Adafruit CircuitPython DHT Library Documentation

Release 1.0

Mike McWethy

Mar 02, 2021

Contents

1 Dependencies **3**

2 Installing from PyPI **5**

3 Usage Example **7**
- 3.1 Hardware Set-up . . . 7
- 3.2 Basics . . . 7
- 3.3 Read temperature and humidity . . . 7

4 Contributing **9**

5 Documentation **11**

6 Table of Contents **13**
- 6.1 Simple test . . . 13
- 6.2 `adafruit_dhtlib` . . . 14

7 Indices and tables **15**

Python Module Index **17**

Index **19**

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
```

CHAPTER 3

Usage Example

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 at least 2 seconds. If you try again to get a result within 2 seconds, cached values are returned.

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.

CHAPTER 6

Table of Contents

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/dht_simpletest.py

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

import time
import board
import adafruit_dht

# Initial the dht device, with data pin connected to:
dhtDevice = adafruit_dht.DHT22(board.D18)

# you can pass DHT22 use_pulseio=False if you wouldn't like to use pulseio.
# This may be necessary on a Linux single board computer like the Raspberry Pi,
# but it will not work in CircuitPython.
# dhtDevice = adafruit_dht.DHT22(board.D18, use_pulseio=False)

while True:
    try:
        # Print the values to the serial port
        temperature_c = dhtDevice.temperature
        temperature_f = temperature_c * (9 / 5) + 32
        humidity = dhtDevice.humidity
        print(
            "Temp: {:.1f} F / {:.1f} C    Humidity: {}% ".format(
                temperature_f, temperature_c, humidity
            )
        )

```

(continues on next page)

(continued from previous page)

```
    except RuntimeError as error:
        # Errors happen fairly often, DHT's are hard to read, just keep going
        print(error.args[0])
        time.sleep(2.0)
        continue
    except Exception as error:
        dhtDevice.exit()
        raise error

    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*, *use_pulseio=True*)

Support for DHT11 device.

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

class adafruit_dht.**DHT22**(*pin*, *use_pulseio=True*)

Support for DHT22 device.

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

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

base support for DHT11 and DHT22 devices

exit()

Cleans up the PulseIn process. Must be called explicitly

humidity

humidity current reading. It makes sure a reading is available

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

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

adafruit_dht, 14

Index

A

`adafruit_dht` (*module*), 14

D

`DHT11` (*class in adafruit_dht*), 14
`DHT22` (*class in adafruit_dht*), 14
`DHTBase` (*class in adafruit_dht*), 14

E

`exit()` (*adafruit_dht.DHTBase method*), 14

H

`humidity` (*adafruit_dht.DHTBase attribute*), 14

M

`measure()` (*adafruit_dht.DHTBase method*), 14

T

`temperature` (*adafruit_dht.DHTBase attribute*), 14