Adafruit MicroPython TCS24725 Library Documentation

Release 1.0

Radomir Dopieralski

Jun 13, 2017

Contents

Contents:

tcs34725 module

TCS34725

class `tcs34725.`**`TCS34725`**(*i2c*[, *address*])

The basic class for handling the communication with the sensor.

The `i2c` parameter is an initialized I²C bus, and the optional address specifies which sensor to connect to, if you have more than one and have changed their addresses with the `Addr` pin.

May raise a `RuntimeError` if connected to a wrong model of sensor.

`active`([*True|False*])

Get or set the power status of the sensor.

If called without any parameters, returns the current status (`True` for active, `False` for inactive). If called with a parameter, sets the status.

The functions that require the sensor to be active will activate it for the time necessary, and then return it to the previous state.

`gain`([*1|4|16|60*])

Get or set the measurement gain of the sensor.

`integration_time`([*value*])

Get or set the integraton time for measurements in milliseconds, from 2.4ms to 614.4ms in steps of 2.4ms.

`sensor_id`()

Return the internal ID of the sensor, describing its model and revision number.

`read`([*raw*])

Read a measurement from the sensor.

If `raw` is `False` (the deafult), returns a tuple of floating point values denoting color temperature in kelvins and luminosity in lux. If it's `True`, returns a 4-tuple of integer numbers, corresponding to the readings of the red, green, blue and clear channels.

If at the moment of calling this method, the sensor is inactive, it will be activated, the method will block for a time necessary for the sensor to make the measurement, and then return and deactivate the sensor. If the sensor was active, the library may still wait for the sensor to complete the measurement, if it's not ready.

threshold (*[cycles] [, min_value] [, max_value]*)
> Get or set the interrupt settings.

> If called without arguments, returns a triple of values.

> The `cycles` argument specifies how many cycles are needed to trigger the interrupt. When set to 0, each measurement will do it. If set to 1, 2, 3, or 5-60 (in increments of 5), the interrupt will be triggered only when so many measurements in a row fall out of bounds. When set to -1, interrupts are disabled.

> The `min_value` and `max_value` specify the window for the clear channel to fit in.

> Whenever the interrupt is activated, the `Int` pin of the sensor will be pulled low and stay in that state until cleared. You can connect it to one of the pins and set a pin interrupt to detect that event.

> Note that for the interrupt pin to work, the sensor must stay active.

interrupt ($[$*False*$]$)
> Get or set the status of interrupt.

> Without arguments, returns `True` if an interrupt is triggered, otherwise returns `False`.

> When `False` is passed as an argument, clears the interrupt.

Usage Examples

Connect your sensor in following way:

- `3vo 3V`

- `gnd gnd`

- `sda gpio4`

- `scl gpio5`

- `int` any other pin (optional)

Now, to make basic measurement:

```python
import tcs34725
from machine import I2C, Pin
i2c = I2C(Pin(5), Pin(4))
sensor = tcs34725.TCS34725(i2c)
print(sensor.read())
```

To perform continuous measurement:

```python
import time
sensor.active(True)
time.sleep_ms(500)
while True:
    print(sensor.read())
    time.sleep_ms(20)
```

To change the gain and integration time:

```python
sensor.gain(16)
sensor.integration_time(402)
```

To use the interrupt pin (connected to `gpio0` here):

```
def handler(pin):
    print("interrupt!")
    sensor.interrupt(False)

int_pin = Pin(0, Pin.IN, Pin.PULL_UP)
int_pin.irq(handler=handler, trigger=Pin.IRQ_FALLING)
sensor.active(True)
sensor.threshold(1, 10000, 30000)
```

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

t

Index

A

active() (tcs34725.TCS34725 method),

G

gain() (tcs34725.TCS34725 method),

I

integration_time() (tcs34725.TCS34725 method),
interrupt() (tcs34725.TCS34725 method),

R

read() (tcs34725.TCS34725 method),

S

sensor_id() (tcs34725.TCS34725 method),

T

TCS34725 (class in tcs34725),
tcs34725 (module),
threshold() (tcs34725.TCS34725 method),