
AdafruitMPL3115A2 Library Documentation

Release 1.0

Tony DiCola

Oct 25, 2021

Contents

1	Dependencies	3
2	Installing from PyPI	5
3	Usage Example	7
4	Documentation	9
5	Contributing	11
6	Documentation	13
7	Table of Contents	15
7.1	Simple test	15
7.2	adafruit_mpl3115a2	16
7.2.1	Implementation Notes	16
8	Indices and tables	19
	Python Module Index	21
	Index	23

CircuitPython module for the MPL3115A2 barometric pressure & temperature sensor.

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-mp13115a2
```

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

```
sudo pip3 install adafruit-circuitpython-mp13115a2
```

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-mp13115a2
```


CHAPTER 3

Usage Example

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

CHAPTER 4

Documentation

API documentation for this library can be found on [Read the Docs](#).

CHAPTER 5

Contributing

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

CHAPTER 6

Documentation

For information on building library documentation, please check out [this guide](#).

7.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/mpl3115a2_simpletest.py

```
1  # SPDX-FileCopyrightText: 2019 Tony DiCola for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  # Simple demo of the MPL3115A2 sensor.
5  # Will read the pressure and temperature and print them out every second.
6  import time
7  import board
8  import adafruit_mpl3115a2
9
10
11 # Create sensor object, communicating over the board's default I2C bus
12 i2c = board.I2C() # uses board.SCL and board.SDA
13
14 # Initialize the MPL3115A2.
15 sensor = adafruit_mpl3115a2.MPL3115A2(i2c)
16 # Alternatively you can specify a different I2C address for the device:
17 # sensor = adafruit_mpl3115a2.MPL3115A2(i2c, address=0x10)
18
19 # You can configure the pressure at sealevel to get better altitude estimates.
20 # This value has to be looked up from your local weather forecast or meteorological
21 # reports. It will change day by day and even hour by hour with weather
22 # changes. Remember altitude estimation from barometric pressure is not exact!
23 # Set this to a value in pascals:
24 sensor.sealevel_pressure = 102250
25
26 # Main loop to read the sensor values and print them every second.
27 while True:
```

(continues on next page)

(continued from previous page)

```

28     pressure = sensor.pressure
29     print("Pressure: {0:0.3f} pascals".format(pressure))
30     altitude = sensor.altitude
31     print("Altitude: {0:0.3f} meters".format(altitude))
32     temperature = sensor.temperature
33     print("Temperature: {0:0.3f} degrees Celsius".format(temperature))
34     time.sleep(1.0)

```

7.2 adafruit_mpl3115a2

CircuitPython module for the MPL3115A2 barometric pressure & temperature sensor.

- Author(s): Tony DiCola

7.2.1 Implementation Notes

Hardware:

- Adafruit MPL3115A2 - I2C Barometric Pressure/Altitude/Temperature Sensor

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_mpl3115a2.MPL3115A2` (*i2c*, *, *address=96*)
Instance of the MPL3115A2 sensor.

Parameters

- **i2c** (*I2C*) – The I2C bus the MPL3115A2 is connected to.
- **address** (*int*) – The I2C device address. Defaults to 0x60

Quickstart: Importing and using the MPL3115A2

Here is an example of using the `MPL3115A2` class. First you will need to import the libraries to use the sensor

```
import board
import adafruit_mpl3115a2
```

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_mpl3115a2.MPL3115A2(i2c)
```

Now you have access to the `temperature`, `pressure` and `altitude` attributes

```
temperature = sensor.temperature
pressure = sensor.pressure
altitude = sensor.altitude
```

altitude

Read the altitude as calculated based on the sensor pressure and previously configured pressure at sea-level.

This will return a value in meters. Set the sea-level pressure by updating the *sealevel_pressure* property first to get a more accurate altitude value.

pressure

Read the barometric pressure detected by the sensor in Pascals.

sealevel_pressure

Read and write the pressure at sea-level used to calculate altitude. You must look this up from a local weather or meteorological report for the best accuracy. This is a value in Pascals.

temperature

Read the temperature as measured by the sensor in degrees Celsius.

CHAPTER 8

Indices and tables

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

a

`adafruit_mpl3115a2`, 16

A

adafruit_mpl3115a2 (*module*), 16
altitude (*adafruit_mpl3115a2.MPL3115A2 attribute*), 16

M

MPL3115A2 (*class in adafruit_mpl3115a2*), 16

P

pressure (*adafruit_mpl3115a2.MPL3115A2 attribute*), 17

S

sealevel_pressure
(*adafruit_mpl3115a2.MPL3115A2 attribute*),
17

T

temperature (*adafruit_mpl3115a2.MPL3115A2 attribute*), 17