Adafruit CIRCUITPYTHON$_{TSL2561}$ Library Documentation

Release 1.0

Carter Nelson

Feb 07, 2019

Contents

CircuitPython driver for TSL2561 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

```
>>> import board
>>> import busio
>>> i2c = busio.I2C(board.SCL, board.SDA)
>>> import adafruit_tsl2561
>>> tsl = adafruit_tsl2561.TSL2561(i2c)
>>> tsl.lux
3294.37
```

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-tls2561 --
↪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.

CHAPTER 5

Table of Contents

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/tsl2561_simpletest.py

```
import board
import busio
import adafruit_tsl2561

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

# Create the TSL2561 instance, passing in the I2C bus
tsl = adafruit_tsl2561.TSL2561(i2c)

# Print chip info
print("Chip ID = {}".format(tsl.chip_id))
print("Enabled = {}".format(tsl.enabled))
print("Gain = {}".format(tsl.gain))
print("Integration time = {}".format(tsl.integration_time))

print("Configuring TSL2561...")

# Enable the light sensor
tsl.enabled = True

# Set gain 0=1x, 1=16x
tsl.gain = 0

# Set integration time (0=13.7ms, 1=101ms, 2=402ms, or 3=manual)
tsl.integration_time = 1

```

(continues on next page)

(continued from previous page)

```
print("Getting readings...")

# Get raw (luminosity) readings individually
broadband = tsl.broadband
infrared = tsl.infrared

# Get raw (luminosity) readings using tuple unpacking
#broadband, infrared = tsl.luminosity

# Get computed lux value (tsl.lux can return None or a float)
lux = tsl.lux

# Print results
print("Enabled = {}".format(tsl.enabled))
print("Gain = {}".format(tsl.gain))
print("Integration time = {}".format(tsl.integration_time))
print("Broadband = {}".format(broadband))
print("Infrared = {}".format(infrared))
if lux is not None:
    print("Lux = {}".format(lux))
else:
    print("Lux value is None. Possible sensor underrange or overrange.")

# Disble the light sensor (to save power)
tsl.enabled = False
```

5.2 adafruit_tsl2561

CircuitPython driver for TSL2561 Light Sensor.

- Author(s): Carter Nelson

5.2.1 Implementation Notes

Hardware:

- Adafruit TSL2561 Digital Luminosity/Lux/Light Sensor Breakout (Product ID: 439)
- Adafruit STEMMA - TSL2561 Digital Lux / Light Sensor (Product ID: 3611)
- Adafruit Flora Lux Sensor - TSL2561 Light Sensor (Product ID: 1246)

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

class adafruit_tsl2561.**TSL2561**(*i2c*, *address=57*)

Class which provides interface to TSL2561 light sensor.

broadband

The broadband channel value.

chip_id

A tuple containing the part number and the revision number.

clear_interrupt()
Clears any pending interrupt.

cycles
The number of integration cycles for which an out of bounds value must persist to cause an interrupt.

enabled
The state of the sensor.

gain
The gain. 0:1x, 1:16x.

infrared
The infrared channel value.

integration_time
The integration time. 0:13.7ms, 1:101ms, 2:402ms, or 3:manual

interrupt_mode
The interrupt mode selection.

Mode	Description
0	Interrupt output disabled
1	Level Interrupt
2	SMBAlert compliant
3	Test Mode

luminosity
The overall luminosity as a tuple containing the broadband channel and the infrared channel value.

lux
The computed lux value or None when value is not computable.

threshold_high
The upper light interrupt threshold level.

threshold_low
The low light interrupt threshold level.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index