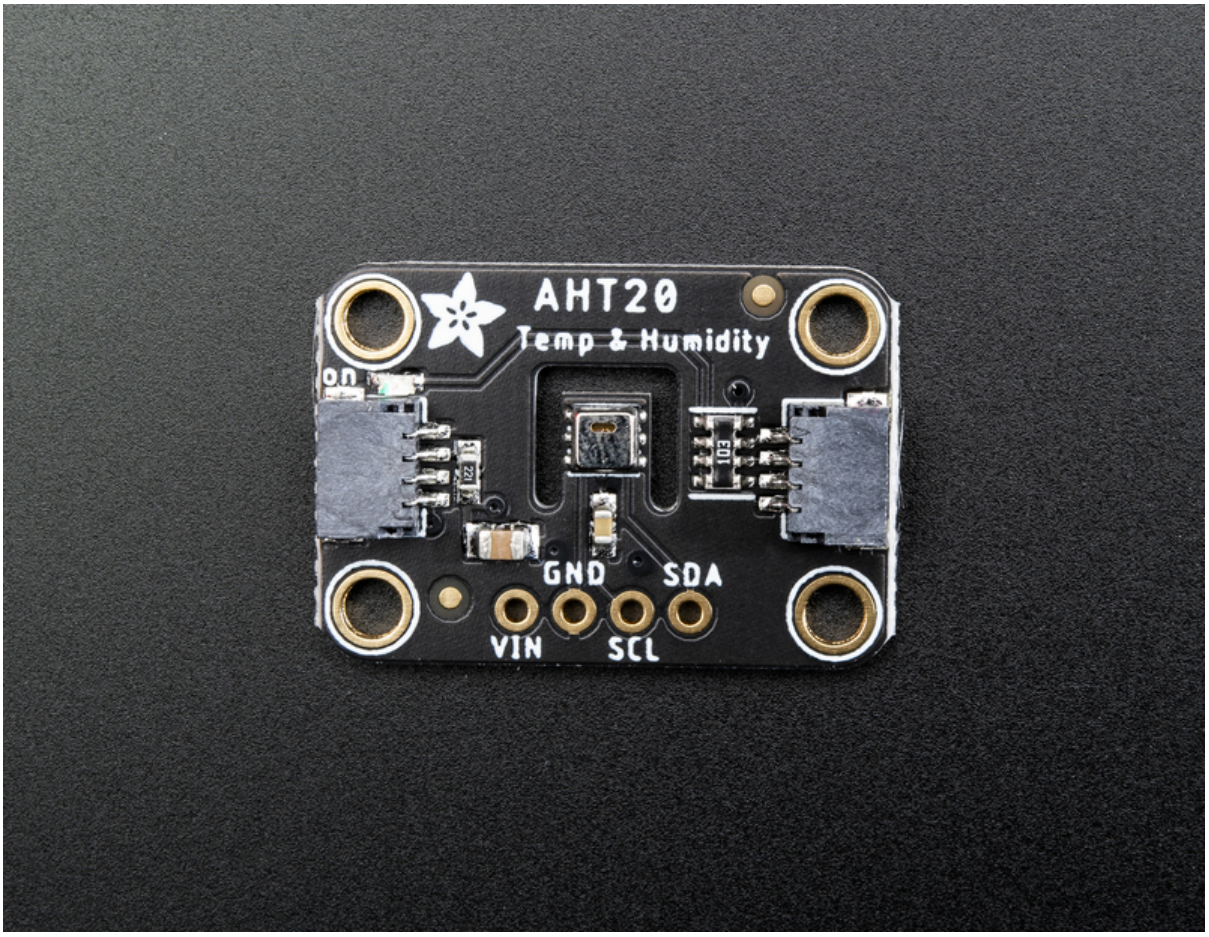
Adafruit AHT20 Temperature & Humidity Sensor

Created by Kattni Rembor



https://learn.adafruit.com/adafruit-aht20

Last updated on 2026-08-19 04:09:45 PM UTC

Table of Contents

Overview 3

Pinouts 4

- Power Pins
- I2C Logic Pins
- The "on" LED

Arduino 6

- Wiring
- Installation
- Load Example
- Example Code

Arduino Docs 9

Python & CircuitPython 9

- CircuitPython Microcontroller Wiring
- Python Computer Wiring
- CircuitPython Installation of AHT20 Library
- Python Installation of AHT20 Library
- CircuitPython & Python Usage
- Full Example Code

