

---

# **Adafruit CircuitPython DHT Library Documentation**

***Release 1.0***

**Mike McWethy**

**Oct 05, 2021**



---

## 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	Known Issues . . . . .	7
3.3	Basics . . . . .	7
3.4	Read temperature and humidity . . . . .	8
<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	DHT to Led Display . . . . .	14
6.3	Time calibration advance test . . . . .	15
6.4	adafruit_dhtlib . . . . .	17
<b>7</b>	<b>Indices and tables</b>	<b>19</b>
	<b>Python Module Index</b>	<b>21</b>
	<b>Index</b>	<b>23</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

Designed specifically to work with the Adafruit DHT series sensors:

- Adafruit [DHT22 temperature-humidity sensor + extras](#)
- Adafruit [DHT11 temperature-humidity sensor + extras](#)

---

**Note:** 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 the device datasheet for the appropriate value.
- Be sure that you are running Buster Operating System.
- Make sure that your user is part of the `gpio` group.

### 3.2 Known Issues

- Library may or may not work in Linux 64-bit platforms
- Raspberry PI-ZERO does not provide reliable readings
- Readings in FeatherS2 does not work as expected.

### 3.3 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>)
```

## 3.4 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](#).





### 6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/dht\_simpletest.py

```
1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  import time
5  import board
6  import adafruit_dht
7
8  # Initial the dht device, with data pin connected to:
9  dhtDevice = adafruit_dht.DHT22(board.D18)
10
11  # you can pass DHT22 use_pulseio=False if you wouldn't like to use pulseio.
12  # This may be necessary on a Linux single board computer like the Raspberry Pi,
13  # but it will not work in CircuitPython.
14  # dhtDevice = adafruit_dht.DHT22(board.D18, use_pulseio=False)
15
16  while True:
17      try:
18          # Print the values to the serial port
19          temperature_c = dhtDevice.temperature
20          temperature_f = temperature_c * (9 / 5) + 32
21          humidity = dhtDevice.humidity
22          print(
23              "Temp: {:.1f} F / {:.1f} C    Humidity: {}% ".format(
24                  temperature_f, temperature_c, humidity
25              )
26          )
27
```

(continues on next page)

(continued from previous page)

```

28     except RuntimeError as error:
29         # Errors happen fairly often, DHT's are hard to read, just keep going
30         print(error.args[0])
31         time.sleep(2.0)
32         continue
33     except Exception as error:
34         dhtDevice.exit()
35         raise error
36
37     time.sleep(2.0)

```

## 6.2 DHT to Led Display

Example of reading temperature and humidity from a DHT device and displaying results to the serial port and a 8 digit 7-segment display

Listing 2: examples/dht\_to\_led\_display.py

```

1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  """
5  example of reading temperature and humidity from a DHT device
6  and displaying results to the serial port and a 8 digit 7-segment display
7  the DHT device data wire is connected to board.D2
8  """
9  # import for dht devices and 7-segment display devices
10 import time
11 from board import D2, TX, RX, D1
12 import busio
13 import digitalio
14 from adafruit_max7219 import bcddigits
15 import adafruit_dht
16
17
18 clk = RX
19 din = TX
20 cs = digitalio.DigitalInOut(D1)
21 spi = busio.SPI(clk, MOSI=din)
22 display = bcddigits.BCDDigits(spi, cs, nDigits=8)
23 display.brightness(5)
24
25 # initial the dht device
26 dhtDevice = adafruit_dht.DHT22(D2)
27
28 while True:
29     try:
30         # show the values to the serial port
31         temperature = dhtDevice.temperature * (9 / 5) + 32
32         humidity = dhtDevice.humidity
33         # print("Temp: {:.1f} F Humidity: {}% ".format(temperature, humidity))
34
35         # now show the values on the 8 digit 7-segment display
36         display.clear_all()

```

(continues on next page)

(continued from previous page)

```

37     display.show_str(0, "{:5.1f}{:5.1f}".format(temperature, humidity))
38     display.show()
39
40     except RuntimeError as error:
41         print(error.args[0])
42
43     time.sleep(2.0)

```

## 6.3 Time calibration advance test

Example to identify best waiting time for the sensor

Listing 3: examples/dht\_time\_calibration\_advance.py

