
AdafruitAS726x Library Documentation

Release 1.0

Dean Miller

Jun 07, 2021

Contents

1	Installation and Dependencies	3
1.1	Installing from PyPI	3
2	Contributing	5
3	Documentation	7
4	Table of Contents	9
4.1	Simple test	9
4.2	adafruit_as726x	10
4.2.1	Implementation Notes	10
5	Indices and tables	15
	Python Module Index	17
	Index	19

Driver for the AS726x spectral sensors

Installation and Dependencies

This driver depends on:

- [Adafruit CircuitPython](#)
- [Bus Device](#)
- [Register](#)

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the Adafruit library and driver bundle.

1.1 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-as726x
```

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

```
sudo pip3 install adafruit-circuitpython-as726x
```

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-as726x
```


CHAPTER 2

Contributing

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

CHAPTER 3

Documentation

For information on building library documentation, please check out [this guide](#).

4.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/as726x_simpletest.py

```
1  # SPDX-FileCopyrightText: 2020 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  import time
5  import board
6
7  # for I2C use:
8  from adafruit_as726x import AS726x_I2C
9
10 # for UART use:
11 # from adafruit_as726x import AS726x_UART
12
13 # maximum value for sensor reading
14 max_val = 16000
15
16 # max number of characters in each graph
17 max_graph = 80
18
19
20 def graph_map(x):
21     return min(int(x * max_graph / max_val), max_graph)
22
23
24 # for I2C use:
25 i2c = board.I2C()
26 sensor = AS726x_I2C(i2c)
27
```

(continues on next page)

(continued from previous page)

```

28 # for UART use:
29 # uart = board.UART()
30 # sensor = AS726x_UART(uart)
31
32 sensor.conversion_mode = sensor.MODE_2
33
34 while True:
35     # Wait for data to be ready
36     while not sensor.data_ready:
37         time.sleep(0.1)
38
39     # plot plot the data
40     print("\n")
41     print("V: " + graph_map(sensor.violet) * "=")
42     print("B: " + graph_map(sensor.blue) * "=")
43     print("G: " + graph_map(sensor.green) * "=")
44     print("Y: " + graph_map(sensor.yellow) * "=")
45     print("O: " + graph_map(sensor.orange) * "=")
46     print("R: " + graph_map(sensor.red) * "=")
47
48     time.sleep(1)

```

4.2 adafruit_as726x

Driver for the AS726x spectral sensors

- Author(s): Dean Miller

4.2.1 Implementation Notes

Hardware:

- Adafruit AS7262 6-Channel Visible Light / Color Sensor Breakout (Product ID: 3779)

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://circuitpython.org/downloads>
- Adafruit's Bus Device library: https://github.com/adafruit/Adafruit_CircuitPython_BusDevice

class adafruit_as726x.**AS726x**

AS726x spectral sensor base class.

MODE_0 = 0

Continuously gather samples of violet, blue, green and yellow. Orange and red are skipped and read zero.

MODE_1 = 1

Continuously gather samples of green, yellow, orange and red. Violet and blue are skipped and read zero.

MODE_2 = 2

Continuously gather samples of all colors

ONE_SHOT = 3

Gather a single sample of all colors and then stop

driver_led

True when the driver LED is on. False otherwise.

indicator_led

True when the indicator LED is on. False otherwise.

driver_led_current

The current limit for the driver LED in milliamps. One of:

- 12.5 mA
- 25 mA
- 50 mA
- 100 mA

indicator_led_current

The current limit for the indicator LED in milliamps. One of:

- 1 mA
- 2 mA
- 4 mA
- 8 mA

conversion_mode

The conversion mode. One of:

- *MODE_0*
- *MODE_1*
- *MODE_2*
- *ONE_SHOT*

gain

The gain for the sensor

integration_time

The integration time in milliseconds between 2.8 and 714 ms

start_measurement ()

Begin a measurement.

This will set the device to One Shot mode and values will not change after *data_ready* until *start_measurement ()* is called again or the *conversion_mode ()* is changed.

read_channel (channel)

Read an individual sensor channel

read_calibrated_value (channel)

Read a calibrated sensor channel

data_ready

True if the sensor has data ready to be read, False otherwise

temperature

The temperature of the device in Celsius

violet

Calibrated violet (450nm) value

blue

Calibrated blue (500nm) value

green

Calibrated green (550nm) value

yellow

Calibrated yellow (570nm) value

orange

Calibrated orange (600nm) value

red

Calibrated red (650nm) value

raw_violet

Raw violet (450nm) 16-bit value

raw_blue

Raw blue (500nm) 16-bit value

raw_green

Raw green (550nm) 16-bit value

raw_yellow

Raw yellow (570nm) 16-bit value

raw_orange

Raw orange (600nm) 16-bit value

raw_red

Raw red (650nm) 16-bit value

class adafruit_as726x.**AS726x_I2C** (*i2c_bus*, *address=73*)

AS726x spectral sensor via I2C.