Python Docs 12

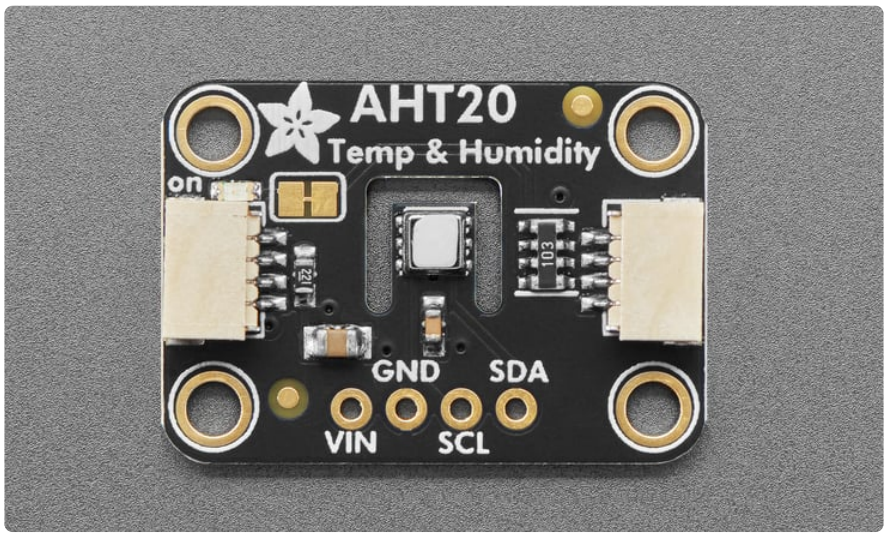
WipperSnapper 13

- What is WipperSnapper
- Wiring
- Usage

Downloads 20

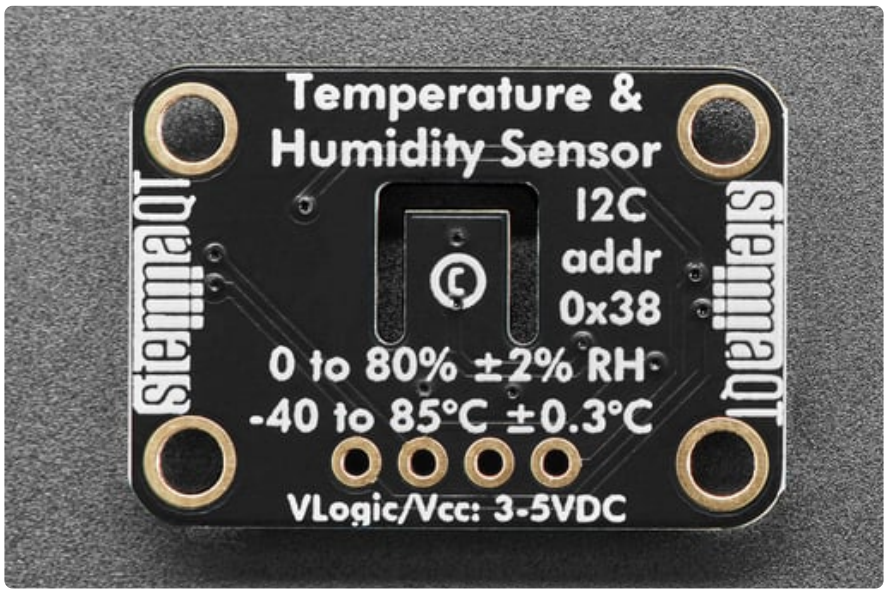
- Files
- Schematic
- Fab Print

Overview

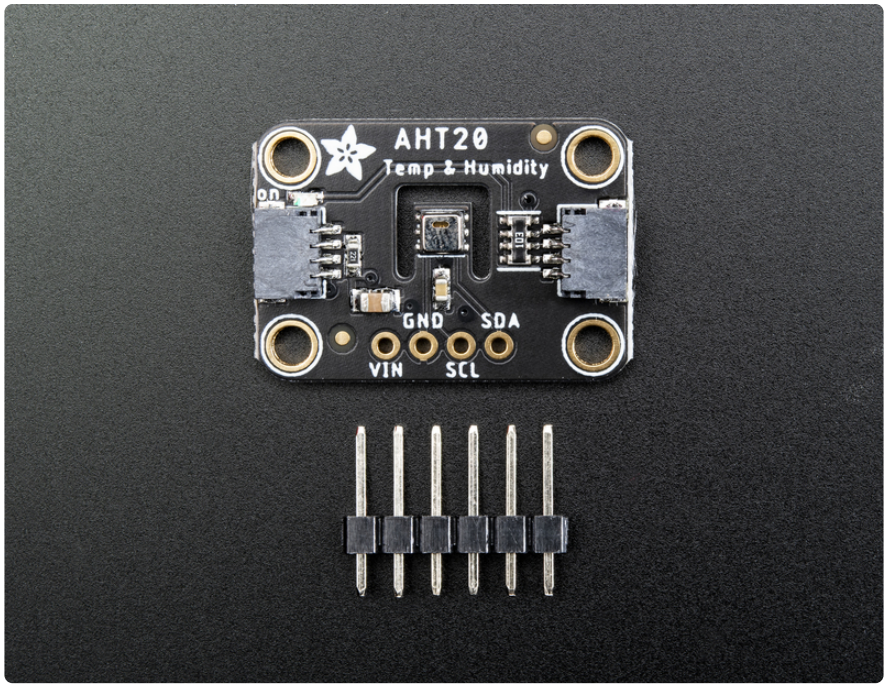


The AHT20 is a nice but inexpensive temperature and humidity sensor [from the same folks that brought us the DHT22 (http://adafru.it/385)](). You can take sensor readings as often as you like, and it uses standard I2C so its super easy to use with any Arduino or Linux/Raspberry Pi board.

This sensor has a typical accuracy of +- 2% relative humidity, and +-0.3 °C. There is only one I2C address so its not a good option when you need multiple humidity sensors.



As with all Adafruit breakouts, we've done the work to make this handy temperature-and-humidity super easy to use. We've put it on a breakout board with the required support circuitry and connectors to make it easy to work with and is now a trend we've added SparkFun Qwiic (https://adafru.it/Fpw) compatible **STEMMA QT** (https://adafru.it/Ft4) JST SH connectors that allow you to get going **without needing to solder.** Just use a STEMMA QT adapter cable (http://adafru.it/4209), plug it into your favorite microcontroller or Blinka supported SBC and you're ready to rock!



Pinouts



The default I2C address is **0x38**. It cannot be changed.

Power Pins

The sensor on the breakout requires between a 2.7V and 5.5V, and can be easily used with most microcontrollers from an Arduino to a Feather or something else.

- **VIN** - this is the power pin. To power the board, give it the same power as the logic level of your microcontroller - e.g. for a 5V micro like Arduino, use 5V
- **GND** - common ground for power and logic

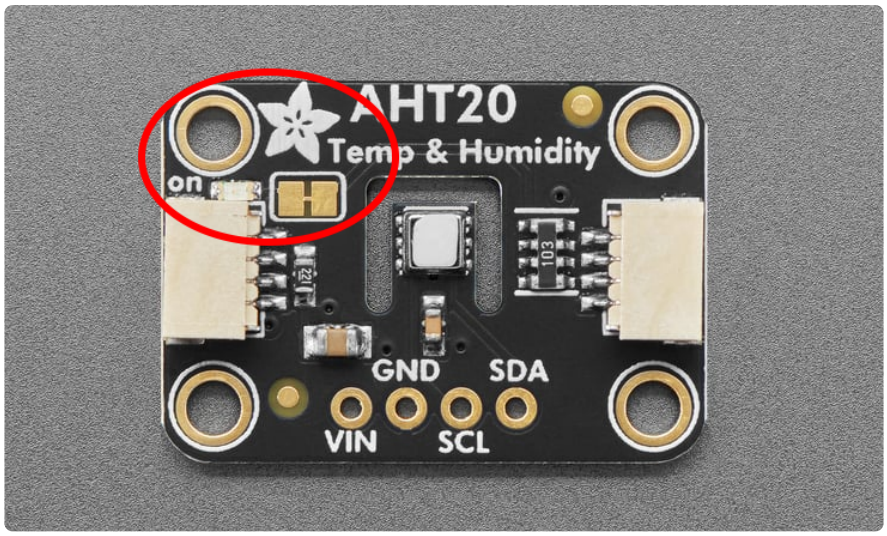
I2C Logic Pins

- **SCL** - I2C clock pin, connect to your microcontrollers I2C clock line. The logic level is the same as **VIN** and it has a 10K pullup already on it.
- **SDA** - I2C data pin, connect to your microcontrollers I2C data line. The logic level is the same as **VIN**. and it has a 10K pullup already on it.
- STEMMA QT **(https://adafru.it/Ft4)** - These connectors allow you to connect to dev boards with **STEMMA QT** connectors or to other things with various associated accessories (https://adafru.it/Ft6)

The "on" LED

There is a small LED next to the left STEMMA QT connector for telling when the breakout is on. On Rev A and B boards (denoted in a circled letter in the center back of the board). this LED cannot be disabled (unless the user removes or covers up the LED).

