

---

# **Adafruit CircuitPython DHT Library Documentation**

***Release 1.0***

**Mike McWethy**

**Aug 03, 2018**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Usage Example</b>	<b>5</b>
2.1	Hardware Set-up . . . . .	5
2.2	Basics . . . . .	5
2.3	Read temperature and humidity . . . . .	5
<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_dhtlib . . . . .	11
<b>6</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Python Module Index</b>	<b>15</b>



CircuitPython support for the DHT11 and DHT22 temperature and humidity devices.



# 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](#).





### 2.1 Hardware Set-up

The DHT11 and DHT22 devices both need a pull-resistor on the data signal wire. This resistor is in the range of 1k to 5k. Please check your device datasheet for the appropriate value.

### 2.2 Basics

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

```
import adafruit_dht
```

The DHT type devices use single data wire, so import the board pin

```
from board import <pin>
```

Now, to initialize the DHT11 device:

```
dht_device = adafruit_dht.DHT11(<pin>)
```

OR initialize the DHT22 device:

```
dht_device = adafruit_dht.DHT22(<pin>)
```

### 2.3 Read temperature and humidity

Now get the temperature and humidity values

```
temperature = dht_device.temperature  
humidity = dht_device.humidity
```

These properties may raise an exception if a problem occurs. You should use try/raise logic and catch `RuntimeError` and then retry getting the values after 1/2 second.

## 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-venv16070 --
↳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/dht\_simpletest.py

```
1 import time
2 from board import D2
3 import adafruit_dht
4
5 #initial the dht device
6 dhtDevice = adafruit_dht.DHT22(D2)
7
8 while True:
9     try:
10         # show the values to the serial port
11         temperature = dhtDevice.temperature * (9 / 5) + 32
12         humidity = dhtDevice.humidity
13         print("Temp: {:.1f} F Humidity: {}% ".format(temperature, humidity))
14
15     except RuntimeError as error:
16         print(error.args)
17
18     time.sleep(2.0)
```

## 5.2 adafruit\_dhtlib

CircuitPython support for the DHT11 and DHT22 temperature and humidity devices.

- Author(s): Mike McWethy

**class** `adafruit_dht.DHT11(pin)`

Support for DHT11 device.

**Parameters** `pin` (*Pin*) – digital pin used for communication

**class** `adafruit_dht.DHT22(pin)`

Support for DHT22 device.

**Parameters** `pin` (*Pin*) – digital pin used for communication

**class** `adafruit_dht.DHTBase(dht11, pin, trig_wait)`

base support for DHT11 and DHT22 devices

**humidity**

humidity current reading. It makes sure a reading is available

Raises `RuntimeError` exception for checksum failure and for insufficient data returned from the device (try again)

**measure()**

measure runs the communications to the DHT11/22 type device. if successful, the class properties temperature and humidity will return the reading returned from the device.

Raises `RuntimeError` exception for checksum failure and for insufficient data returned from the device (try again)

**temperature**

temperature current reading. It makes sure a reading is available

Raises `RuntimeError` exception for checksum failure and for insufficient data returned from the device (try again)



## CHAPTER 6

---

### Indices and tables

---

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



### **a**

`adafruit_dht`, [11](#)



### A

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

### D

`DHT11` (class in `adafruit_dht`), [11](#)

`DHT22` (class in `adafruit_dht`), [12](#)

`DHTBase` (class in `adafruit_dht`), [12](#)

### H

`humidity` (`adafruit_dht.DHTBase` attribute), [12](#)

### M

`measure()` (`adafruit_dht.DHTBase` method), [12](#)

### T

`temperature` (`adafruit_dht.DHTBase` attribute), [12](#)