

---

# **Adafruit MAX31855 Library Documentation**

***Release 1.0***

**Radomir Dopieralski**

**Mar 02, 2021**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Installing from PyPI</b>	<b>5</b>
<b>3</b>	<b>Usage Example</b>	<b>7</b>
<b>4</b>	<b>Contributing</b>	<b>9</b>
<b>5</b>	<b>Documentation</b>	<b>11</b>
<b>6</b>	<b>Table of Contents</b>	<b>13</b>
6.1	Simple test . . . . .	13
6.2	adafruit_max31855 . . . . .	13
6.2.1	Implementation Notes . . . . .	14
<b>7</b>	<b>Indices and tables</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>
	<b>Index</b>	<b>19</b>



CircuitPython driver for the [MAX31855 Thermocouple Amplifier Breakout](#)



# 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-max31855
```

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

```
sudo pip3 install adafruit-circuitpython-max31855
```

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-max31855
```



## CHAPTER 3

---

### Usage Example

---

Of course, you must import the library to use it:

```
import adafruit_max31855
```

You also need to create an SPI interface object, and a pin object for the chip select pin. You can use any pin for the CS, but we use D5 here:

```
from busio import SPI
from digitalio import DigitalInOut
import board

spi = SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)
cs = DigitalInOut(board.D5)
```

Next, just create the sensor object:

```
sensor = adafruit_max31855.MAX31855(spi, cs)
```

And you can start making measurements:

```
print(sensor.temperature)
```

The temperature is read in degrees Celsius (°C). You have to convert it to other units yourself, if you need it.



## 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/max31855\_simpletest.py

```
1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  import time
5  import board
6  import busio
7  import digitalio
8  import adafruit_max31855
9
10 spi = busio.SPI(board.SCK, MOSI=board.MOSI, MISO=board.MISO)
11 cs = digitalio.DigitalInOut(board.D5)
12
13 max31855 = adafruit_max31855.MAX31855(spi, cs)
14 while True:
15     tempC = max31855.temperature
16     tempF = tempC * 9 / 5 + 32
17     print("Temperature: {} C {} F ".format(tempC, tempF))
18     time.sleep(2.0)
```

### 6.2 adafruit\_max31855

This is a CircuitPython driver for the Maxim Integrated MAX31855 thermocouple amplifier module.

- Author(s): Radomir Dopieralski

## 6.2.1 Implementation Notes

### Hardware:

- Adafruit [MAX31855 Thermocouple Amplifier Breakout](#) (Product ID: 269)

### 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](https://github.com/adafruit/Adafruit_CircuitPython_BusDevice)

**class** `adafruit_max31855.MAX31855` (*spi, cs*)

Driver for the MAX31855 thermocouple amplifier.

**reference\_temperature**

Internal reference temperature in degrees Celsius.

**temperature**

Thermocouple temperature in degrees Celsius.

**temperature\_NIST**

Thermocouple temperature in degrees Celsius, computed using raw voltages and NIST approximation for Type K, see: [https://srdata.nist.gov/its90/download/type\\_k.tab](https://srdata.nist.gov/its90/download/type_k.tab)

## CHAPTER 7

---

### Indices and tables

---

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



### a

adafruit\_max31855, [13](#)



## A

`adafruit_max31855` (*module*), [13](#)

## M

`MAX31855` (*class in adafruit\_max31855*), [14](#)

## R

`reference_temperature`  
(*adafruit\_max31855.MAX31855 attribute*),  
[14](#)

## T

`temperature` (*adafruit\_max31855.MAX31855 attribute*), [14](#)

`temperature_NIST` (*adafruit\_max31855.MAX31855 attribute*), [14](#)