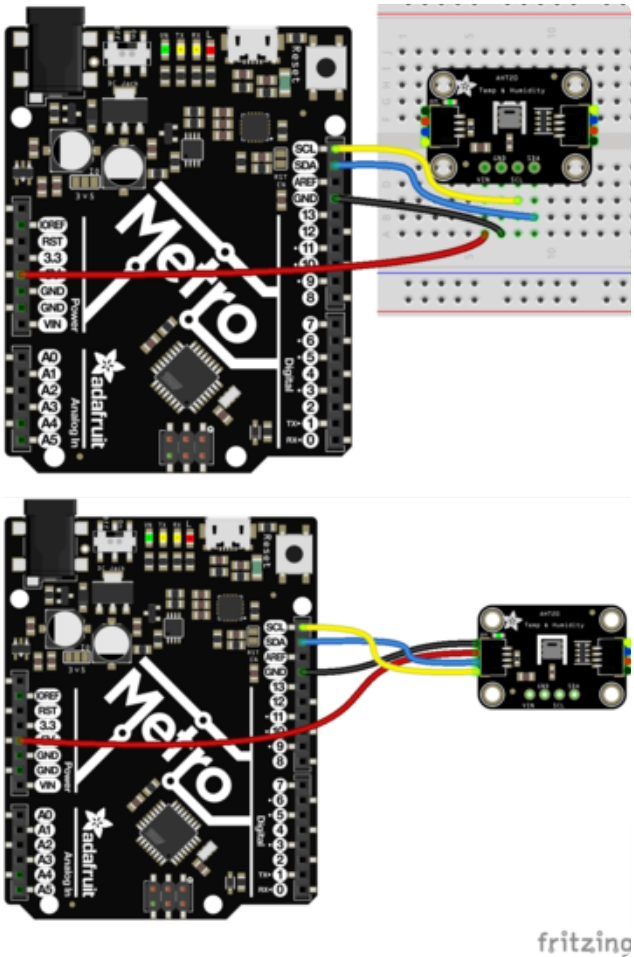
Rev C, below, has a small jumper near the LED. Uncut, the LED is on when the board is on. If the LED is not wanted, cut the small trace between the gold pads to completely disable the LED. To reenable, solder bridge the two gold pads.



Arduino

Wiring

Connecting the AHT20 to your Feather or Arduino is easy:

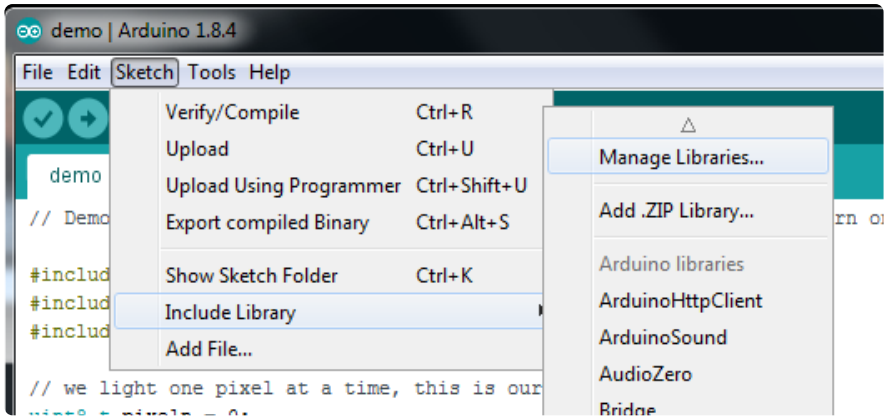


If you are running a Feather (3.3V), connect **Feather 3V** to **board VIN**
If you are running a 5V Arduino (Uno, etc.), connect **Arduino 5V** to **board VIN**
Connect **Feather or Arduino GND** to **board GND**
Connect **Feather or Arduino SCL** to **board SCL**
Connect **Feather or Arduino SDA** to **board SDA**

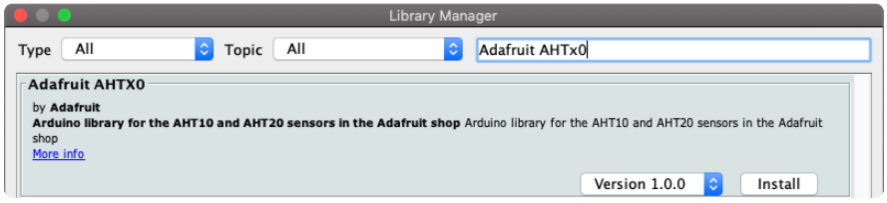
The final results should resemble the illustration above, showing an Adafruit Metro development board.

Installation

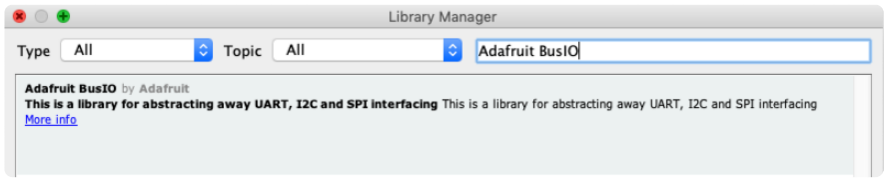
You can install the **Adafruit AHTx0 Library** for Arduino using the Library Manager in the Arduino IDE:



Click the **Manage Libraries ...** menu item, search for **Adafruit AHTx0**, and select the **Adafruit AHTx0** library:



Then follow the same process for the **Adafruit BusIO** library.



Load Example

Open up **File -> Examples -> Adafruit AHTx0 -> adafruit_aht_test** and upload to your Arduino wired up to the sensor.

Upload the sketch to your board and open up the Serial Monitor (**Tools->Serial Monitor**). You should see the the values for temperature and humidity.

