Orientation: ', end='')
49     if orientation == adafruit_mma8451.PL_PUF:
50         print('Portrait, up, front')
51     elif orientation == adafruit_mma8451.PL_PUB:
52         print('Portrait, up, back')
53     elif orientation == adafruit_mma8451.PL_PDF:
54         print('Portrait, down, front')
55     elif orientation == adafruit_mma8451.PL_PDB:
56         print('Portrait, down, back')
57     elif orientation == adafruit_mma8451.PL_LRF:
58         print('Landscape, right, front')
59     elif orientation == adafruit_mma8451.PL_LRB:
60         print('Landscape, right, back')
61     elif orientation == adafruit_mma8451.PL_LLF:
62         print('Landscape, left, front')
63     elif orientation == adafruit_mma8451.PL_LLB:
64         print('Landscape, left, back')
65     time.sleep(1.0)

```

## 5.2 adafruit\_mma8451

CircuitPython module for the MMA8451 3 axis accelerometer. See examples/simpletest.py for a demo of the usage.

- Author(s): Tony DiCola

**class** adafruit\_mma8451.**MMA8451** (i2c, \*, address=29)

MMA8451 accelerometer. Create an instance by specifying: - i2c: The I2C bus connected to the sensor.

Optionally specify: - address: The I2C address of the sensor if not the default of 0x1D.

### **acceleration**

Get the acceleration measured by the sensor. Will return a 3-tuple of X, Y, Z axis acceleration values in m/s<sup>2</sup>.

### **data\_rate**



Get and set the data rate of the sensor. Must be a value of: - DATARATE\_800HZ: 800Hz (the default) - DATARATE\_400HZ: 400Hz - DATARATE\_200HZ: 200Hz - DATARATE\_100HZ: 100Hz - DATARATE\_50HZ: 50Hz - DATARATE\_12\_5HZ: 12.5Hz - DATARATE\_6\_25HZ: 6.25Hz - DATARATE\_1\_56HZ: 1.56Hz

**orientation**

Get the orientation of the MMA8451. Will return a value of: - PL\_PUF: Portrait, up, front - PL\_PUB: Portrait, up, back - PL\_PDF: Portrait, down, front - PL\_PDB: Portrait, down, back - PL\_LRF: Landscape, right, front - PL\_LRB: Landscape, right, back - PL\_LLF: Landscape, left, front - PL\_LLB: Landscape, left, back

**range**

Get and set the range of the sensor. Must be a value of: - RANGE\_8G: +/- 8g - RANGE\_4G: +/- 4g (the default) - RANGE\_2G: +/- 2g



## CHAPTER 6

---

### Indices and tables

---

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



### a

adafruit\_mma8451, [12](#)



## A

`acceleration` (*adafruit\_mma8451.MMA8451 attribute*), [12](#)  
`adafruit_mma8451` (*module*), [12](#)

## D

`data_rate` (*adafruit\_mma8451.MMA8451 attribute*),  
[12](#)

## M

`MMA8451` (*class in adafruit\_mma8451*), [12](#)

## O

`orientation` (*adafruit\_mma8451.MMA8451 attribute*), [13](#)

## R

`range` (*adafruit\_mma8451.MMA8451 attribute*), [13](#)