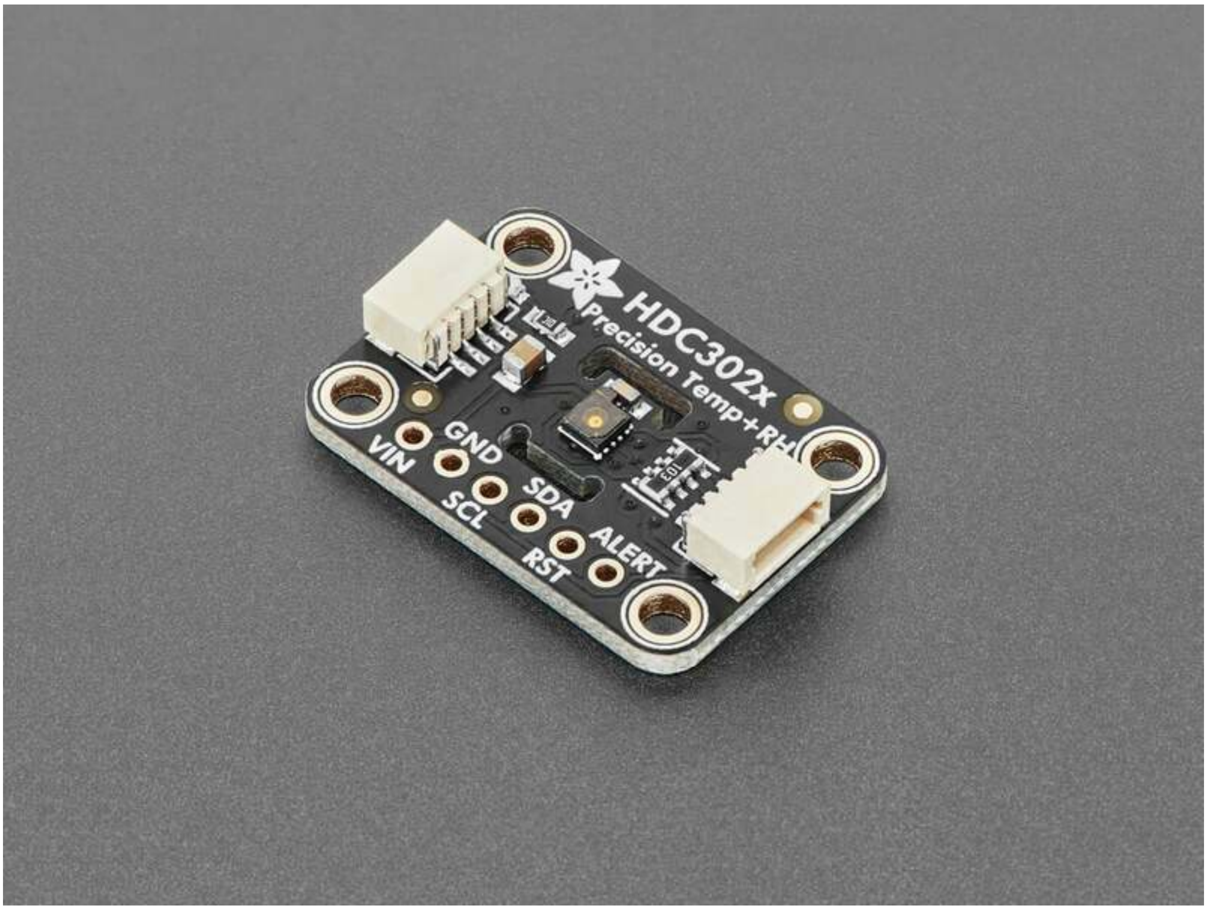
Adafruit HDC302x Precision Temperature & Humidity Sensor

Created by Liz Clark



https://learn.adafruit.com/adafruit-hdc3021-precision-temperature-humidity-sensor

Last updated on 2024-12-17 03:10:31 PM EST

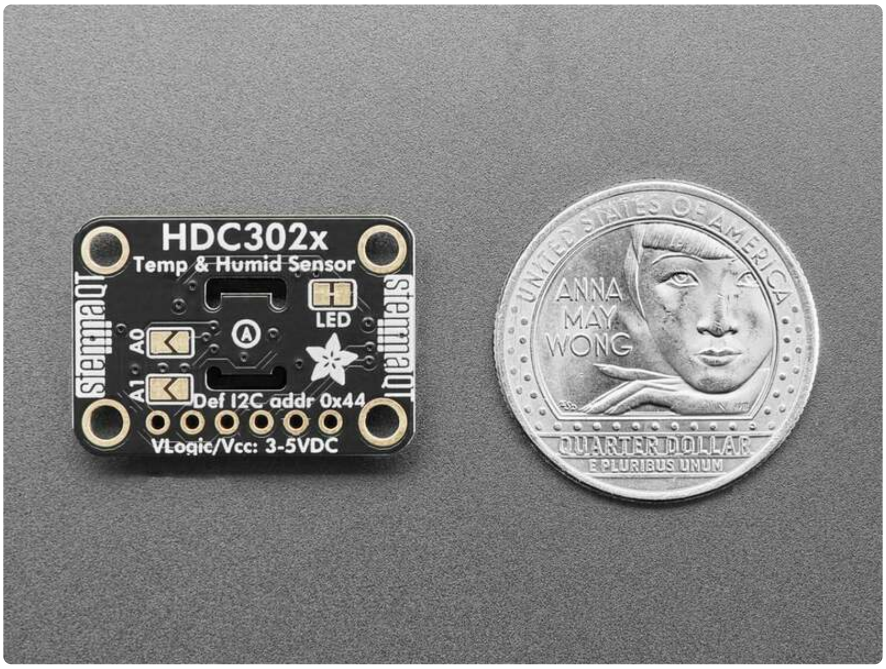
Table of Contents

Overview

As of September 23, 2024 – We've upgraded this sensor to now be the "HDC3022" rather than the "HDC3021". It is completely the same other than it now has a white PTFE filter cover that should not be removed and will keep your sensor nice and clean.

Adafruit stocks a large number of Temperature/Humidity sensors, so you're probably wondering why another one? Well, this **Adafruit HDC3021 Precision Temperature & Humidity Sensor** breakout features the highest accuracy & precision one we've see so far!

