

---

# **Adafruit CircuitPython DHT Library Documentation**

***Release 1.0***

**Mike McWethy**

**Jan 23, 2020**



---

## 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>
3.1	Hardware Set-up . . . . .	7
3.2	Basics . . . . .	7
3.3	Read temperature and humidity . . . . .	7
<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_dhtlib . . . . .	13
<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 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](#).





## 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-dht
```

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

```
sudo pip3 install adafruit-circuitpython-dht
```

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-dht
```



### 3.1 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.

### 3.2 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>)
```

### 3.3 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 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/dht\_simpletest.py

```
1 import time
2 import board
3 import adafruit_dht
4
5 # Initial the dht device, with data pin connected to:
6 dhtDevice = adafruit_dht.DHT22(board.D18)
7
8 while True:
9     try:
10         # Print the values to the serial port
11         temperature_c = dhtDevice.temperature
12         temperature_f = temperature_c * (9 / 5) + 32
13         humidity = dhtDevice.humidity
14         print("Temp: {:.1f} F / {:.1f} C    Humidity: {}% "
15               .format(temperature_f, temperature_c, humidity))
16
17     except RuntimeError as error:
18         # Errors happen fairly often, DHT's are hard to read, just keep going
19         print(error.args[0])
20
21     time.sleep(2.0)
```

## 6.2 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 7

---

### Indices and tables

---

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



---

## Python Module Index

---

### a

`adafruit_dht`, [13](#)



### A

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

### D

`DHT11` (*class in adafruit\_dht*), [14](#)

`DHT22` (*class in adafruit\_dht*), [14](#)

`DHTBase` (*class in adafruit\_dht*), [14](#)

### H

`humidity` (*adafruit\_dht.DHTBase attribute*), [14](#)

### M

`measure()` (*adafruit\_dht.DHTBase method*), [14](#)

### T

`temperature` (*adafruit\_dht.DHTBase attribute*), [14](#)