Example Code

The following example code is part of the standard library, but illustrates how you can retrieve sensor data from the AHT20 for the temperature and humidity values:

```
#include <Adafruit_AHTX0.h>

Adafruit_AHTX0 aht;

void setup() {
  Serial.begin(115200);
  Serial.println("Adafruit AHT10/AHT20 demo!");

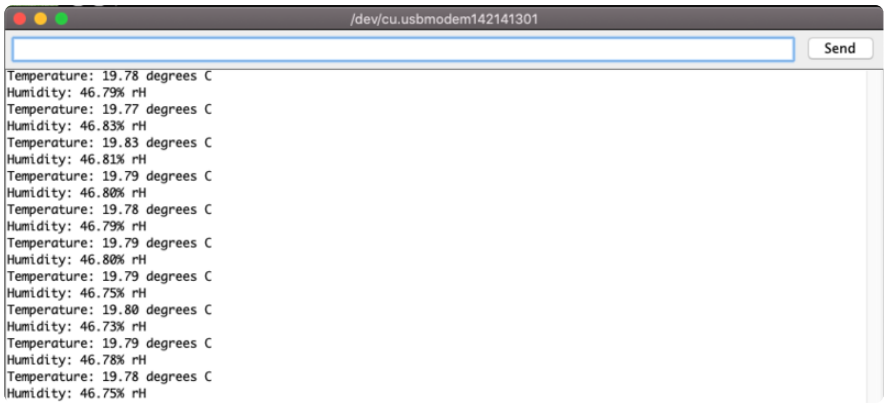
  if (!aht.begin()) {
    Serial.println("Could not find AHT? Check wiring");
    while (1)
      delay(10);
  }
  Serial.println("AHT10 or AHT20 found");
}

void loop() {
  sensors_event_t humidity, temp;
  aht.getEvent(&humidity,
               &temp); // populate temp and humidity objects with fresh data
  Serial.print("Temperature: ");
  Serial.print(temp.temperature);
  Serial.println(" degrees C");
  Serial.print("Humidity: ");
  Serial.print(humidity.relative_humidity);
  Serial.println("% rH");

  delay(500);
}
```

https://github.com/adafruit/Adafruit_AHTX0/blob/master/examples/adafruit_aht_test/adafruit_aht_test.ino

You should get something resembling the following output when you open the Serial Monitor at 115200 baud:



Arduino Docs

[Arduino Docs](https://adafru.it/LxB) (https://adafru.it/LxB)

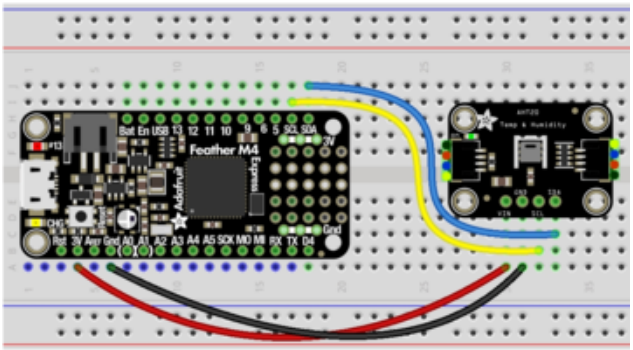
Python & CircuitPython

It's easy to use the AHT20 sensor with CircuitPython and the [Adafruit CircuitPython AHT20](https://adafru.it/QDP) (https://adafru.it/QDP) module. This module allows you to easily write Python code that reads the temperature and humidity from the sensor.

You can use this sensor with any CircuitPython microcontroller board or with a computer that has GPIO and Python [thanks to Adafruit_Blinka, our CircuitPython-for-Python compatibility library](https://adafru.it/BSN) (https://adafru.it/BSN).

CircuitPython Microcontroller Wiring

First wire up a AHT20 to your board exactly as follows. Here is an example of the AHT20 wired to a Feather using I2C:

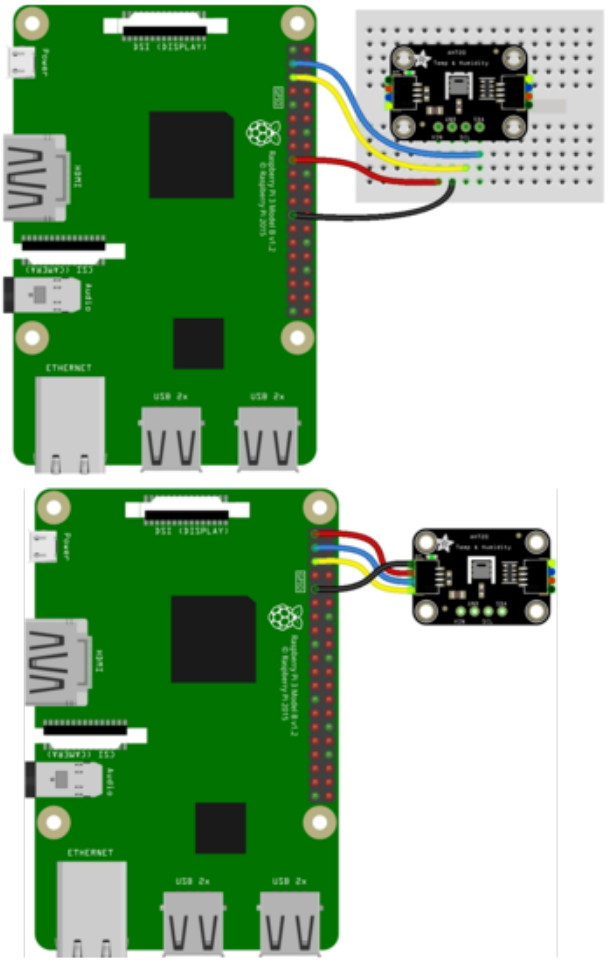


Board 3V to **sensor VIN**
Board GND to **sensor GND**
Board SCL to **sensor SCL**
Board SDA to **sensor SDA**

Python Computer Wiring

Since there's dozens of Linux computers/boards you can use we will show wiring for Raspberry Pi. For other platforms, please visit the guide for CircuitPython on Linux to see whether your platform is supported (https://adafru.it/BSN).

Here's the Raspberry Pi wired with I2C:



Pi 3V3 to **sensor VIN**
Pi GND to **sensor GND**
Pi SCL to **sensor SCL**
Pi SDA to **sensor SDA**

CircuitPython Installation of AHT20 Library

You'll need to install the Adafruit CircuitPython AHT20 (https://adafru.it/QDP) library on your CircuitPython board.

First make sure you are running the latest version of Adafruit CircuitPython (https://adafru.it/Amd) for your board.

