
Adafruit CircuitPython DHT Library Documentation

Release 1.0

Mike McWethy

Oct 23, 2018

Contents

1	Dependencies	3
2	Usage Example	5
2.1	Hardware Set-up	5
2.2	Basics	5
2.3	Read temperature and humidity	5
3	Contributing	7
4	Building locally	9
4.1	Sphinx documentation	9
5	Table of Contents	11
5.1	Simple test	11
5.2	adafruit_dhtlib	11
6	Indices and tables	13
	Python Module Index	15

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 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)
```

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`), [12](#)

`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](#)