
AdafruitTSL2591 Library Documentation

Release 1.0

Tony DiCola

Aug 13, 2018

Contents

| | | |
|----------|--------------------------------|-----------|
| 1 | Dependencies | 3 |
| 2 | Usage Example | 5 |
| 3 | Contributing | 7 |
| 4 | Building locally | 9 |
| 4.1 | Sphinx documentation | 9 |
| 5 | Table of Contents | 11 |
| 5.1 | Simple test | 11 |
| 5.2 | adafruit_tsl2591 | 12 |
| 5.2.1 | Implementation Notes | 12 |
| 6 | Indices and tables | 15 |
| | Python Module Index | 17 |

CircuitPython module for the TSL2591 high precision light 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

Usage Example

See `examples/tsl2591_simpletest.py` for a demo of the usage.

CHAPTER 3

Contributing

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

CHAPTER 4

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-tsl2591 --
↳library_location .
```

4.1 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.

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/tsl2591_simpletest.py

```
1 # Simple demo of the TSL2591 sensor. Will print the detected light value
2 # every second.
3 import time
4
5 import board
6 import busio
7
8 import adafruit_tsl2591
9
10 # Initialize the I2C bus.
11 i2c = busio.I2C(board.SCL, board.SDA)
12
13 # Initialize the sensor.
14 sensor = adafruit_tsl2591.TSL2591(i2c)
15
16 # You can optionally change the gain and integration time:
17 #sensor.gain = adafruit_tsl2591.GAIN_LOW (1x gain)
18 #sensor.gain = adafruit_tsl2591.GAIN_MED (25x gain, the default)
19 #sensor.gain = adafruit_tsl2591.GAIN_HIGH (428x gain)
20 #sensor.gain = adafruit_tsl2591.GAIN_MAX (9876x gain)
21 #sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_100MS (100ms, default)
22 #sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_200MS (200ms)
23 #sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_300MS (300ms)
24 #sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_400MS (400ms)
25 #sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_500MS (500ms)
26 #sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_600MS (600ms)
27
```

(continues on next page)

(continued from previous page)

```

28 # Read the total lux, IR, and visible light levels and print it every second.
29 while True:
30     # Read and calculate the light level in lux.
31     lux = sensor.lux
32     print('Total light: {0}lux'.format(lux))
33     # You can also read the raw infrared and visible light levels.
34     # These are unsigned, the higher the number the more light of that type.
35     # There are no units like lux.
36     # Infrared levels range from 0-65535 (16-bit)
37     infrared = sensor.infrared
38     print('Infrared light: {0}'.format(infrared))
39     # Visible-only levels range from 0-2147483647 (32-bit)
40     visible = sensor.visible
41     print('Visible light: {0}'.format(visible))
42     # Full spectrum (visible + IR) also range from 0-2147483647 (32-bit)
43     full_spectrum = sensor.full_spectrum
44     print('Full spectrum (IR + visible) light: {0}'.format(full_spectrum))
45     time.sleep(1.0)

```

5.2 adafruit_ts12591

CircuitPython module for the TSL2591 precision light sensor. See examples/simpletest.py for a demo of the usage.

- Author(s): Tony DiCola

5.2.1 Implementation Notes

Hardware:

- Adafruit TSL2591 High Dynamic Range Digital Light Sensor (Product ID: 1980)

Software and Dependencies:

- Adafruit CircuitPython firmware 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

adafruit_ts12591.**GAIN_HIGH** = 32
High gain (428x)

adafruit_ts12591.**GAIN_LOW** = 0
Low gain (1x)

adafruit_ts12591.**GAIN_MAX** = 48
Max gain (9876x)

adafruit_ts12591.**GAIN_MED** = 16
Medium gain (25x)

adafruit_ts12591.**INTEGRATIONTIME_100MS** = 0
100 millis

adafruit_ts12591.**INTEGRATIONTIME_200MS** = 1
200 millis

adafruit_tsl2591.**INTEGRATIONTIME_300MS** = 2
300 millis

adafruit_tsl2591.**INTEGRATIONTIME_400MS** = 3
400 millis

adafruit_tsl2591.**INTEGRATIONTIME_500MS** = 4
500 millis

adafruit_tsl2591.**INTEGRATIONTIME_600MS** = 5
600 millis

class adafruit_tsl2591.**TSL2591** (*i2c, address=41*)

TSL2591 high precision light sensor. :param busio.I2C i2c: The I2C bus connected to the sensor :param int address: The I2C address of the sensor. If not specified the sensor default will be used.

disable ()

Disable the device and go into low power mode.

enable ()

Put the device in a fully powered enabled mode.

full_spectrum

Read the full spectrum (IR + visible) light and return its value as a 32-bit unsigned number.

gain

Get and set the gain of the sensor. Can be a value of:

- GAIN_LOW (1x)
- GAIN_MED (25x)
- GAIN_HIGH (428x)
- GAIN_MAX (9876x)

infrared

Read the infrared light and return its value as a 16-bit unsigned number.

integration_time

Get and set the integration time of the sensor. Can be a value of:

- INTEGRATIONTIME_100MS (100 millis)
- INTEGRATIONTIME_200MS (200 millis)
- INTEGRATIONTIME_300MS (300 millis)
- INTEGRATIONTIME_400MS (400 millis)
- INTEGRATIONTIME_500MS (500 millis)
- INTEGRATIONTIME_600MS (600 millis)

lux

Read the sensor and calculate a lux value from both its infrared and visible light channels.

raw_luminosity

Read the raw luminosity from the sensor (both IR + visible and IR only channels) and return a 2-tuple of those values. The first value is IR + visible luminosity (channel 0) and the second is the IR only (channel 1). Both values are 16-bit unsigned numbers (0-65535).

visible

Read the visible light and return its value as a 32-bit unsigned number.

CHAPTER 6

Indices and tables

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

a

`adafruit_tsl2591`, 12

A

adafruit_tsl2591 (module), 12

D

disable() (adafruit_tsl2591.TSL2591 method), 13

E

enable() (adafruit_tsl2591.TSL2591 method), 13

F

full_spectrum (adafruit_tsl2591.TSL2591 attribute), 13

G

gain (adafruit_tsl2591.TSL2591 attribute), 13

GAIN_HIGH (in module adafruit_tsl2591), 12

GAIN_LOW (in module adafruit_tsl2591), 12

GAIN_MAX (in module adafruit_tsl2591), 12

GAIN_MED (in module adafruit_tsl2591), 12

I

infrared (adafruit_tsl2591.TSL2591 attribute), 13

integration_time (adafruit_tsl2591.TSL2591 attribute),
13

INTEGRATIONTIME_100MS (in module
adafruit_tsl2591), 12

INTEGRATIONTIME_200MS (in module
adafruit_tsl2591), 12

INTEGRATIONTIME_300MS (in module
adafruit_tsl2591), 12

INTEGRATIONTIME_400MS (in module
adafruit_tsl2591), 13

INTEGRATIONTIME_500MS (in module
adafruit_tsl2591), 13

INTEGRATIONTIME_600MS (in module
adafruit_tsl2591), 13

L

lux (adafruit_tsl2591.TSL2591 attribute), 13

R

raw_luminosity (adafruit_tsl2591.TSL2591 attribute), 13

T

TSL2591 (class in adafruit_tsl2591), 13

V

visible (adafruit_tsl2591.TSL2591 attribute), 13