AdafruitMPL3115A2 Library Documentation

Release 1.0

Tony DiCola

Jan 22, 2020

Contents

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-mpl3115a2
```

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

```
sudo pip3 install adafruit-circuitpython-mpl3115a2
```

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-mpl3115a2
```

CHAPTER 3

Usage Example

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

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 test

Ensure your device works with this simple test.

Listing 1: examples/mpl3115a2_simpletest.py

```
# Simple demo of the MPL3115A2 sensor.
# Will read the pressure and temperature and print them out every second.
# Author: Tony DiCola
import time

import board
import busio

import adafruit_mpl3115a2


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

# Initialize the MPL3115A2.
sensor = adafruit_mpl3115a2.MPL3115A2(i2c)
# Alternatively you can specify a different I2C address for the device:
#sensor = adafruit_mpl3115a2.MPL3115A2(i2c, address=0x10)

# You can configure the pressure at sealevel to get better altitude estimates.
# This value has to be looked up from your local weather forecast or meteorlogical
# reports.  It will change day by day and even hour by hour with weather
# changes.  Remember altitude estimation from barometric pressure is not exact!
# Set this to a value in pascals:
sensor.sealevel_pressure = 102250

# Main loop to read the sensor values and print them every second.
```

(continues on next page)

(continued from previous page)

```
while True:
    pressure = sensor.pressure
    print('Pressure: {0:0.3f} pascals'.format(pressure))
    altitude = sensor.altitude
    print('Altitude: {0:0.3f} meters'.format(altitude))
    temperature = sensor.temperature
    print('Temperature: {0:0.3f} degrees Celsius'.format(temperature))
    time.sleep(1.0)
```

6.2 adafruit_mpl3115a2

CircuitPython module for the MPL3115A2 barometric pressure & temperature sensor. See examples/simpletest.py for a demo of the usage.

- Author(s): Tony DiCola

class adafruit_mpl3115a2.**MPL3115A2**(*i2c*, ***, *address=96*)

Instance of the MPL3115A2 sensor. Must specify the following parameters when creating an instance of this device: - i2c: The I2C bus connected to the sensor.

In addition you can specify the following optional keyword arguments: - address: The I2C address of the device if it's different from the default.

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 meteorlogical report for the best accuracy. This is a value in Pascals.

temperature

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

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index