Adafruit TCS34725 Library Documentation

Release 1.0

Tony DiCola

Mar 20, 2020

Contents

1 Dependencies **3**

2 Installing from PyPI **5**

3 Usage Example **7**

4 Contributing **9**

5 Documentation **11**

6 Table of Contents **13**

6.1 Simple test 13

6.2 `adafruit_tcs34725` 14

6.2.1 Implementation Notes 14

7 Indices and tables **17**

Python Module Index **19**

Index **21**

CircuitPython module for the TCS34725 color 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-tcs34725
```

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

```
sudo pip3 install adafruit-circuitpython-tcs34725
```

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-tcs34725
```

CHAPTER 3

Usage Example

See examples/tcs34725_simpletest.py for an example of the module's 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/tcs34725_simpletest.py

```
# Simple demo of the TCS34725 color sensor.
# Will detect the color from the sensor and print it out every second.
import time

import board
import busio

import adafruit_tcs34725


# Initialize I2C bus and sensor.
i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_tcs34725.TCS34725(i2c)

# Main loop reading color and printing it every second.
while True:
    # Read the color temperature and lux of the sensor too.
    temp = sensor.color_temperature
    lux = sensor.lux
    print("Temperature: {0}K Lux: {1}".format(temp, lux))
    # Delay for a second and repeat.
    time.sleep(1.0)
```

6.2 adafruit_tcs34725

CircuitPython module for the TCS34725 color sensor. Ported from the micropython-adafruit-tcs34725 module by Radomir Dopieralski: https://github.com/adafruit/micropython-adafruit-tcs34725

See examples/tcs34725_simpletest.py for an example of the usage.

- Author(s): Tony DiCola, Carter Nelson

6.2.1 Implementation Notes

Hardware:

- Adafruit RGB Color Sensor with IR filter and White LED - TCS34725 (Product ID: 1334)
- Flora Color Sensor with White Illumination LED - TCS34725 (Product ID: 1356)

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_tcs34725.**TCS34725**(*i2c*, *address=41*)

Driver for the TCS34725 color sensor.

active

The active state of the sensor. Boolean value that will enable/activate the sensor with a value of True and disable with a value of False.

color

Read the RGB color detected by the sensor. Returns an int with 8 bits per channel. Examples: Red = 16711680 (0xff0000), Green = 65280 (0x00ff00), Blue = 255 (0x0000ff), SlateGray = 7372944 (0x708090)

color_raw

Read the raw RGBC color detected by the sensor. Returns a 4-tuple of 16-bit red, green, blue, clear component byte values (0-65535).

color_rgb_bytes

Read the RGB color detected by the sensor. Returns a 3-tuple of red, green, blue component values as bytes (0-255).

color_temperature

The color temperature in degrees Kelvin.

cycles

The persistence cycles of the sensor.

gain

The gain of the sensor. Should be a value of 1, 4, 16, or 60.

glass_attenuation

The Glass Attenuation (FA) factor used to compensate for lower light levels at the device due to the possible presence of glass. The GA is the inverse of the glass transmissivity (T), so GA = 1/T. A transmissivity of 50% gives GA = 1 / 0.50 = 2. If no glass is present, use GA = 1. See Application Note: DN40-Rev 1.0 – Lux and CCT Calculations using ams Color Sensors for more details.

integration_time

The integration time of the sensor in milliseconds.

interrupt
True if the interrupt is set. Can be set to False (and only False) to clear the interrupt.

lux
The lux value computed from the color channels.

max_value
The minimum threshold value (AIHT register) of the sensor as a 16-bit unsigned value.

min_value
The minimum threshold value (AILT register) of the sensor as a 16-bit unsigned value.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

adafruit_tcs34725, 13

Index

A

`active` (*adafruit_tcs34725.TCS34725 attribute*), 14
`adafruit_tcs34725` (*module*), 13

C

`color` (*adafruit_tcs34725.TCS34725 attribute*), 14
`color_raw` (*adafruit_tcs34725.TCS34725 attribute*), 14
`color_rgb_bytes` (*adafruit_tcs34725.TCS34725 attribute*), 14
`color_temperature` (*adafruit_tcs34725.TCS34725 attribute*), 14
`cycles` (*adafruit_tcs34725.TCS34725 attribute*), 14

G

`gain` (*adafruit_tcs34725.TCS34725 attribute*), 14
`glass_attenuation` (*adafruit_tcs34725.TCS34725 attribute*), 14

I

`integration_time` (*adafruit_tcs34725.TCS34725 attribute*), 14
`interrupt` (*adafruit_tcs34725.TCS34725 attribute*), 14

L

`lux` (*adafruit_tcs34725.TCS34725 attribute*), 15

M

`max_value` (*adafruit_tcs34725.TCS34725 attribute*), 15
`min_value` (*adafruit_tcs34725.TCS34725 attribute*), 15

T

`TCS34725` (*class in adafruit_tcs34725*), 14