AdafruitBH1750 Library Documentation

Release 1.0

Bryan Siepert

Jul 08, 2020

Contents

CircuitPython library for use with the Adafruit BH1750 Ambient Light Sensor Breakout

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-bh1750
```

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

```
sudo pip3 install adafruit-circuitpython-bh1750
```

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-bh1750
```

CHAPTER 3

Usage Example

```
import time
import board
import busio
import adafruit_bh1750

i2c = busio.I2C(board.SCL, board.SDA)

sensor = adafruit_bh1750.BH1750(i2c)

while True:
    print("%.2f Lux"%sensor.lux)
    time.sleep(1)
```

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/bh1750_simpletest.py

```
# SPDX-FileCopyrightText: 2020 Bryan Siepert, written for Adafruit Industries

# SPDX-License-Identifier: Unlicense
import time
import board
import adafruit_bh1750

i2c = board.I2C()

sensor = adafruit_bh1750.BH1750(i2c)

while True:
    print("%.2f Lux" % sensor.lux)
    time.sleep(1)
```

6.2 adafruit_bh1750

CircuitPython library for use with the Adafruit BH1750 breakout

- Author(s): Bryan Siepert

6.2.1 Implementation Notes

Hardware:

- Adafruit BH1750 Breakout

Software and Dependencies:

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

class adafruit_bh1750.**BH1750**(*i2c_bus*, *address=35*)
Library for the BH1750 Sensor

Parameters

- **i2c_bus** (*I2C*) – The I2C bus the BH1750 is connected to.
- **address** – The I2C slave address of the sensor. Defaults to `0x23`. Can be set to `0x5C` by pulling the address pin high.

initialize()
Configure the sensors with the default settings.

lux
Light value in lux.

This example prints the light data in lux. Cover the sensor to see the values change.

```
import time
import board
import busio
import adafruit_bh1750

i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_bh1750.BH1750(i2c)

while True:
    print("Lux:", sensor.lux)
    time.sleep(0.1)
```

class adafruit_bh1750.**CV**
struct helper

classmethod add_values(*value_tuples*)
Add CV values to the class

classmethod is_valid(*value*)
Validate that a given value is a member

class adafruit_bh1750.**Mode**
Options for `mode`

class adafruit_bh1750.**RWBitfields**(*num_bits*, *lowest_bit*)
A class to do bitwise operations to get and set a range of bits within a byte but gets and sets the full byte value from the `_settings` attribute of the calling object.

Values are `int` between 0 and `2**num_bits - 1`

Parameters

- **num_bits** (*int*) – The number of bits in the field.
- **lowest_bit** (*type*) – The lowest bits index within the byte at `register_address`

class adafruit_bh1750.**Resolution**
Options for `resolution`

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index