
AdafruitMAX31856 Library Documentation

Release 1.0

Bryan Siepert

Apr 26, 2021

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	Thresholds and Fault example	13
6.3	MAX31856	15
6.3.1	Implementation Notes	15
7	Indices and tables	17
	Python Module Index	19
	Index	21

A CircuitPython driver for the MAX31856 Universal Thermocouple Amplifier

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-max31856
```

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

```
sudo pip3 install adafruit-circuitpython-max31856
```

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-max31856
```


CHAPTER 3

Usage Example

```
import board
import digitalio
import adafruit_max31856

# Create sensor object, communicating over the board's default SPI bus
spi = board.SPI()

# allocate a CS pin and set the direction
cs = digitalio.DigitalInOut(board.D5)
cs.direction = digitalio.Direction.OUTPUT

# create a thermocouple object with the above
thermocouple = adafruit_max31856.MAX31856(spi, cs)

# print the temperature!
print(thermocouple.temperature)
```


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](#).

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/max31856_simpletest.py

```
1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  import board
5  import digitalio
6  import adafruit_max31856
7
8  # Create sensor object, communicating over the board's default SPI bus
9  spi = board.SPI()
10
11 # allocate a CS pin and set the direction
12 cs = digitalio.DigitalInOut(board.D5)
13 cs.direction = digitalio.Direction.OUTPUT
14
15 # create a thermocouple object with the above
16 thermocouple = adafruit_max31856.MAX31856(spi, cs)
17
18 # print the temperature!
19 print(thermocouple.temperature)
```

6.2 Thresholds and Fault example

Example showing how to use thresholds

Listing 2: examples/max31856_thresholds_and_faults.py

```

1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  import time
5  import board
6  import digitalio
7  import adafruit_max31856
8
9  # Create sensor object, communicating over the board's default SPI bus
10 spi = board.SPI()
11
12 # allocate a CS pin and set the direction
13 cs = digitalio.DigitalInOut(board.D0)
14 cs.direction = digitalio.Direction.OUTPUT
15
16 # create a thermocouple object with the above
17 thermocouple = adafruit_max31856.MAX31856(spi, cs)
18
19 # set the temperature thresholds for the thermocouple and cold junction
20 thermocouple.temperature_thresholds = (-1.5, 30.8)
21 thermocouple.reference_temperature_thresholds = (-1.0, 30.5)
22 current_faults = {}
23 current_cj_thresholds = (0, 0)
24 current_temp_thresholds = (0, 0)
25 print(thermocouple.reference_temperature_thresholds)
26 while True:
27     current_temp_thresholds = thermocouple.temperature_thresholds
28     current_cj_thresholds = thermocouple.reference_temperature_thresholds
29     current_faults = thermocouple.fault
30     print(
31         "Temps:      %.2f :: cj: %.2f"
32         % (thermocouple.temperature, thermocouple.reference_temperature)
33     )
34     print("Thresholds:")
35     print("Temp low: %.2f high: %.2f" % current_temp_thresholds)
36     print("CJ low:   %.2f high: %.2f" % current_cj_thresholds)
37     print("")
38     print("Faults:")
39     print(
40         "Temp Hi:    %s | CJ Hi:    %s"
41         % (current_faults["tc_high"], current_faults["cj_high"])
42     )
43     print(
44         "Temp Low:   %s | CJ Low:   %s"
45         % (current_faults["tc_low"], current_faults["cj_low"])
46     )
47     print(
48         "Temp Range: %s | CJ Range: %s"
49         % (current_faults["tc_range"], current_faults["cj_range"])
50     )
51     print("")
52     print(
53         "Open Circuit: %s Voltage Over/Under: %s"
54         % (current_faults["open_tc"], current_faults["voltage"])
55     )

```

(continues on next page)

(continued from previous page)

```

56     print("")
57
58     time.sleep(1.0)

```

6.3 MAX31856

CircuitPython module for the MAX31856 Universal Thermocouple Amplifier. See `examples/simpletest.py` for an example of the usage.

- Author(s): Bryan Siepert

6.3.1 Implementation Notes

Hardware:

- Adafruit [Universal Thermocouple Amplifier MAX31856 Breakout](#) (Product ID: 3263)

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_max31856.MAX31856` (*spi, cs, thermocouple_type=3*)
 Driver for the MAX31856 Universal Thermocouple Amplifier

Parameters

- **spi** (*SPI*) – The SPI bus the MAX31856 is connected to.
- **cs** (*Pin*) – The pin used for the CS signal.
- **thermocouple_type** (*ThermocoupleType*) – The type of thermocouple. Default is Type K.

Quickstart: Importing and using the MAX31856

Here is an example of using the `MAX31856` class. First you will need to import the libraries to use the sensor

```

import board
from digitalio import DigitalInOut, Direction
import adafruit_max31856

```

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

```

spi = board.SPI()
cs = digitalio.DigitalInOut(board.D5) # Chip select of the MAX31856,
↪board.
sensor = adafruit_max31856.MAX31856(spi, cs)

```

Now you have access to the `temperature` attribute

```

temperature = sensor.temperature

```

fault

A dictionary with the status of each fault type where the key is the fault type and the value is a bool if the fault is currently active

Key	Fault type
"cj_range"	Cold junction range fault
"tc_range"	Thermocouple range fault
"cj_high"	Cold junction high threshold fault
"cj_low"	Cold junction low threshold fault
"tc_high"	Thermocouple high threshold fault
"tc_low"	Thermocouple low threshold fault
"voltage"	Over/under voltage fault
"open_tc"	Thermocouple open circuit fault

reference_temperature

The temperature of the cold junction in degrees Celsius. (read-only)

reference_temperature_thresholds

The cold junction's low and high temperature thresholds as a (low_temp, high_temp) tuple

temperature

The temperature of the sensor and return its value in degrees Celsius. (read-only)

temperature_thresholds

The thermocouple's low and high temperature thresholds as a (low_temp, high_temp) tuple

class adafruit_max31856.**ThermocoupleType**

An enum-like class representing the different types of thermocouples that the MAX31856 can use. The values can be referenced like ThermocoupleType.K or ThermocoupleType.S Possible values are

- ThermocoupleType.B
- ThermocoupleType.E
- ThermocoupleType.J
- ThermocoupleType.K
- ThermocoupleType.N
- ThermocoupleType.R
- ThermocoupleType.S
- ThermocoupleType.T

CHAPTER 7

Indices and tables

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

a

adafruit_max31856, [15](#)

A

`adafruit_max31856` (*module*), [15](#)

F

`fault` (*adafruit_max31856.MAX31856 attribute*), [15](#)

M

`MAX31856` (*class in adafruit_max31856*), [15](#)

R

`reference_temperature`
(*adafruit_max31856.MAX31856 attribute*),
[16](#)

`reference_temperature_thresholds`
(*adafruit_max31856.MAX31856 attribute*),
[16](#)

T

`temperature` (*adafruit_max31856.MAX31856 attribute*), [16](#)

`temperature_thresholds`
(*adafruit_max31856.MAX31856 attribute*),
[16](#)

`ThermocoupleType` (*class in adafruit_max31856*), [16](#)