

---

# **Adafruit THERMISTOR Library Documentation**

***Release 1.0***

**Phillip Moyer**

**Aug 03, 2018**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Usage Example</b>	<b>5</b>
<b>3</b>	<b>Contributing</b>	<b>7</b>
<b>4</b>	<b>Building locally</b>	<b>9</b>
4.1	Sphinx documentation . . . . .	9
<b>5</b>	<b>Table of Contents</b>	<b>11</b>
5.1	Simple test . . . . .	11
5.2	adafruit_thermistor - Read temperature with a thermistor . . . . .	12
5.2.1	Implementation Notes . . . . .	12
<b>6</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Python Module Index</b>	<b>15</b>



Thermistors are resistors that predictably change resistance with temperature. This driver uses an analog reading and math to determine the temperature. They are commonly used as a low cost way to measure temperature.



# CHAPTER 1

---

## Dependencies

---

This driver depends on:

- [Adafruit CircuitPython](#)

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





## CHAPTER 2

---

### Usage Example

---

The hardest part of using the driver is its initialization. Here is an example for the thermistor on the Circuit Playground and Circuit Playground Express. Its a 10k series resistor, 10k nominal resistance, 25 celsius nominal temperature and 3950 B coefficient.



## CHAPTER 3

---

### Contributing

---

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



## CHAPTER 4

---

### Building locally

---

To build this library locally you'll need to install the `circuitpython-build-tools` package.

```
python3 -m venv .env
source .env/bin/activate
pip install circuitpython-build-tools
```

Once installed, make sure you are in the virtual environment:

```
source .env/bin/activate
```

Then run the build:

```
circuitpython-build-bundles --filename_prefix adafruit-circuitpython-thermistor --
↳library_location .
```

### 4.1 Sphinx documentation

Sphinx is used to build the documentation based on rST files and comments in the code. First, install dependencies (feel free to reuse the virtual environment from above):

```
python3 -m venv .env
source .env/bin/activate
pip install Sphinx sphinx-rtd-theme
```

Now, once you have the virtual environment activated:

```
cd docs
sphinx-build -E -W -b html . _build/html
```

This will output the documentation to `docs/_build/html`. Open the `index.html` in your browser to view them. It will also (due to `-W`) error out on any warning like Travis will. This is a good way to locally verify it will pass.



## 5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/thermistor\_simpletest.py

```
1 import time
2 import board
3 import adafruit_thermistor
4
5 # these values work with the Adafruit CircuitPlayground Express.
6 # they may work with other thermistors as well, as they're fairly standard,
7 # though the pin will likely need to change (ie board.A1)
8 # pylint: disable=no-member
9 pin = board.TEMPERATURE
10 resistor = 10000
11 resistance = 10000
12 nominal_temp = 25
13 b_coefficient = 3950
14
15 thermistor = adafruit_thermistor.Thermistor(pin, resistor, resistance, nominal_temp,
16 ↪b_coefficient)
17
18 # print the temperature in C and F to the serial console every second
19 while True:
20     celsius = thermistor.temperature
21     fahrenheit = (celsius * 9 / 5) + 32
22     print('== Temperature ==\n{} *C\n{} *F\n'.format(celsius, fahrenheit))
23     time.sleep(1)
```

## 5.2 adafruit\_thermistor - Read temperature with a thermistor

A thermistor is a resistor that varies with temperature. This driver takes the parameters of that resistor and its series resistor to determine the current temperature. To hook one up, connect an analog input pin to the connection between the resistor and the thermistor. Be careful to note if the thermistor is connected on the high side (from analog input up to high logic level/3.3 or 5 volts) or low side (from analog input down to ground). The initializer takes an optional `high_side` boolean that defaults to `True` and indicates if that the thermistor is connected on the high side vs. low side.

- Author(s): Scott Shawcroft

### 5.2.1 Implementation Notes

#### Hardware:

- Adafruit [10K Precision Epoxy Thermistor - 3950 NTC](#) (Product ID: 372)
- Adafruit [Circuit Playground Express](#) (Product ID: 3333)

#### Software and Dependencies:

- Adafruit CircuitPython firmware: <https://github.com/adafruit/circuitpython/releases>

#### Notes:

1. Check the datasheet of your thermistor for the values.

```
class adafruit_thermistor.Thermistor(pin, series_resistor, nominal_resistance, nomi-  
nal_temperature, b_coefficient, *, high_side=True)
```

Thermistor driver

**temperature**

The temperature of the thermistor in celsius



## CHAPTER 6

---

### Indices and tables

---

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



### a

`adafruit_thermistor`, [11](#)



## A

`adafruit_thermistor` (module), [11](#)

## T

`temperature` (`adafruit_thermistor.Thermistor` attribute), [12](#)

`Thermistor` (class in `adafruit_thermistor`), [12](#)