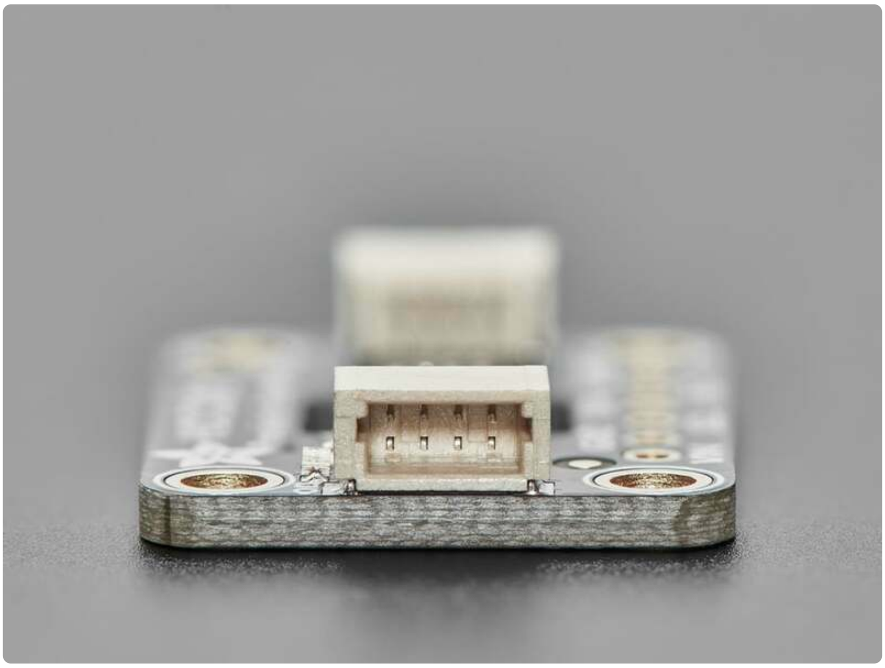
It uses the TI HDC302x series chip (https://adafru.it/1a4g) with typical 0.5% accuracy for the RH sensor (with 0.19% long-term drift) and ±0.1°C typical accuracy for the temperature sensor. All at a very nice price.



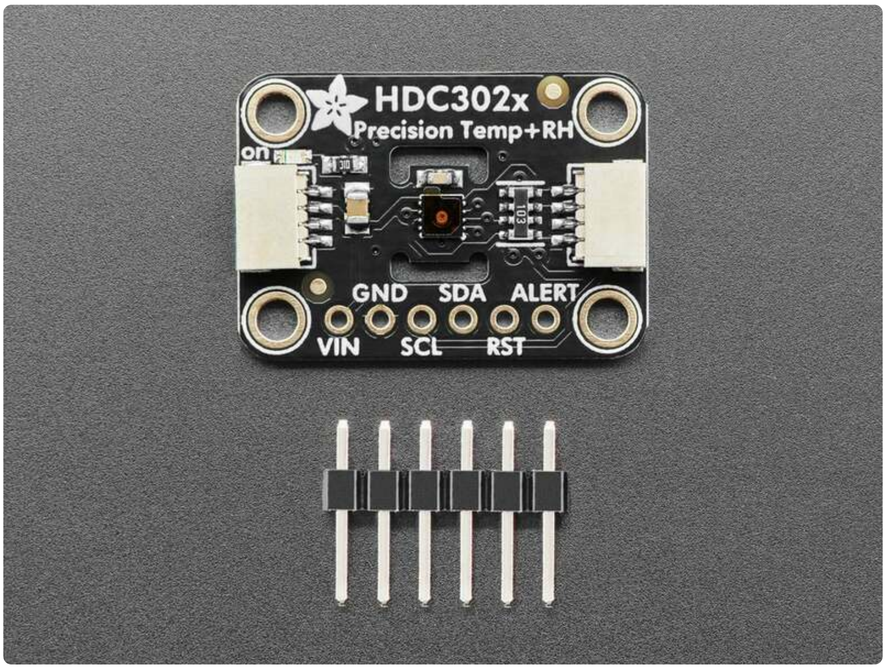
This sensor has a true I2C interface for easy interfacing with only two wires (plus power and ground!). That makes it preferable to the janky DHT11/DHT22 sensors.

The sensor is 5V safe, so you can use 1.8V to 5V power and logic levels without the need for shifters, so you can power and communicate with it using any microcontroller or microcomputer.

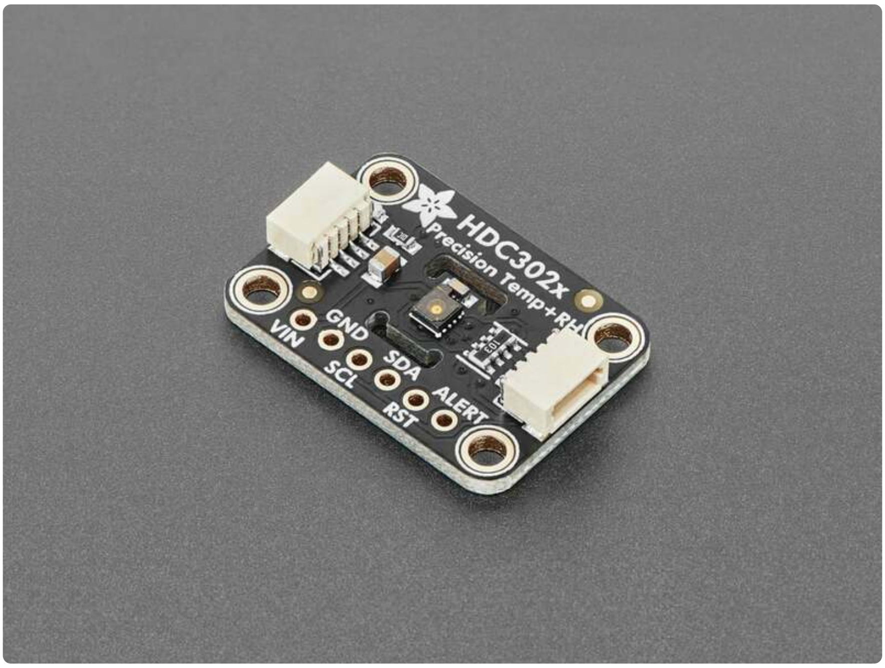
With 4 selectable I2C addresses, you can have multiple sensors on one bus - unlike many other fixed-address T/RH sensors.