```

1  # SPDX-FileCopyrightText: 2021 yeyeto2788 for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  """
5  This script let's you check the best timing for you sensor as other people have face_
6  ↳ timing issues
7  as seen on issue https://github.com/adafruit/Adafruit\_CircuitPython\_DHT/issues/66.
8
9  By changing the variables values below you will be able to check the best timing for_
10 ↳ you sensor,
11 take into account that by most datasheets the timing for the sensor are 0.001 DHT22_
12 ↳ and
13 0.018 for DHT11 which are the default values of the library.
14 """
15
16 import json
17 import time
18
19 import board
20
21 import adafruit_dht
22
23 # Change the pin used below
24 pin_to_use = "PG6"
25
26 # Maximum number of tries per timing
27 max_retries_per_time = 10
28 # Minimum wait time from where to start testing
29 min_time = 1500
30 # Maximum wait time on where to stop testing
31 max_time = 2000
32 # Increment on time
33 time_increment = 100
34
35 # Variable to store all reads on a try
36 reads = {}
37
38 initial_msg = f"""
39 \nInitializing test with the following parameters:

```

(continues on next page)

(continued from previous page)

```

38 - Maximum retries per waiting time: {max_retries_per_time}
39 - Start time (ms): {min_time}
40 - End time (ms): {max_time}
41 - Increment time (ms): {time_increment}
42
43 This execution will try to read the sensor {max_retries_per_time} times
44 for {len(range(min_time, max_time, time_increment))} different wait times values.
45
46 """
47 # Print initial message on the console.
48 print(initial_msg)
49
50 for milliseconds in range(min_time, max_time, time_increment):
51     # Instantiate the DHT11 object.
52     dhtDevice = adafruit_dht.DHT11(pin=getattr(board, pin_to_use))
53     # Change the default wait time for triggering the read.
54     # pylint: disable=protected-access
55     dhtDevice._trig_wait = milliseconds
56
57     # pylint: disable=protected-access
58     print(f"Using 'trig_wait' of {dhtDevice._trig_wait}")
59     # Reset the read count for next loop
60     reads_count = 0
61
62     # Create the key on the reads dictionary with the milliseconds used on
63     # this try.
64     if milliseconds not in reads:
65         reads[milliseconds] = {"total_reads": 0}
66
67     for try_number in range(0, max_retries_per_time):
68         try:
69             # Read temperature and humidity
70             temperature = dhtDevice.temperature
71             humidity = dhtDevice.humidity
72             read_values = {"temperature": temperature, "humidity": humidity}
73
74             if try_number not in reads[milliseconds]:
75                 reads[milliseconds][try_number] = read_values
76
77             reads_count += 1
78         except RuntimeError as e:
79             time.sleep(2)
80         else:
81             time.sleep(1)
82
83     reads[milliseconds]["total_reads"] = reads_count
84
85     print(f"Total read(s): {reads[milliseconds]['total_reads']}\n")
86     dhtDevice.exit()
87
88 # Gather the highest read numbers from all reads done.
89 best_result = max(
90     [
91         reads[milliseconds]["total_reads"]
92         for milliseconds in reads # pylint: disable=consider-using-dict-items
93     ]
94 )

```

(continues on next page)

(continued from previous page)

```

95 # Gather best time(s) in milliseconds where we got more reads
96 best_times = [
97     milliseconds
98     for milliseconds in reads # pylint: disable=consider-using-dict-items
99     if reads[milliseconds]["total_reads"] == best_result
100 ]
101 print(
102     f"Maximum reads: {best_result} out of {max_retries_per_time} with the "
103     f"following times: {' '.join([str(t) for t in best_times])}"
104 )
105
106 # change the value on the line below to see all reads performed.
107 print_all = False
108 if print_all:
109     print(json.dumps(reads))

```

## 6.4 adafruit\_dhtlib

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

- Author(s): Mike McWethy

### Hardware:

- Adafruit [DHT22 temperature-humidity sensor + extras](#) (Product ID: 385)
- Adafruit [DHT11 basic temperature-humidity sensor + extras](#) (Product ID: 386)

### Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://circuitpython.org/downloads>

**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 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`





### **a**

`adafruit_dht`, [17](#)



### A

`adafruit_dht` (*module*), [17](#)

### D

`DHT11` (*class in adafruit\_dht*), [17](#)

`DHT22` (*class in adafruit\_dht*), [17](#)

`DHTBase` (*class in adafruit\_dht*), [17](#)

### E

`exit()` (*adafruit\_dht.DHTBase method*), [17](#)

### H

`humidity` (*adafruit\_dht.DHTBase attribute*), [17](#)

### M

`measure()` (*adafruit\_dht.DHTBase method*), [17](#)

### T

`temperature` (*adafruit\_dht.DHTBase attribute*), [17](#)