AdafruitLSM303 Library Documentation

Release 1.0

Dave Astels

Dec 21, 2018

Contents

Adafruit CircuitPython module for the LSM303 6-DoF with 3-axis accelerometer and magnetometer

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

```
import time
import board
import busio

import adafruit_lsm303

i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_lsm303.LSM303(i2c)

while True:
        raw_accel_x, raw_accel_y, raw_accel_z = sensor.raw_acceleration
        accel_x, accel_y, accel_z = sensor.acceleration
        raw_mag_x, raw_mag_y, raw_mag_z = sensor.raw_magnetic
        mag_x, mag_y, mag_z = sensor.magnetic

        print('Acceleration raw: ({0:6d}, {1:6d}, {2:6d}), (m/s^2): ({3:10.3f}, {4:10.
↪3f}, {5:10.3f})'.format(raw_accel_x, raw_accel_y, raw_accel_z, accel_x, accel_y,␣
↪accel_z))
        print('Magnetometer raw: ({0:6d}, {1:6d}, {2:6d}), (gauss): ({3:10.3f}, {4:10.
↪3f}, {5:10.3f})'.format(raw_mag_x, raw_mag_y, raw_mag_z, mag_x, mag_y, mag_z))
        print('')
        time.sleep(1.0)
```

CHAPTER 3

Contributing

Contributions are welcome! Please read our Code of Conduct before contributing to help this project stay welcoming.

CHAPTER 4

Building locally

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-lsm303 --library_
↪location .
```

4.1 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 tests

Ensure your device works with these simple tests.

Listing 1: examples/lsm303_simpletest.py

```
""" Display both accelerometer and magnetometer data once per second """

import time
import board
import busio
import adafruit_lsm303

i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_lsm303.LSM303(i2c)

while True:
    acc_x, acc_y, acc_z = sensor.acceleration
    mag_x, mag_y, mag_z = sensor.magnetic

    print('Acceleration (m/s^2): ({0:10.3f}, {1:10.3f}, {2:10.3f})'.format(acc_x, acc_
↪y, acc_z))
    print('Magnetometer (gauss): ({0:10.3f}, {1:10.3f}, {2:10.3f})'.format(mag_x, mag_
↪y, mag_z))
    print('')
    time.sleep(1.0)
```

Listing 2: examples/fast_accel/fast_accel.py

```
""" Read data from the accelerometer and print it out, ASAP! """

import board
import busio
```

(continues on next page)

(continued from previous page)

```

import adafruit_lsm303

i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_lsm303.LSM303(i2c)

while True:
    accel_x, accel_y, accel_z = sensor.acceleration
    print('{0:10.3f} {1:10.3f} {2:10.3f}'.format(accel_x, accel_y, accel_z))
```

Listing 3: examples/fast_mag/fast_mag.py

```
""" Read data from the magnetometer and print it out, ASAP! """

import board
import busio
import adafruit_lsm303

i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_lsm303.LSM303(i2c)

while True:
    mag_x, mag_y, mag_z = sensor.magnetic
    print('{0:10.3f} {1:10.3f} {2:10.3f}'.format(mag_x, mag_y, mag_z))
```

Listing 4: examples/raw_and_cooked/raw_and_cooked.py

```
""" Display both accelerometer and magnetometer data once per second """

import time
import board
import busio

import adafruit_lsm303

i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_lsm303.LSM303(i2c)

while True:
    raw_accel_x, raw_accel_y, raw_accel_z = sensor.raw_acceleration
    accel_x, accel_y, accel_z = sensor.acceleration
    raw_mag_x, raw_mag_y, raw_mag_z = sensor.raw_magnetic
    mag_x, mag_y, mag_z = sensor.magnetic

    print('Acceleration raw: ({0:6d}, {1:6d}, {2:6d}), (m/s^2): ({3:10.3f}, {4:10.3f},
↪ {5:10.3f})'
          .format(raw_accel_x, raw_accel_y, raw_accel_z, accel_x, accel_y, accel_z))
    print('Magnetometer raw: ({0:6d}, {1:6d}, {2:6d}), (gauss): ({3:10.3f}, {4:10.3f},
↪ {5:10.3f})'
          .format(raw_mag_x, raw_mag_y, raw_mag_z, mag_x, mag_y, mag_z))
    print('')
    time.sleep(1.0)
```

5.2 adafruit_lsm303

CircuitPython driver for the LSM303 accelerometer + magnetometer.

- Author(s): Dave Astels

5.2.1 Implementation Notes

Hardware:

- Adafruit Triple-axis Accelerometer+Magnetometer (Compass) Board - LSM303 (Product ID: 1120)
- Adafruit FLORA Accelerometer/Compass Sensor - LSM303 - v1.0 (Product ID: 1247)

Software and Dependencies:

- Adafruit CircuitPython firmware for the ESP8622 and M0-based boards: https://github.com/adafruit/circuitpython/releases
- Adafruit's Bus Device library: https://github.com/adafruit/Adafruit_CircuitPython_BusDevice

class adafruit_lsm303.**LSM303**(*i2c*)

Driver for the LSM303 accelerometer/magnetometer.

acceleration

The processed accelerometer sensor values. A 3-tuple of X, Y, Z axis values in meters per second squared that are signed floats.

mag_gain

The magnetometer's gain.

mag_rate

The magnetometer update rate.

magnetic

The processed magnetometer sensor values. A 3-tuple of X, Y, Z axis values in microteslas that are signed floats.

raw_acceleration

The raw accelerometer sensor values. A 3-tuple of X, Y, Z axis values that are 16-bit signed integers.

raw_magnetic

The raw magnetometer sensor values. A 3-tuple of X, Y, Z axis values that are 16-bit signed integers.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index