We've written full-featured Arduino (https://adafru.it/1a4h) and CircuitPython / Python libraries (https://adafru.it/1a4i), so whether you have an 8-bit microcontroller or a quad-core Raspberry Pi, you will be able to quickly get data from the sensor for your environmental monitoring needs. The libraries can setup the sensor rate/power, heater, alerts, offsets, and read data or NIST tracking ID.



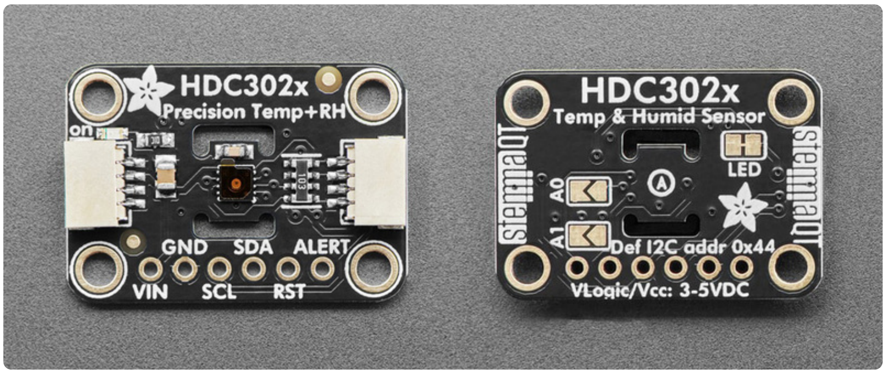
Such a lovely chip - so we spun up a breakout board with the HDC3021 and some supporting circuitry such as pullup resistors and capacitors. To make things even easier, we've included SparkFun Qwiic (https://adafru.it/Fpw) compatible STEMMA QT (https://adafru.it/Ft4) connectors for the I2C bus so you don't even need to solder! QT Cable is not included, but we have a variety in the shop (https://adafru.it/17VE).



If you prefer working on a breadboard, each order comes with one fully assembled and tested PCB breakout and a small piece of header. You'll need to solder the header onto the PCB, but it's fairly easy and takes only a few minutes even for a beginner.

Please note: this sensor comes with a protective polyamide sticker on top, please remove before usage!

Pinouts



The default I2C address is **0x44**.

As of September 23, 2024 – We've upgraded this sensor to now be the "HDC3022" rather than the "HDC3021". It is completely the same other than it now has a white PTFE filter cover that should not be removed and will keep your sensor nice and clean.

Power Pins

- **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 microcontroller's I2C clock line. This pin can use 3-5V logic, and there's a **10K pullup** on this pin.
- **SDA** - I2C data pin, connect to your microcontroller's I2C data line. This pin can use 3-5V logic, and there's a **10K pullup** on this pin.
- STEMMA QT **(https://adafru.it/Ft4)** - These connectors allow you to connect to dev boards with **STEMMA QT** (Qwiic) connectors or to other things with various associated accessories (https://adafru.it/Ft6).

Other Pins

- **RST** - This is the reset pin. If this pin is pulled low, it will reset the sensor.

- **ALERT** - This is the interrupt pin. It is a push-pull output, meaning that you can monitor the state of this pin for interrupts that can be configured over I2C.

I2C Address Jumpers

On the back of the board are **two address jumpers**, labeled **A0** and **A1,** between the STEMMA QT logo and cutouts for the HDC3021 sensor. These jumpers allow you to chain up to 4 of these boards on the same pair of I2C clock and data pins. To do so, you solder the jumpers "closed" by connecting the two pads.

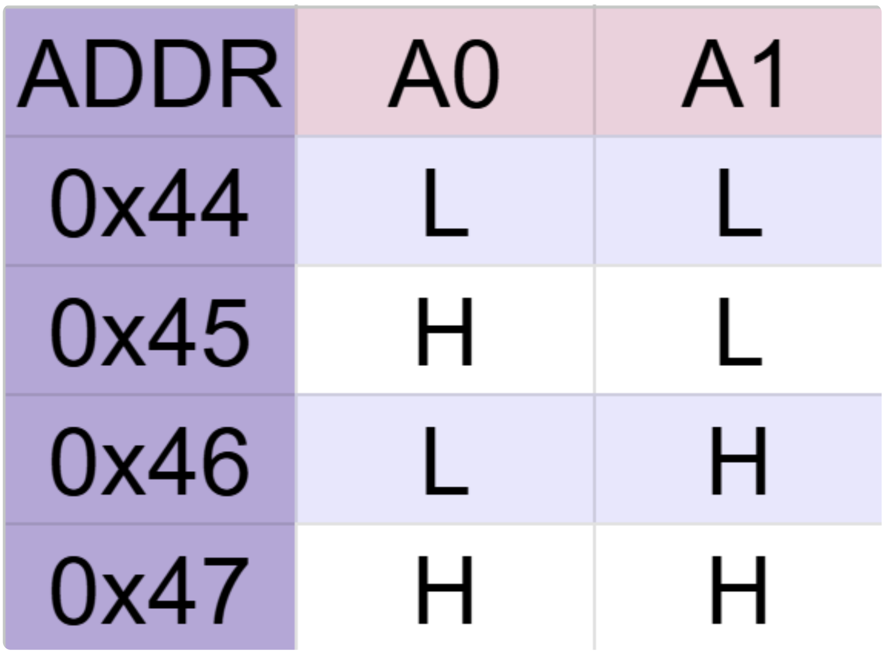
The default I2C address is **0x44**. The other address options can be calculated by “adding” the **A0/A1** to the base of **0x44**.

A0 sets the lowest bit with a value of **1** and **A1** sets the next bit with a value of **2**. The final address is **0x44** + **A1** + **A0** which would be **0x47**.

If only **A0** is soldered closed, the address is **0x44** + **1** = **0x45**

If only **A1** is soldered closed, the address is **0x44** + **2** = **0x46**

The table below shows all possible addresses, and whether the pin(s) should be high (closed) or low (open).



ADDR	A0	A1
0x44	L	L
0x45	H	L
0x46	L	H
0x47	H	H

Power LED and Jumper

- **Power LED -** In the upper left corner, above the STEMMA connector, on the front of the board, is the power LED, labeled **on**. It is a green LED.
- **LED jumper** - This jumper is located on the back of the board and is labeled **LED** on the board silk. Cut the trace on this jumper to cut power to the "**on**" LED.