Next you'll need to install the necessary libraries to use the hardware--carefully follow the steps to find and install these libraries from Adafruit's CircuitPython library bundle (https://adafru.it/uap). Our CircuitPython starter guide has a great page on how to install the library bundle (https://adafru.it/ABU).

Copy the following files from the bundle to the **lib** folder on your **CIRCUITPY** drive:

- **adafruit_ahtx0.mpy**
- **adafruit_bus_device**

Before continuing make sure your board's **lib** folder or root filesystem has the **adafruit_ahtx0.mpy,** and **adafruit_bus_device** file and folder copied over.

Next connect to the board's serial REPL (https://adafru.it/Awz)so you are at the CircuitPython >>> prompt.

Python Installation of AHT20 Library

You'll need to install the **Adafruit_Blinka** library that provides the CircuitPython support in Python. This may also require enabling I2C on your platform and verifying you are running Python 3. Since each platform is a little different, and Linux changes often, please visit the CircuitPython on Linux guide to get your computer ready (https://adafru.it/BSN)!

Once that's done, from your command line run the following command:

- `pip3 install adafruit-circuitpython-ahtx0`

If your default Python is version 3 you may need to run 'pip' instead. Just make sure you aren't trying to use CircuitPython on Python 2.x, it isn't supported!

CircuitPython & Python Usage

To demonstrate the usage of the sensor we'll initialize it and read the temperature and humidity from the board's Python REPL.

Run the following code to import the necessary modules and initialize the I2C connection with the sensor:

```
import board
import adafruit_ahtx0

sensor = adafruit_ahtx0.AHTx0(board.I2C())
```

Now you're ready to read values from the sensor using these properties:

- **temperature** - The temperature in Celsius.

- **relative_humidity** - The relative humidity in percent.

For example to print temperature and relative humidity values:

```
print("\nTemperature: %0.1f C" % sensor.temperature)
print("Humidity: %0.1f %%" % sensor.relative_humidity)
```

That's all there is to using the AHT20 sensor with CircuitPython!

Full Example Code

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

"""
Basic `AHTx0` example test
"""

import time

import board

import adafruit_ahtx0

# Create sensor object, communicating over the board's default I2C bus
i2c = board.I2C()  # uses board.SCL and board.SDA
# i2c = board.STEMMA_I2C()  # For using the built-in STEMMA QT connector on a microcontroller
sensor = adafruit_ahtx0.AHTx0(i2c)

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

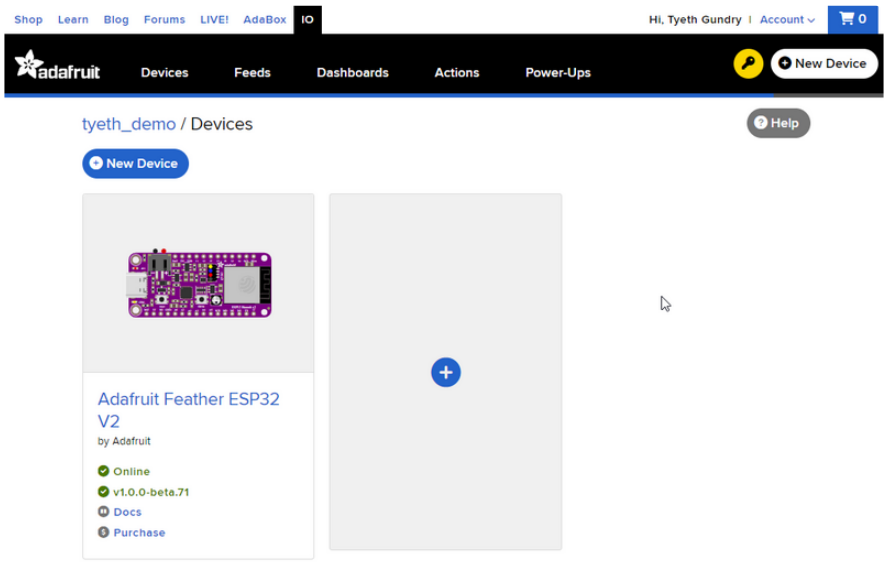
https://github.com/adafruit/Adafruit_CircuitPython_AHTx0/blob/main/examples/ahtx0_simpletest.py

---

Python Docs

Python Docs (https://adafru.it/LA7)

WipperSnapper



What is WipperSnapper

WipperSnapper is a firmware designed to turn any WiFi-capable board into an Internet-of-Things device without programming a single line of code. WipperSnapper connects to Adafruit IO (https://adafru.it/fsU), a web platform designed (by Adafruit! (https://adafru.it/Bo5)) to display, respond, and interact with your project's data.

Simply load the WipperSnapper firmware onto your board, add credentials, and plug it into power. Your board will automatically register itself with your Adafruit IO account.

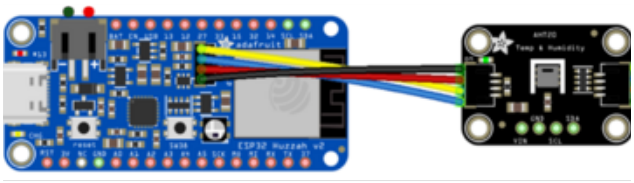
From there, you can add components to your board such as buttons, switches, potentiometers, sensors, and more! Components are dynamically added to hardware, so you can immediately start interacting, logging, and streaming the data your projects produce without writing code.

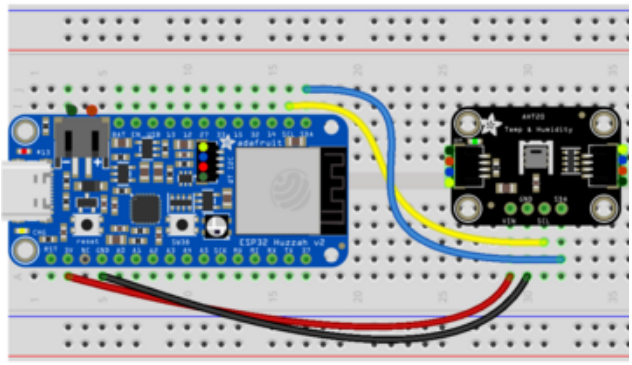
If you've never used WipperSnapper, click below to read through the quick start guide before continuing.

https://adafru.it/Vfd

Wiring

First, wire up an AHT20 to your board exactly as follows. Here is an example of the AHT20 wired to an Adafruit ESP32 Feather V2 (http://adafru.it/5400) using I2C with a STEMMA QT cable (no soldering required) (http://adafru.it/4210)



