

---

# **AdafruitTSL2591 Library Documentation**

*Release 1.0*

**Tony DiCola**

**May 07, 2020**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Installing from PyPI</b>	<b>5</b>
<b>3</b>	<b>Usage Example</b>	<b>7</b>
<b>4</b>	<b>Contributing</b>	<b>9</b>
<b>5</b>	<b>Documentation</b>	<b>11</b>
<b>6</b>	<b>Table of Contents</b>	<b>13</b>
6.1	Simple test .....	13
6.2	adafruit_tsl2591 .....	14
6.2.1	Implementation Notes .....	14
<b>7</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>
	<b>Index</b>	<b>21</b>



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

---

### 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-tsl2591
```

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

```
sudo pip3 install adafruit-circuitpython-tsl2591
```

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-tsl2591
```



## CHAPTER 3

---

### Usage Example

---

See `examples/tsl2591_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](#).





## 6.1 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)
```

## 6.2 adafruit\_ts12591

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

- Author(s): Tony DiCola

### 6.2.1 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](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. Note: lux is not calibrated!

**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 7

---

## Indices and tables

---

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



**a**

adafruit\_tsl2591, 14





**A**

`adafruit_tsl2591` (module), 14

**D**

`disable()` (*adafruit\_tsl2591.TSL2591* method), 15

**E**

`enable()` (*adafruit\_tsl2591.TSL2591* method), 15

**F**

`full_spectrum` (*adafruit\_tsl2591.TSL2591* attribute), 15

**G**

`gain` (*adafruit\_tsl2591.TSL2591* attribute), 15

`GAIN_HIGH` (in module *adafruit\_tsl2591*), 14

`GAIN_LOW` (in module *adafruit\_tsl2591*), 14

`GAIN_MAX` (in module *adafruit\_tsl2591*), 14

`GAIN_MED` (in module *adafruit\_tsl2591*), 14

**I**

`infrared` (*adafruit\_tsl2591.TSL2591* attribute), 15

`integration_time` (*adafruit\_tsl2591.TSL2591* attribute), 15

`INTEGRATIONTIME_100MS` (in module *adafruit\_tsl2591*), 14

`INTEGRATIONTIME_200MS` (in module *adafruit\_tsl2591*), 14

`INTEGRATIONTIME_300MS` (in module *adafruit\_tsl2591*), 14

`INTEGRATIONTIME_400MS` (in module *adafruit\_tsl2591*), 15

`INTEGRATIONTIME_500MS` (in module *adafruit\_tsl2591*), 15

`INTEGRATIONTIME_600MS` (in module *adafruit\_tsl2591*), 15

**L**

`lux` (*adafruit\_tsl2591.TSL2591* attribute), 15

**R**

`raw_luminosity` (*adafruit\_tsl2591.TSL2591* attribute), 15

**T**

`TSL2591` (class in *adafruit\_tsl2591*), 15

**V**

`visible` (*adafruit\_tsl2591.TSL2591* attribute), 15