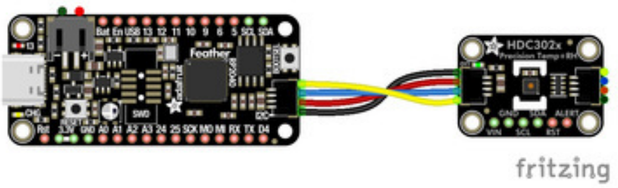
CircuitPython and Python

It's easy to use the **HDC3021** with Python or CircuitPython, and the Adafruit_CircuitPython_HDC302x (https://adafru.it/1a4i) module. This module allows you to easily write Python code to read temperature and humidity data from the sensor.

You can use this driver 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).

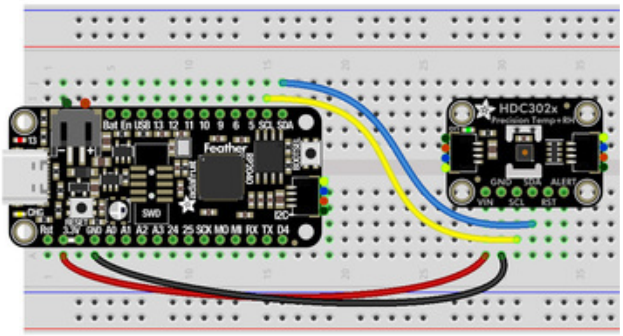
CircuitPython Microcontroller Wiring

First wire up the sensor to your board exactly as follows. The following is the sensor wired to a Feather RP2040 using the STEMMA connector:



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

The following is the sensor wired to a Feather RP2040 using a solderless breadboard:

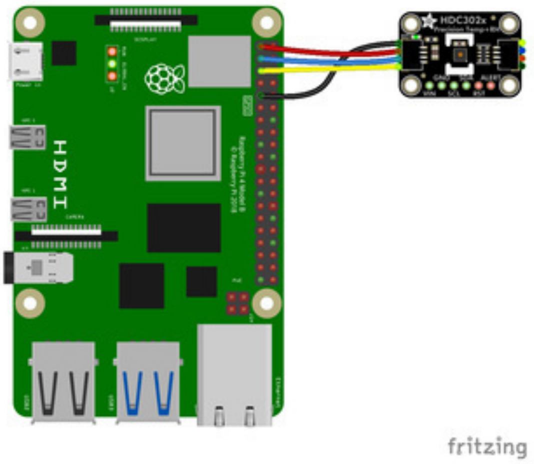


Board 3V to **sensor VIN (red wire)**
Board GND to **sensor GND (black wire)**
Board SCL to **sensor SCL (yellow wire)**
Board SDA to **sensor SDA (blue wire)**

Python Computer Wiring

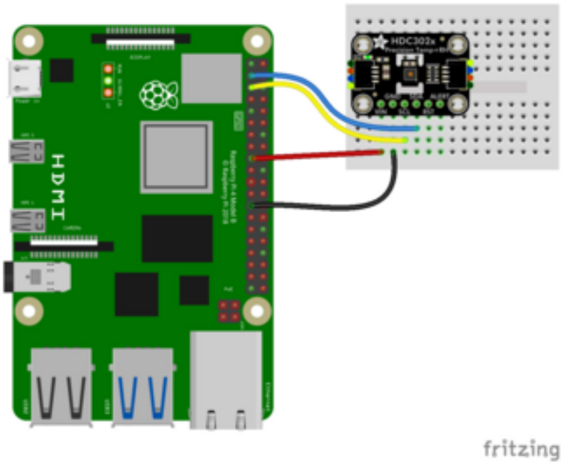
Since there are dozens of Linux computers/boards you can use, we will show wiring for Raspberry Pi. For other platforms, please visit the guide for CircuitPython on Linux to see whether your platform is supported (https://adafru.it/BSN).

Here's the Raspberry Pi wired with I2C using the STEMMA connector:



Pi 3V to **sensor VIN (red wire)**
Pi GND to **sensor GND (black wire)**
Pi SCL to **sensor SCL (yellow wire)**
Pi SDA to **sensor SDA (blue wire)**

Here's the Raspberry Pi wired with I2C using a solderless breadboard:



Pi 3V to **sensor VIN (red wire)**
Pi GND to **sensor GND (black wire)**
Pi SCL to **sensor SCL (yellow wire)**
Pi SDA to **sensor SDA (blue wire)**

Python Installation of HDC302x 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-hdc302x`

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!

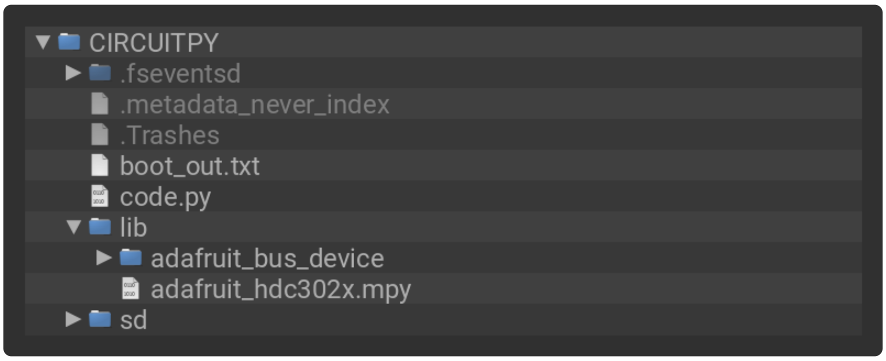
CircuitPython Usage

To use with CircuitPython, you need to first install the **Adafruit_CircuitPython_HDC302x** library, and its dependencies, into the **lib** folder on your **CIRCUITPY** drive. Then you need to update **code.py** with the example script.

Thankfully, we can do this in one go. In the example below, click the **Download Project Bundle** button below to download the necessary libraries and the **code.py** file in a zip file. Extract the contents of the zip file, and copy the **entire lib folder** and the **code.py** file to your **CIRCUITPY** drive.

