

---

# AdafruitHTU21D Library Documentation

*Release 1.0*

**ktown**

**May 08, 2019**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Usage Example</b>	<b>5</b>
<b>3</b>	<b>Contributing</b>	<b>7</b>
<b>4</b>	<b>Building locally</b>	<b>9</b>
4.1	Zip release files . . . . .	9
4.2	Sphinx documentation . . . . .	9
<b>5</b>	<b>Table of Contents</b>	<b>11</b>
5.1	Simple test . . . . .	11
5.2	adafruit_htu21d . . . . .	11
5.2.1	Implementation Notes . . . . .	11
<b>6</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Python Module Index</b>	<b>15</b>



This driver enables you to use the Adafruit HTU21D-F temperature and humidity breakout with CircuitPython.



# CHAPTER 1

---

## Dependencies

---

This driver depends on:

- [Adafruit CircuitPython](#)
- [Bus Device](#)

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the [Adafruit library and driver bundle](#).





## CHAPTER 2

---

### Usage Example

---

```
import time
import board
import busio
from adafruit_htu21d import HTU21D

# Create library object using our Bus I2C port
i2c = busio.I2C(board.SCL, board.SDA)
sensor = HTU21D(i2c)

while True:
    print("\nTemperature: %0.1f C" % sensor.temperature)
    print("Humidity: %0.1f %% " % sensor.relative_humidity)
    time.sleep(2)
```



## CHAPTER 3

---

### Contributing

---

Contributions are welcome! Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.



### 4.1 Zip release files

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-htu21d --library_
↪location .
```

### 4.2 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/htu21d\_simpletest.py

```
1 import time
2 import board
3 import busio
4 from adafruit_htu21d import HTU21D
5
6 # Create library object using our Bus I2C port
7 i2c = busio.I2C(board.SCL, board.SDA)
8 sensor = HTU21D(i2c)
9
10
11 while True:
12     print("\nTemperature: %0.1f C" % sensor.temperature)
13     print("Humidity: %0.1f %% " % sensor.relative_humidity)
14     time.sleep(2)
```

## 5.2 adafruit\_htu21d

This is a breakout for the Adafruit HTU21D-F humidity sensor breakout.

- Author(s): ktown

### 5.2.1 Implementation Notes

**Hardware:**

---

- Adafruit [HTU21D-F Temperature & Humidity Sensor Breakout Board](#) (Product ID: 1899)

### Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://github.com/adafruit/circuitpython/releases>
- Adafruit's Bus Device library: [https://github.com/adafruit/Adafruit\\_CircuitPython\\_BusDevice](https://github.com/adafruit/Adafruit_CircuitPython_BusDevice)

**class** `adafruit_htu21d.HTU21D` (*i2c\_bus*, *address=64*)

A driver for the HTU21D-F temperature and humidity sensor. :param `i2c_bus`: The `busio.I2C` object to use. This is the only required parameter. :param `int address`: (optional) The I2C address of the device.

**measurement** (*what*)

Starts a measurement. Starts a measurement of either `HUMIDITY` or `TEMPERATURE` depending on the `what` argument. Returns immediately, and the result of the measurement can be retrieved with the `temperature` and `relative_humidity` properties. This way it will take much less time. This can be useful if you want to start the measurement, but don't want the call to block until the measurement is ready – for instance, when you are doing other things at the same time.

**relative\_humidity**

The measured relative humidity in percent.

**temperature**

The measured temperature in degrees Celcius.



## CHAPTER 6

---

### Indices and tables

---

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



**a**

adafruit\_htu21d, [11](#)



## A

adafruit\_htu21d (*module*), 11

## H

HTU21D (*class in adafruit\_htu21d*), 12

## M

measurement () (*adafruit\_htu21d.HTU21D method*),  
12

## R

relative\_humidity (*adafruit\_htu21d.HTU21D attribute*), 12

## T

temperature (*adafruit\_htu21d.HTU21D attribute*),  
12