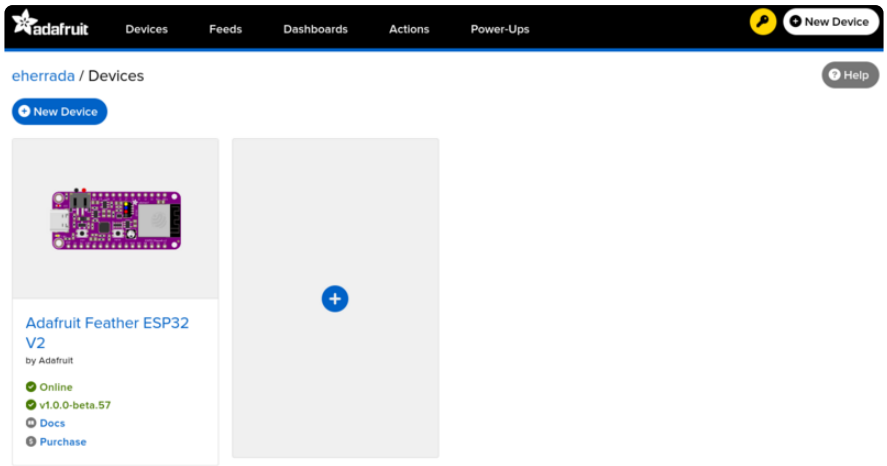


Board 3V to **sensor VIN (red wire on STEMMA QT)**
Board GND to **sensor GND (black wire on STEMMA QT)**
Board SCL to **sensor SCL (yellow wire on STEMMA QT)**
Board SDA to **sensor SDA (blue wire on STEMMA QT)**

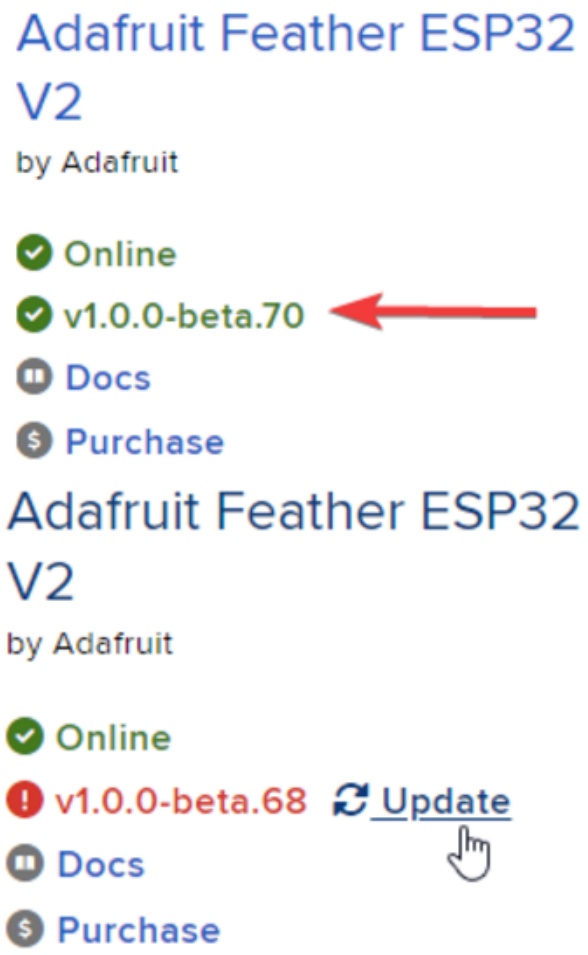
Usage

Connect your board to Adafruit IO Wippersnapper and navigate to the WipperSnapper board list **(https://adafru.it/TAu).**

On this page, **select the WipperSnapper board you're using** to be brought to the board's interface page.



If you do not see your board listed here - you need [to connect your board to Adafruit IO (https://adafru.it/Vfd)](https://adafru.it/Vfd) first.

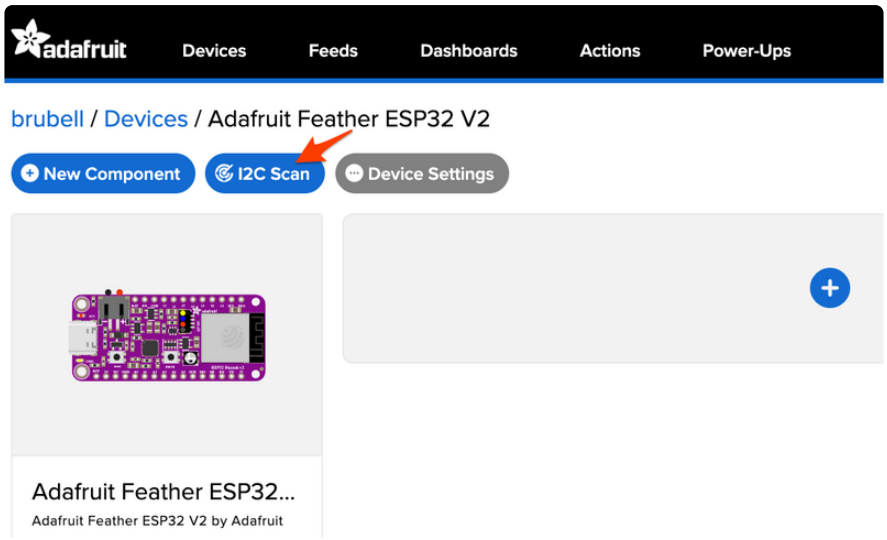


On the device page, quickly **check that you're running the latest version of the WipperSnapper firmware**.

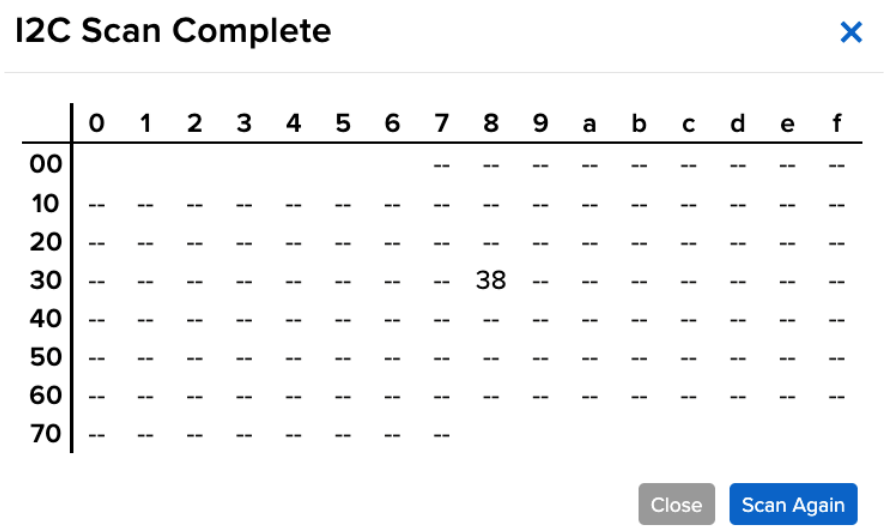
The device tile on the left indicates the version number of the firmware running on the connected board.