Your **CIRCUITPY/lib** folder should contain the following folder and file:

- **adafruit_bus_device/**
- **adafruit_hdc302x.mpy**



```
▼ CIRCUITPY
  ▶ .fseventsd
    .metadata_never_index
    .Trashes
    boot_out.txt
    code.py
  ▼ lib
    ▶ adafruit_bus_device
      adafruit_hdc302x.mpy
  ▶ sd
```

Python Usage

Once you have the library `pip3` installed on your computer, copy or download the following example to your computer, and run the following, replacing **code.py** with whatever you named the file:

`python3 code.py`

Example Code

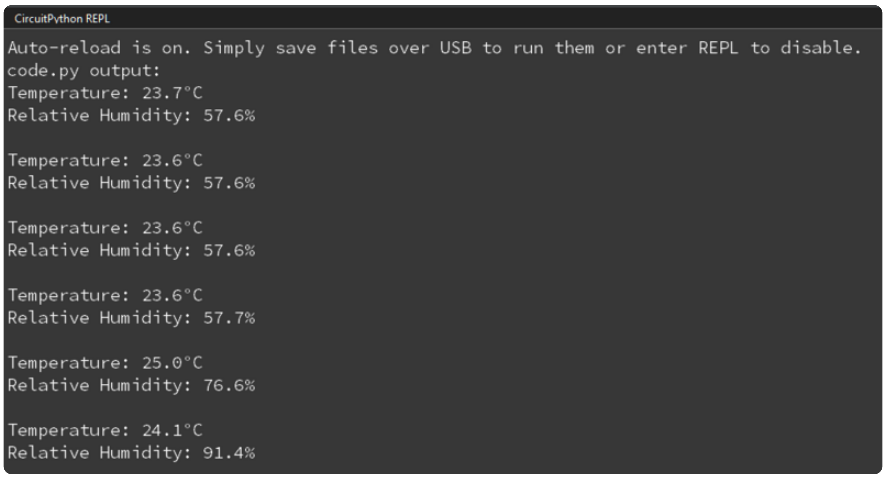
If running CircuitPython: Once everything is saved to the **CIRCUITPY** drive, connect to the serial console (https://adafru.it/Bec) to see the data printed out!

If running Python: The console output will appear wherever you are running Python.

```
# SPDX-FileCopyrightText: 2024 Liz Clark for Adafruit Industries
#
# SPDX-License-Identifier: MIT
```

```
"""HDC302x simple test"""

import time
import board
import adafruit_hdc302x

i2c = board.I2C()
sensor = adafruit_hdc302x.HDC302x(i2c)

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

The HDC3021 is instantiated over I2C. Then in the loop, the temperature and humidity readings are printed to the serial monitor every two seconds.



```
Auto-reload is on. Simply save files over USB to run them or enter REPL to disable.
code.py output:
Temperature: 23.7°C
Relative Humidity: 57.6%

