AdafruitTSL2591 Library Documentation

Release 1.0

Tony DiCola

Mar 02, 2021

Contents

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.

CHAPTER 6

Table of Contents

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/tsl2591_simpletest.py

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

# Simple demo of the TSL2591 sensor.  Will print the detected light value
# every second.
import time

import board
import busio

import adafruit_tsl2591

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

# Initialize the sensor.
sensor = adafruit_tsl2591.TSL2591(i2c)

# You can optionally change the gain and integration time:
# sensor.gain = adafruit_tsl2591.GAIN_LOW (1x gain)
# sensor.gain = adafruit_tsl2591.GAIN_MED (25x gain, the default)
# sensor.gain = adafruit_tsl2591.GAIN_HIGH (428x gain)
# sensor.gain = adafruit_tsl2591.GAIN_MAX (9876x gain)
# sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_100MS (100ms, default)
# sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_200MS (200ms)
# sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_300MS (300ms)
# sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_400MS (400ms)
```

(continues on next page)

(continued from previous page)

```
# sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_500MS (500ms)
# sensor.integration_time = adafruit_tsl2591.INTEGRATIONTIME_600MS (600ms)

# Read the total lux, IR, and visible light levels and print it every second.
while True:
    # Read and calculate the light level in lux.
    lux = sensor.lux
    print("Total light: {0}lux".format(lux))
    # You can also read the raw infrared and visible light levels.
    # These are unsigned, the higher the number the more light of that type.
    # There are no units like lux.
    # Infrared levels range from 0-65535 (16-bit)
    infrared = sensor.infrared
    print("Infrared light: {0}".format(infrared))
    # Visible-only levels range from 0-2147483647 (32-bit)
    visible = sensor.visible
    print("Visible light: {0}".format(visible))
    # Full spectrum (visible + IR) also range from 0-2147483647 (32-bit)
    full_spectrum = sensor.full_spectrum
    print("Full spectrum (IR + visible) light: {0}".format(full_spectrum))
    time.sleep(1.0)
```

6.2 adafruit_tsl2591

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

adafruit_tsl2591.**GAIN_HIGH = 32**
High gain (428x)

adafruit_tsl2591.**GAIN_LOW = 0**
Low gain (1x)

adafruit_tsl2591.**GAIN_MAX = 48**
Max gain (9876x)

adafruit_tsl2591.**GAIN_MED = 16**
Medium gain (25x)

adafruit_tsl2591.**INTEGRATIONTIME_100MS = 0**
100 millis

adafruit_tsl2591.**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

Python Module Index

a

Index