Parameters

- **i2c_bus** (*I2C*) – The I2C bus the AS726x is connected to
- **address** (*int*) – The I2C device address. Defaults to 0x49

Quickstart: Importing and using the AS726x

Here is an example of using the *AS726x_I2C* class. First you will need to import the libraries to use the sensor

```
import board
from adafruit_as726x import AS726x_I2C
```

Once this is done you can define your *board.I2C* object and define your sensor object

```
i2c = board.I2C() # uses board.SCL and board.SDA
sensor = AS726x_I2C(i2c)
```

Now you have access to the different color attributes

```
violet = sensor.violet
blue = sensor.blue
green = sensor.green
yellow = sensor.yellow
orange = sensor.orange
red = sensor.red
```

class adafruit_as726x.**AS726x_UART** (*uart*)

AS726x spectral sensor via UART.

Parameters `uart` (*UART*) – The UART connected to the sensor

Quickstart: Importing and using the AS726x

Here is an example of using the `AS726x_I2C` class. First you will need to import the libraries to use the sensor

```
import board
from adafruit_as726x import AS726x_UART
```

Once this is done you can define your `board.UART` object and define your sensor object

```
uart = board.UART() # uses board.SCL and board.SDA
sensor = AS726x_UART(uart)
```

Now you have access to the different color attributes

```
violet = sensor.violet
blue = sensor.blue
green = sensor.green
yellow = sensor.yellow
orange = sensor.orange
red = sensor.red
```

read_channel (*channel*)

Read an individual sensor channel

read_calibrated_value (*channel*)

Read a calibrated sensor channel

CHAPTER 5

Indices and tables

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

a

`adafruit_as726x`, 10

A

adafruit_as726x (*module*), 10
AS726x (*class in adafruit_as726x*), 10
AS726x_I2C (*class in adafruit_as726x*), 12
AS726x_UART (*class in adafruit_as726x*), 12

B

blue (*adafruit_as726x.AS726x attribute*), 11

C

conversion_mode (*adafruit_as726x.AS726x attribute*), 11

D

data_ready (*adafruit_as726x.AS726x attribute*), 11
driver_led (*adafruit_as726x.AS726x attribute*), 10
driver_led_current (*adafruit_as726x.AS726x attribute*), 11

G

gain (*adafruit_as726x.AS726x attribute*), 11
green (*adafruit_as726x.AS726x attribute*), 11

I

indicator_led (*adafruit_as726x.AS726x attribute*), 10
indicator_led_current (*adafruit_as726x.AS726x attribute*), 11
integration_time (*adafruit_as726x.AS726x attribute*), 11

M

MODE_0 (*adafruit_as726x.AS726x attribute*), 10
MODE_1 (*adafruit_as726x.AS726x attribute*), 10
MODE_2 (*adafruit_as726x.AS726x attribute*), 10

O

ONE_SHOT (*adafruit_as726x.AS726x attribute*), 10

orange (*adafruit_as726x.AS726x attribute*), 12

R

raw_blue (*adafruit_as726x.AS726x attribute*), 12
raw_green (*adafruit_as726x.AS726x attribute*), 12
raw_orange (*adafruit_as726x.AS726x attribute*), 12
raw_red (*adafruit_as726x.AS726x attribute*), 12
raw_violet (*adafruit_as726x.AS726x attribute*), 12
raw_yellow (*adafruit_as726x.AS726x attribute*), 12
read_calibrated_value ()
(*adafruit_as726x.AS726x method*), 11
read_calibrated_value ()
(*adafruit_as726x.AS726x_UART method*), 13
read_channel () (*adafruit_as726x.AS726x method*), 11
read_channel () (*adafruit_as726x.AS726x_UART method*), 13
red (*adafruit_as726x.AS726x attribute*), 12

S

start_measurement () (*adafruit_as726x.AS726x method*), 11

T

temperature (*adafruit_as726x.AS726x attribute*), 11

V

violet (*adafruit_as726x.AS726x attribute*), 11

Y

yellow (*adafruit_as726x.AS726x attribute*), 12