Temperature: 23.6°C
Relative Humidity: 57.6%

Temperature: 23.6°C
Relative Humidity: 57.6%

Temperature: 23.6°C
Relative Humidity: 57.7%

Temperature: 25.0°C
Relative Humidity: 76.6%

Temperature: 24.1°C
Relative Humidity: 91.4%
```

Python Docs

Python Docs (https://adafru.it/1a4b)

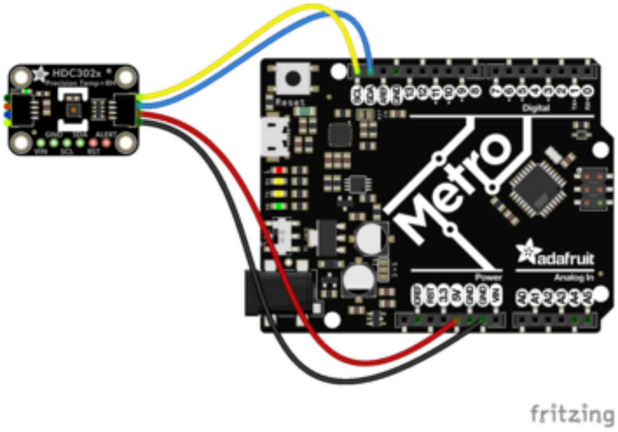
Arduino

Using the HDC3021 breakout with Arduino involves wiring up the breakout to your Arduino-compatible microcontroller, installing the Adafruit_HDC302x (https://adafru.it/1a4h) library, and running the provided example code.

Wiring

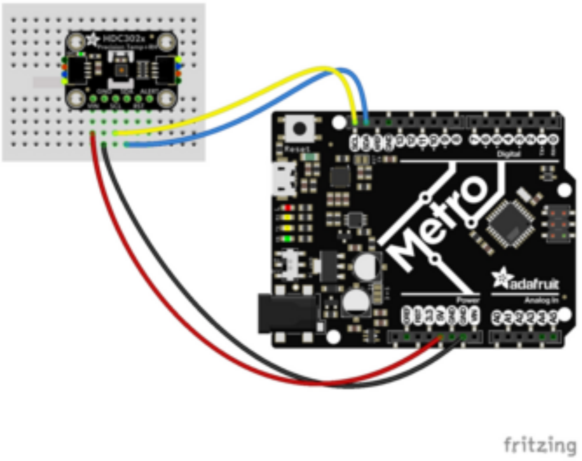
Wire as shown for a **5V** board like an Uno. If you are using a **3V** board, like an Adafruit Feather, wire the board's 3V pin to the breakout VIN.

Here is an Adafruit Metro wired up to the sensor using the STEMMA QT connector.



Board 5V to **sensor VIN (red wire)**
Board GND to **sensor GND (black wire)**
Board SCL to **sensor SCL (yellow wire)**
Board SDA to **sensor SDA (blue wire)**

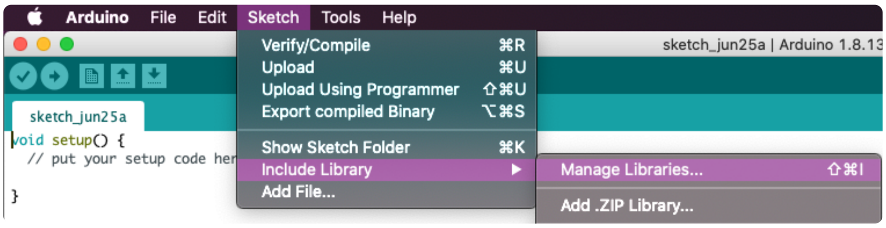
Here is an Adafruit Metro wired up using a solderless breadboard:



Board 5V to **sensor VIN (red wire)**
Board GND to **sensor GND (black wire)**
Board SCL to **sensor SCL (yellow wire)**
Board SDA to **sensor SDA (blue wire)**

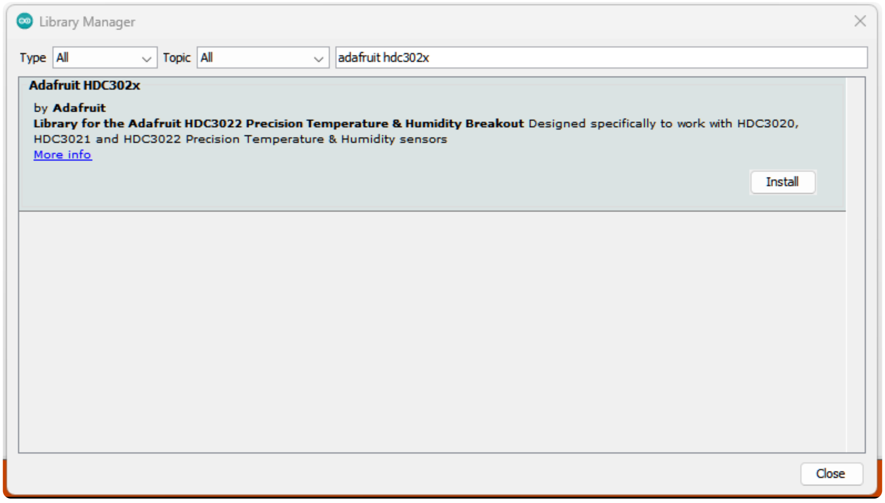
Library Installation

You can install the **Adafruit_HDC302x** library for Arduino using the Library Manager in the Arduino IDE.



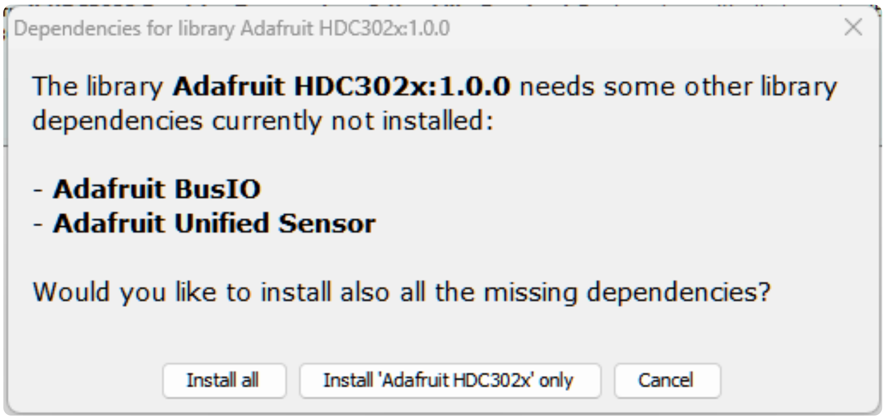
```
void setup() {
  // put your setup code her
}
```

Click the **Manage Libraries ...** menu item, search for **Adafruit_HDC302x**, and select the **Adafruit HDC302x** library:



Adafruit HDC302x
by **Adafruit**
Library for the Adafruit HDC3022 Precision Temperature & Humidity Breakout Designed specifically to work with HDC3020, HDC3021 and HDC3022 Precision Temperature & Humidity sensors
More info

If asked about dependencies, click "Install all".



Dependencies for library Adafruit HDC302x:1.0.0

The library **Adafruit HDC302x:1.0.0** needs some other library dependencies currently not installed:

- **Adafruit BusIO**
- **Adafruit Unified Sensor**

Would you like to install also all the missing dependencies?

Install all
Install 'Adafruit HDC302x' only
Cancel

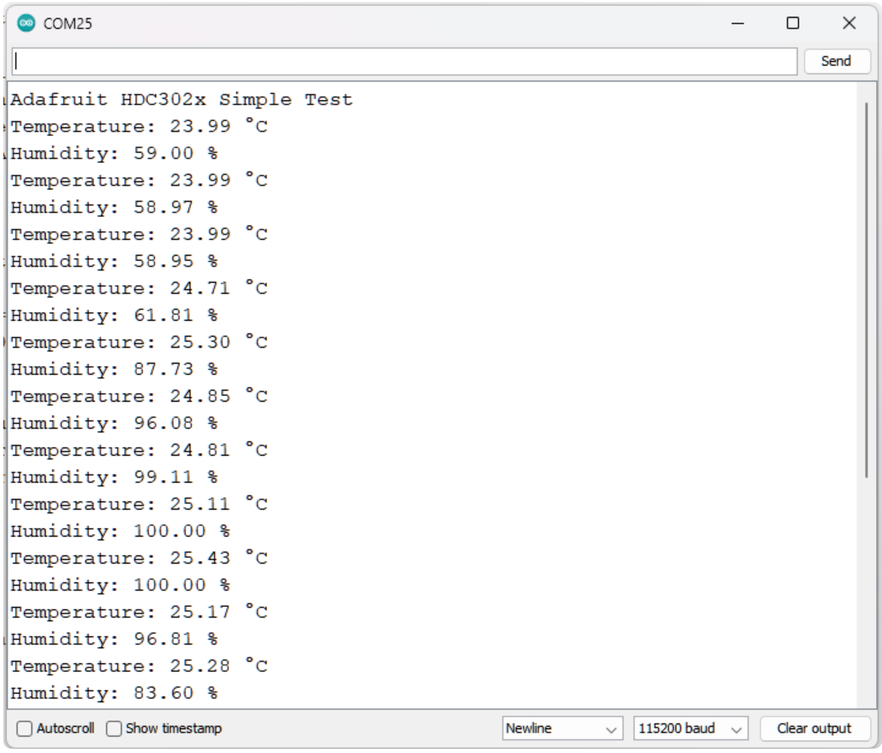
If the "Dependencies" window does not come up, then you already have the dependencies installed.

If the dependencies are already installed, make sure you update them through the Arduino Library Manager before loading the example!

Example Code