If the firmware version is green with a checkmark - continue with this guide.
If the firmware version is red with an "X" - [update to the latest WipperSnapper firmware (https://adafru.it/Vfd)](https://adafru.it/Vfd) on your board before continuing.

Next, make sure the sensor is plugged into your board and click the **I2C Scan** button.



You should see the AHT20's default I2C address of `0x38` pop-up in the I2C scan list.



I don't see the sensor's I2C address listed!

First, double-check the connection and/or wiring between the sensor and the board.

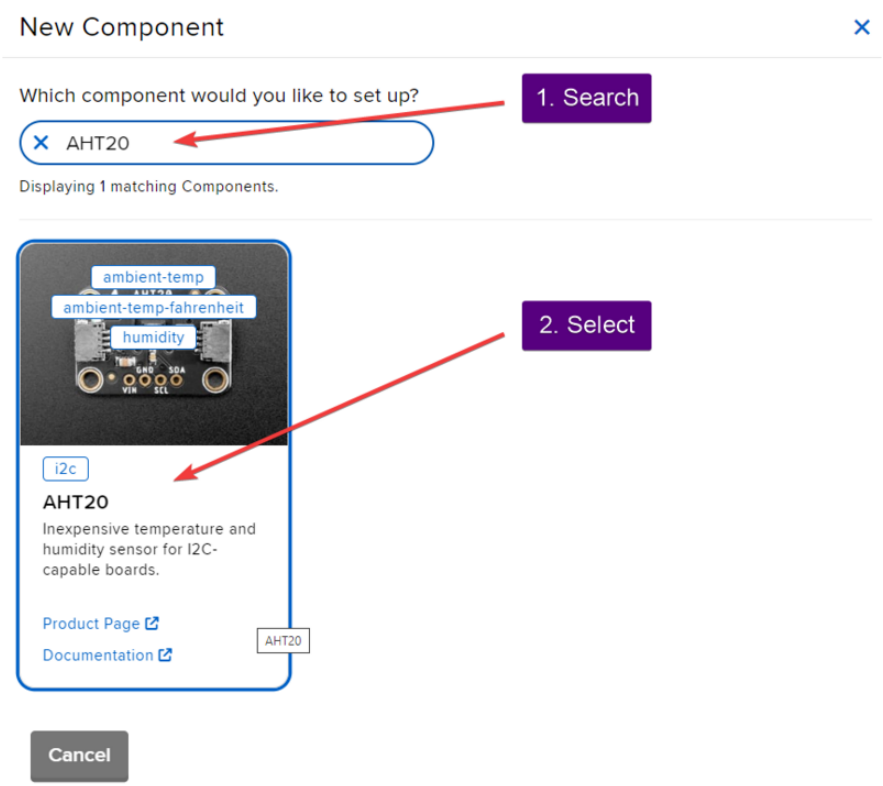
Then, reset the board and let it re-connect to Adafruit IO WipperSnapper.

With the sensor detected in an I2C scan, you're ready to add the sensor to your board.

Click the New Component button or the + button to bring up the component picker.



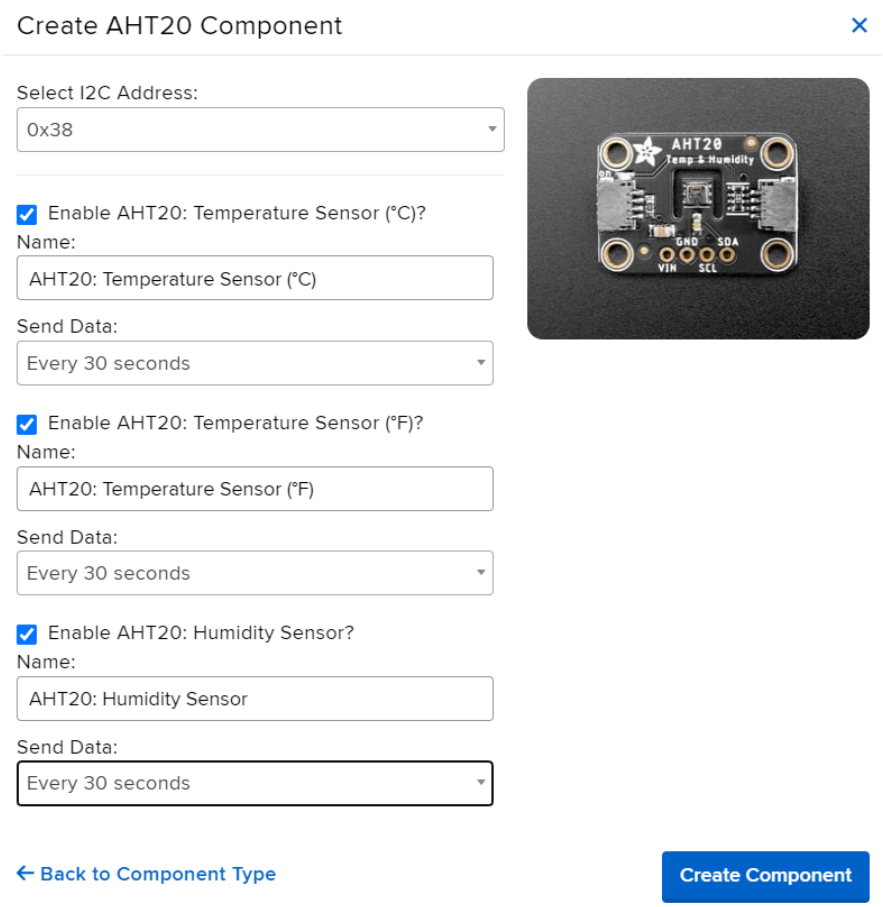
Adafruit IO supports a large amount of components. To quickly find your sensor, type `AHT20` into the search bar, then select the **AHT20** component.



