
Adafruit FXOS8700 Library Documentation

Release 1.0

Tony DiCola

Oct 21, 2019

Contents

1	Dependencies	3
2	Installing from PyPI	5
3	Usage Example	7
4	Contributing	9
5	Building locally	11
5.1	Sphinx documentation	11
6	Table of Contents	13
6.1	Simple test	13
6.2	adafruit_fxos8700	14
6.2.1	Implementation Notes	14
7	Indices and tables	15
	Python Module Index	17
	Index	19

CircuitPython module for the NXP FXOS8700 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

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-fxos8700
```

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

```
sudo pip3 install adafruit-circuitpython-fxos8700
```

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-fxos8700
```


CHAPTER 3

Usage Example

```
import time
import board
import busio
import adafruit_fxos8700

i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_fxos8700.FXOS8700(i2c)

while True:
    accel_x, accel_y, accel_z = sensor.accelerometer
    mag_x, mag_y, mag_z = sensor.magnetometer
    print('Acceleration (m/s^2): ({0:0.3f}, {1:0.3f}, {2:0.3f})'.format(accel_x,
↪ accel_y, accel_z))
    print('Magnetometer (uTesla): ({0:0.3f}, {1:0.3f}, {2:0.3f})'.format(mag_x, mag_y,
↪ mag_z))
    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

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-fxos8700 --
↳library_location .
```

5.1 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

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/fxos8700_simpletest.py

```
1  # Simple demo of the FXOS8700 accelerometer and magnetometer.
2  # Will print the acceleration and magnetometer values every second.
3  import time
4
5  import board
6  import busio
7
8  import adafruit_fxos8700
9
10
11 # Initialize I2C bus and device.
12 i2c = busio.I2C(board.SCL, board.SDA)
13 sensor = adafruit_fxos8700.FXOS8700(i2c)
14 # Optionally create the sensor with a different accelerometer range (the
15 # default is 2G, but you can use 4G or 8G values):
16 #sensor = adafruit_fxos8700.FXOS8700(i2c, accel_range=adafruit_fxos8700.ACCEL_RANGE_
17 #    ↳ 4G)
18 #sensor = adafruit_fxos8700.FXOS8700(i2c, accel_range=adafruit_fxos8700.ACCEL_RANGE_
19 #    ↳ 8G)
20
21 # Main loop will read the acceleration and magnetometer values every second
22 # and print them out.
23 while True:
24     # Read acceleration & magnetometer.
25     accel_x, accel_y, accel_z = sensor.accelerometer
26     mag_x, mag_y, mag_z = sensor.magnetometer
27     # Print values.
```

(continues on next page)

(continued from previous page)

```
26     print('Acceleration (m/s^2): ({0:0.3f}, {1:0.3f}, {2:0.3f})'.format(accel_x, ↵
↵ accel_y, accel_z))
27     print('Magnetometer (uTesla): ({0:0.3f}, {1:0.3f}, {2:0.3f})'.format(mag_x, mag_y,
↵ mag_z))
28     # Delay for a second.
29     time.sleep(1.0)
```

6.2 adafruit_fxos8700

CircuitPython module for the NXP FXOS8700 accelerometer and magnetometer. Based on the driver from: https://github.com/adafruit/Adafruit_FXOS8700

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

- Author(s): Tony DiCola

6.2.1 Implementation Notes

Hardware:

- Adafruit Precision NXP 9-DOF Breakout Board - FXOS8700 + FXAS21002 (Product ID: 3463)

Software and Dependencies:

- Adafruit CircuitPython firmware (2.2.0+) 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_fxos8700.FXOS8700` (*i2c*, *address=31*, *accel_range=0*)
Driver for the NXP FXOS8700 accelerometer and magnetometer.

accelerometer

Read the acceleration from the accelerometer and return its X, Y, Z axis values as a 3-tuple in m/s².

magnetometer

Read the magnetometer values and return its X, Y, Z axis values as a 3-tuple in uTeslas.

read_raw_accel_mag()

Read the raw accelerometer and magnetometer readings. Returns a 2-tuple of 3-tuples:

- Accelerometer X, Y, Z axis 14-bit signed raw values
- Magnetometer X, Y, Z axis 16-bit signed raw values

If you want the acceleration or magnetometer values in friendly units consider using the `accelerometer` and `magnetometer` properties!

CHAPTER 7

Indices and tables

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

a

adafruit_fxos8700, [14](#)

A

`accelerometer` (*adafruit_fxos8700.FXOS8700* attribute), [14](#)

`adafruit_fxos8700` (*module*), [14](#)

F

`FXOS8700` (*class in adafruit_fxos8700*), [14](#)

M

`magnetometer` (*adafruit_fxos8700.FXOS8700* attribute), [14](#)

R

`read_raw_accel_mag()`
(*adafruit_fxos8700.FXOS8700* method),
[14](#)