```
#include <Adafruit_HDC302x.h>

Adafruit_HDC302x hdc = Adafruit_HDC302x();

void setup() {
  Serial.begin(115200);

  Serial.println("Adafruit HDC302x Simple Test");

  if (! hdc.begin(0x44, &Wire)) {
    Serial.println("Could not find sensor?");
    while (1);
  }
  delay(1000);
}

void loop() {
  double temp = 0.0;
  double RH = 0.0;

  hdc.readTemperatureHumidityOnDemand(temp, RH, TRIGGERMODE_LP0);

  Serial.print("Temperature: ");
  Serial.print(temp);
  Serial.println(" °C");

  Serial.print("Humidity: ");
  Serial.print(RH);
  Serial.println(" %");
  delay(2000);

}
```

Upload the sketch to your board and open up the Serial Monitor (**Tools -> Serial Monitor**) at 115200 baud. After the HDC3021 is recognized over I2C, you'll see the temperature and humidity readings print to the Serial Monitor every two seconds.

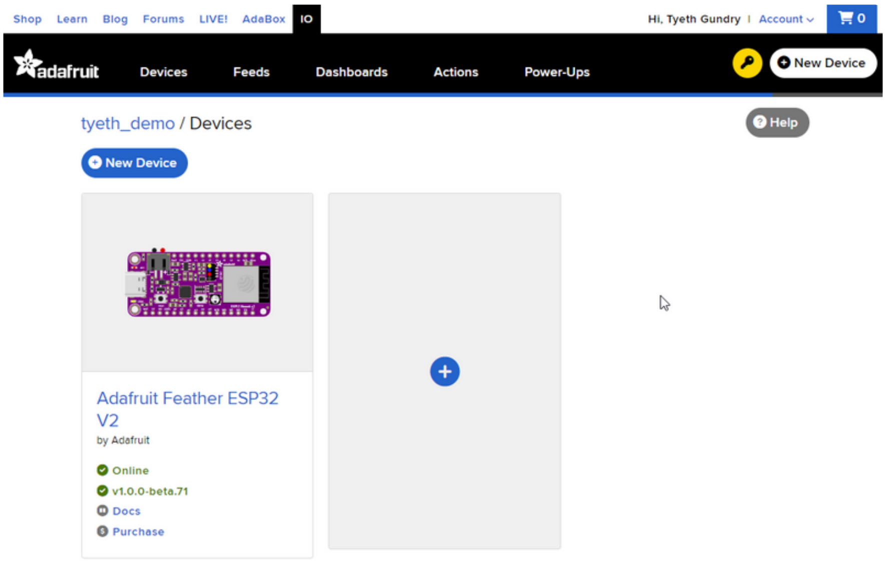