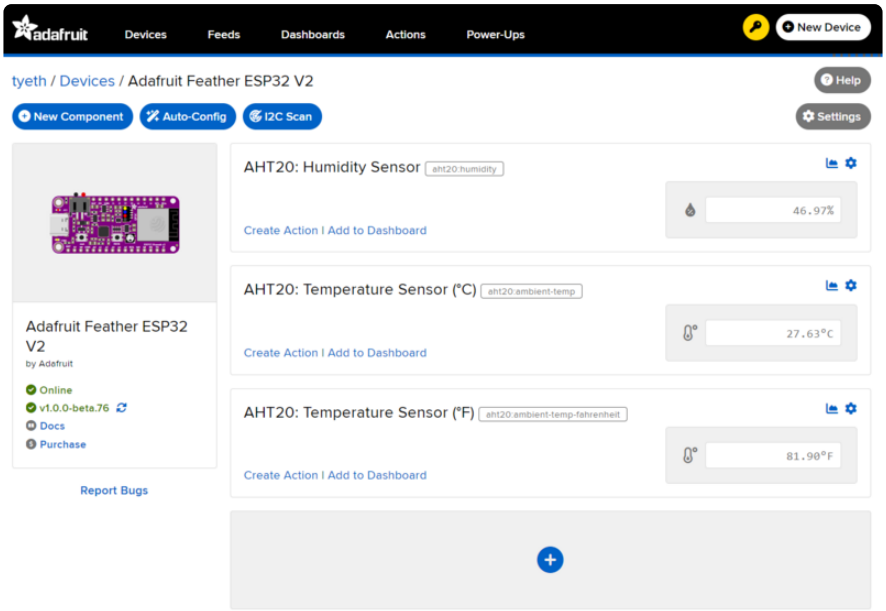
On the component configuration page, the **AHT20's** sensor address should be listed along with the sensor's settings.

The **Send Every** option is specific to each sensor's measurements. This option will tell the Feather how often it should read from the **AHT20** sensor and send the data to Adafruit IO. Measurements can range from every 30 seconds to every 24 hours.

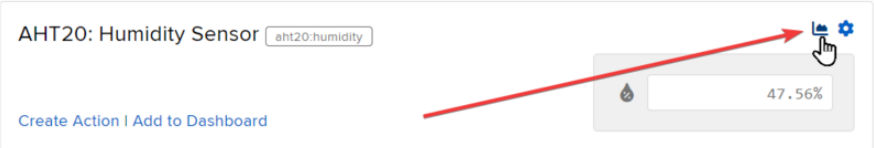
For this example, set the **Send Every** interval to every 30 seconds. When you've configured your AHT20 sensor, click **Create Component**.



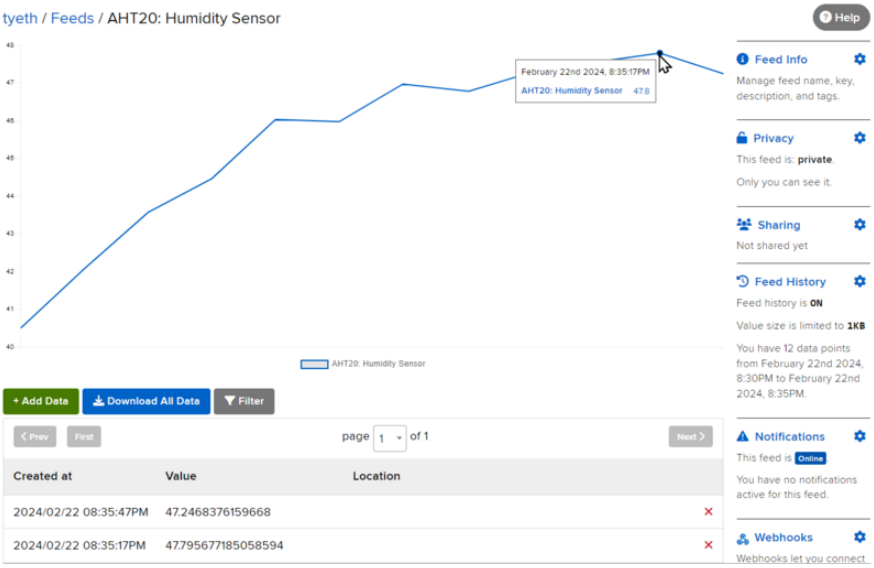
Your device interface should now show the sensor components you created. After the interval you configured elapses, WipperSnapper will automatically read values from the sensor(s) and send them to Adafruit IO.



To view the data that has been logged from the sensor, click on the graph next to the sensor name.



Here you can see the feed history and edit things about the feed such as the name, privacy, webhooks associated with the feed and more. If you want to learn more about how feeds work, [check out this page](https://adafru.it/10aZ).

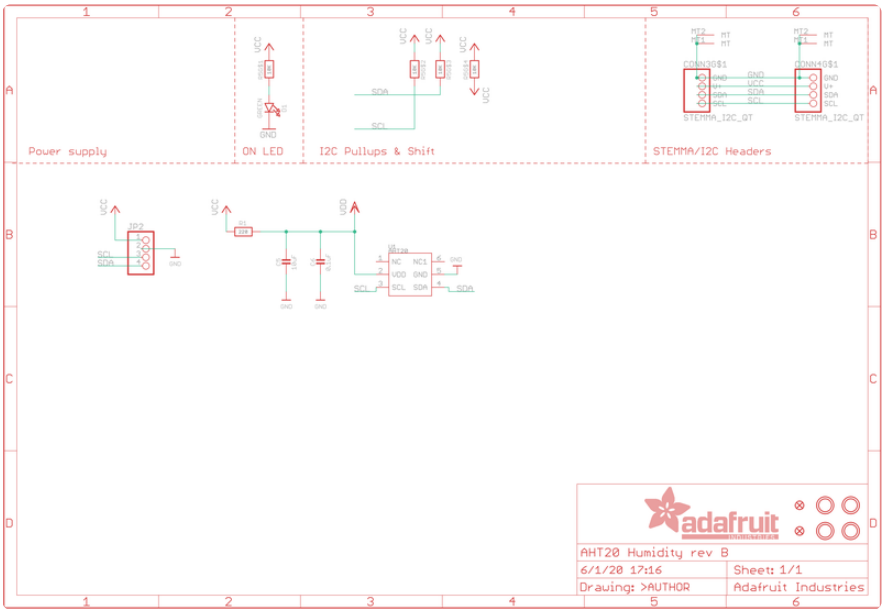


Downloads

Files

- [AHT20 Datasheet](https://adafru.it/19gE)
- [Fritzing object in the Adafruit Fritzing Library](https://adafru.it/LAn)
- [EagleCAD PCB files on GitHub](https://adafru.it/LAo)

Schematic



Fab Print

