AdafruitMMA8451 Library Documentation

Release 1.0

Tony DiCola

Aug 19, 2018

Contents

1 Dependencies **3**

2 Usage Example **5**

3 Contributing **7**

4 Building locally **9**
4.1 Zip release files 9
4.2 Sphinx documentation 9

5 Table of Contents **11**
5.1 Simple test 11
5.2 `adafruit_mma8451` 12

6 Indices and tables **15**

Python Module Index **17**

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.

CHAPTER 4

Building locally

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.

CHAPTER 5

Table of Contents

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/simpletest.py

```
# Simple demo of reading the MMA8451 orientation every second.
# Author: Tony DiCola
import time

import board
import busio

import adafruit_mma8451


# Initialize I2C bus.
i2c = busio.I2C(board.SCL, board.SDA)

# Initialize MMA8451 module.
sensor = adafruit_mma8451.MMA8451(i2c)
# Optionally change the address if it's not the default:
#sensor = adafruit_mma8451.MMA8451(i2c, address=0x1C)

# Optionally change the range from its default of +/-4G:
#sensor.range = adafruit_mma8451.RANGE_2G  # +/- 2G
#sensor.range = adafruit_mma8451.RANGE_4G  # +/- 4G (default)
#sensor.range = adafruit_mma8451.RANGE_8G  # +/- 8G

# Optionally change the data rate from its default of 800hz:
#sensor.data_rate = adafruit_mma8451.DATARATE_800HZ  #  800Hz (default)
#sensor.data_rate = adafruit_mma8451.DATARATE_400HZ  #  400Hz
#sensor.data_rate = adafruit_mma8451.DATARATE_200HZ  #  200Hz
```

(continues on next page)

(continued from previous page)

```
#sensor.data_rate = adafruit_mma8451.DATARATE_100HZ  #  100Hz
#sensor.data_rate = adafruit_mma8451.DATARATE_50HZ   #   50Hz
#sensor.data_rate = adafruit_mma8451.DATARATE_12_5HZ # 12.5Hz
#sensor.data_rate = adafruit_mma8451.DATARATE_6_25HZ # 6.25Hz
#sensor.data_rate = adafruit_mma8451.DATARATE_1_56HZ # 1.56Hz

# Main loop to print the acceleration and orientation every second.
while True:
    x, y, z = sensor.acceleration
    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))
    orientation = sensor.orientation
    # Orientation is one of these values:
    #  - 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
    print('Orientation: ', end='')
    if orientation == adafruit_mma8451.PL_PUF:
        print('Portrait, up, front')
    elif orientation == adafruit_mma8451.PL_PUB:
        print('Portrait, up, back')
    elif orientation == adafruit_mma8451.PL_PDF:
        print('Portrait, down, front')
    elif orientation == adafruit_mma8451.PL_PDB:
        print('Portrait, down, back')
    elif orientation == adafruit_mma8451.PL_LRF:
        print('Landscape, right, front')
    elif orientation == adafruit_mma8451.PL_LRB:
        print('Landscape, right, back')
    elif orientation == adafruit_mma8451.PL_LLF:
        print('Landscape, left, front')
    elif orientation == adafruit_mma8451.PL_LLB:
        print('Landscape, left, back')
    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^2.

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

Python Module Index

a

adafruit_mma8451, 12

Index

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