Adafruit LSM303 Accelerometer Library Documentation

Release 1.0

Bryan Siepert

Jun 07, 2021

Contents

Adafruit CircuitPython module for the LSM303's accelerometer

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.

CHAPTER 2

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-lsm303_accel
```

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

```
sudo pip3 install adafruit-circuitpython-lsm303_accel
```

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-lsm303_accel
```

CHAPTER 3

Usage Example

```
import time
import board
import adafruit_lsm303_accel

i2c = board.I2C()  # uses board.SCL and board.SDA
sensor = adafruit_lsm303_accel.LSM303_Accel(i2c)

while True:
    acc_x, acc_y, acc_z = sensor.acceleration

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

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.

CHAPTER 6

Table of Contents

6.1 Simple tests

Ensure your device works with these simple tests.

Listing 1: examples/lsm303_simpletest.py

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

""" Display accelerometer data once per second """

import time
import board
import adafruit_lsm303_accel

i2c = board.I2C()  # uses board.SCL and board.SDA
sensor = adafruit_lsm303_accel.LSM303_Accel(i2c)

while True:
    acc_x, acc_y, acc_z = sensor.acceleration

    print(
        "Acceleration (m/s^2): ({0:10.3f}, {1:10.3f}, {2:10.3f})".format(
            acc_x, acc_y, acc_z
        )
    )
    print("")
    time.sleep(1.0)
```

6.2 Fast Acceleration Example

Example to demonstrate fast acceleration data acquisition

Listing 2: examples/lsm303_fast_accel.py

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

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

import board
import adafruit_lsm303_accel

i2c = board.I2C()  # uses board.SCL and board.SDA
sensor = adafruit_lsm303_accel.LSM303_Accel(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))
```

6.3 Inclinometer Example

Demonstrate inclinometer example

Listing 3: examples/lsm303_accel_inclinometer.py

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

""" Display inclination data five times per second """

import time
from math import atan2, degrees
import board
import adafruit_lsm303_accel


i2c = board.I2C()  # uses board.SCL and board.SDA
sensor = adafruit_lsm303_accel.LSM303_Accel(i2c)


def vector_2_degrees(x, y):
    angle = degrees(atan2(y, x))
    if angle < 0:
        angle += 360
    return angle


def get_inclination(_sensor):
    x, y, z = _sensor.acceleration
    return vector_2_degrees(x, z), vector_2_degrees(y, z)


while True:
```

(continues on next page)

(continued from previous page)

```
    angle_xz, angle_yz = get_inclination(sensor)
    print("XZ angle = {:6.2f}deg   YZ angle = {:6.2f}deg".format(angle_xz, angle_yz))
    time.sleep(0.2)
```

6.4 Tap Detection Example

Tap detection example

Listing 4: examples/lsm303_accel_tap_detection.py

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

import board
import adafruit_lsm303_accel

i2c = board.I2C()  # uses board.SCL and board.SDA
accel = adafruit_lsm303_accel.LSM303_Accel(i2c)
accel.range = adafruit_lsm303_accel.Range.RANGE_8G
accel.set_tap(1, 30)

while True:
    if accel.tapped:
        print("Tapped!\n")
```

6.5 adafruit_lsm303_accel

CircuitPython driver for the accelerometer in LSM303 sensors.

- Author(s): Dave Astels, Bryan Siepert

6.5.1 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 supported boards: https://circuitpython.org/downloads
- Adafruit's Bus Device library: https://github.com/adafruit/Adafruit_CircuitPython_BusDevice

class adafruit_lsm303_accel.**LSM303_Accel**(*i2c*)

Driver for the LSM303's accelerometer.

Parameters **i2c** (*I2C*) – The I2C bus the device is connected to.

Quickstart: Importing and using the device

Here is an example of using the *LSM303_Accel* class. First you will need to import the libraries to use the sensor

```
import board
import adafruit_lsm303_accel
```

Once this is done you can define your `board.I2C` object and define your sensor object

```
i2c = board.I2C()  # uses board.SCL and board.SDA
sensor = adafruit_lsm303_accel.LSM303_Accel(i2c)
```

Now you have access to the *acceleration* attribute

```
acc_x, acc_y, acc_z = sensor.acceleration
```

acceleration
The measured accelerometer sensor values. A 3-tuple of X, Y, Z axis values in m/s^2 squared that are signed floats.

data_rate
Select the rate at which the sensor takes measurements. Must be a *Rate*

mode
Sets the power mode of the sensor. The mode must be a *Mode*. Note that the mode and range will both affect the accuracy of the sensor

range
Adjusts the range of values that the sensor can measure, from +- 2G to +-16G Note that larger ranges will be less accurate. Must be a *Range*

set_tap(*tap*, *threshold*, *, *time_limit=10*, *time_latency=20*, *time_window=255*, *tap_cfg=None*)
The tap detection parameters.

Parameters

- **tap** (*int*) – 0 to disable tap detection, 1 to detect only single taps, and 2 to detect only double taps.
- **threshold** (*int*) – A threshold for the tap detection. The higher the value the less sensitive the detection. This changes based on the accelerometer range. Good values are 5-10 for 16G, 10-20 for 8G, 20-40 for 4G, and 40-80 for 2G.
- **time_limit** (*int*) – TIME_LIMIT register value. Defaults to `10`
- **time_latency** (*int*) – TIME_LATENCY register value. Defaults to `20`
- **time_window** (*int*) – TIME_WINDOW register value. Defaults to `255`
- **tap_cfg** (*int*) – CLICK_CFG register value. Defaults to `None`

tapped
True if a tap was detected recently. Whether its a single tap or double tap is determined by the tap param on *set_tap()*. *tapped* may be True over multiple reads even if only a single tap or single double tap occurred.

class adafruit_lsm303_accel.**Mode**
Options for *mode*

class adafruit_lsm303_accel.**Range**
Options for *range*

class adafruit_lsm303_accel.**Rate**
Options for *data_rate*

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index