```
Adafruit HDC302x Simple Test
Temperature: 23.99 °C
Humidity: 59.00 %
Temperature: 23.99 °C
Humidity: 58.97 %
Temperature: 23.99 °C
Humidity: 58.95 %
Temperature: 24.71 °C
Humidity: 61.81 %
Temperature: 25.30 °C
Humidity: 87.73 %
Temperature: 24.85 °C
Humidity: 96.08 %
Temperature: 24.81 °C
Humidity: 99.11 %
Temperature: 25.11 °C
Humidity: 100.00 %
Temperature: 25.43 °C
Humidity: 100.00 %
Temperature: 25.17 °C
Humidity: 96.81 %
Temperature: 25.28 °C
Humidity: 83.60 %
```

Arduino Docs

Arduino Docs (https://adafru.it/1a4c)

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.

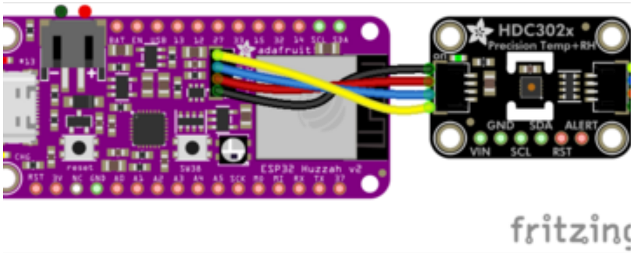
Quickstart: Adafruit IO WipperSnapper

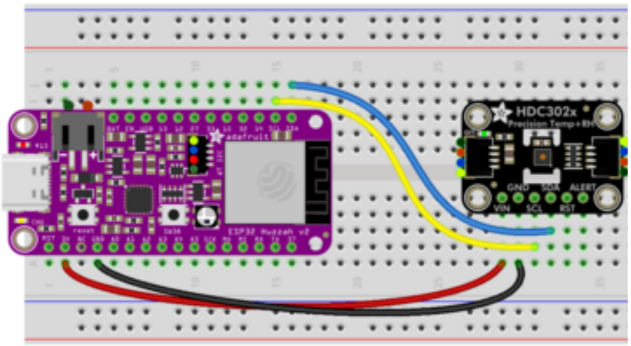
https://adafru.it/Vfd

Wiring

First, wire up an HDC302x (HDC3021/HDC3022) to your board exactly as follows. Here is an example of the HDC3021 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)



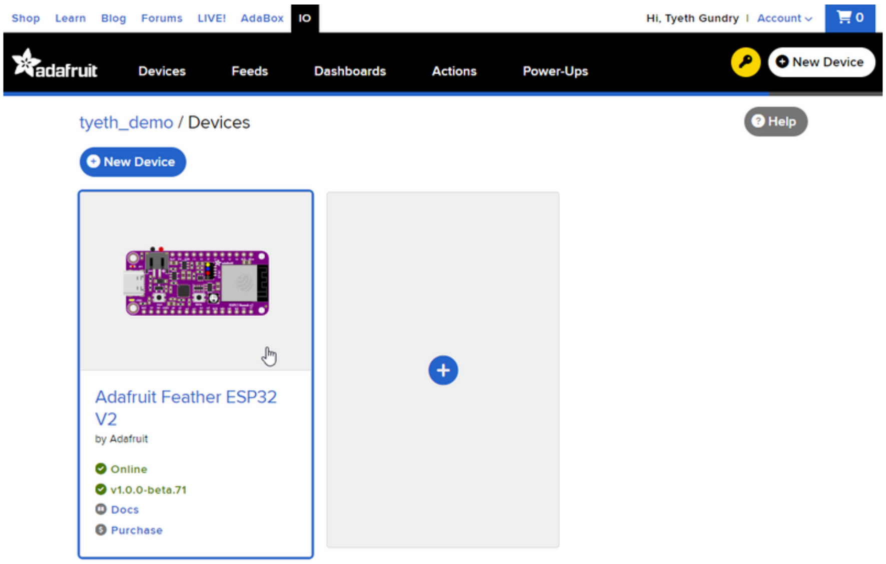


Board 3V to **sensor VIN (red wire on STEMMA QT)**
Board GND to **sensor GND (black wire on STEMMA QT)**
Board SCL to **sensor SCL (yellow wire on STEMMA QT)**
Board SDA to **sensor SDA (blue wire on STEMMA QT)**

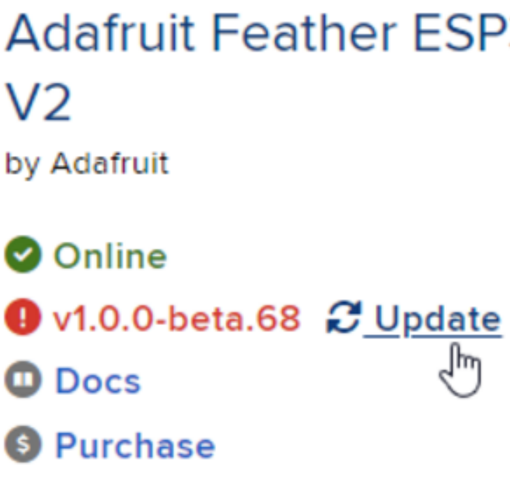
Usage

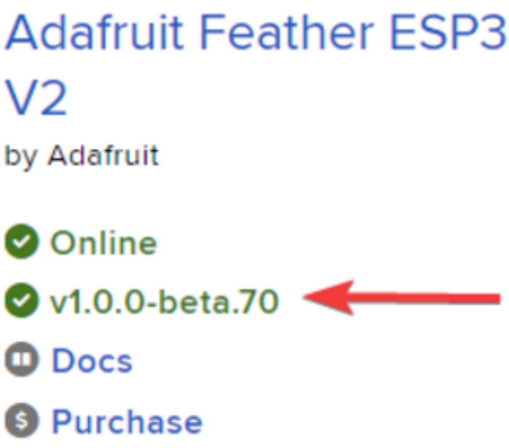
Connect your board to Adafruit IO Wippersnapper and navigate to the WipperSnapper board list (https://adafru.it/TAu).

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) first.



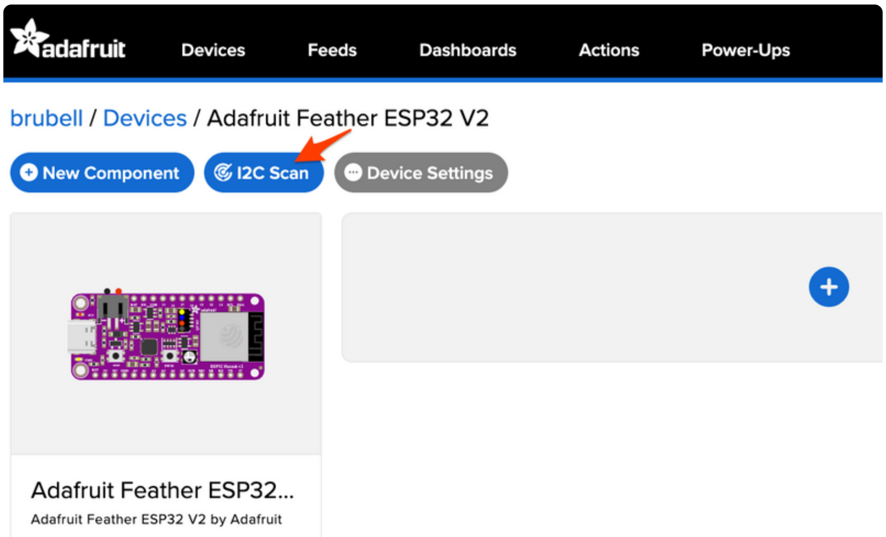


On the device page, quickly **check that you're running the latest version of the WipperSnapper firmware**.

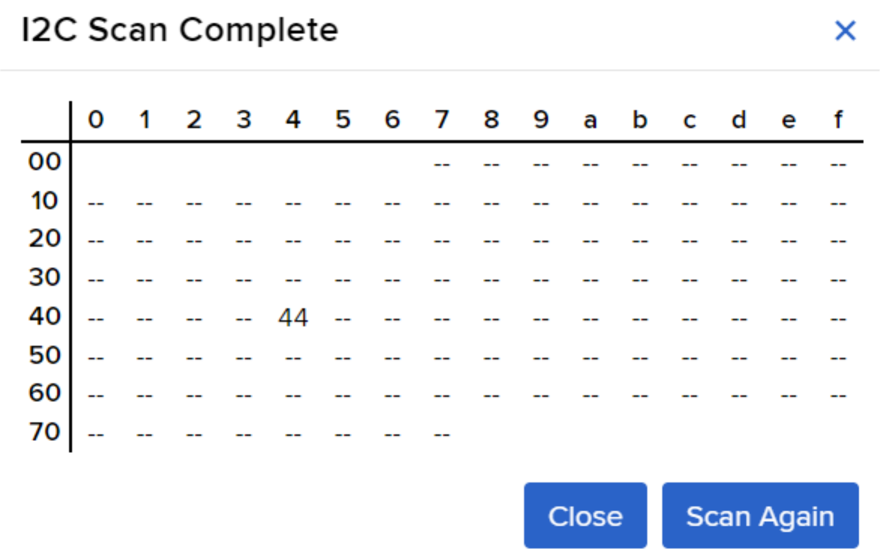
The device tile on the left indicates the version number of the firmware running on the connected board.

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) on your board before continuing.

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


adafruit
Devices
Feeds
Dashboards
Actions
Power-Ups
brubell / Devices / Adafruit Feather ESP32 V2
New Component
I2C Scan
Device Settings
+
Adafruit Feather ESP32...
Adafruit Feather ESP32 V2 by Adafruit

You should see the HDC302x's default I2C address of `0x44` pop-up in the I2C scan list.



	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00								--	--	--	--	--	--	--	--	--
10	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
20	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
30	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
40	--	--	--	--	44	--	--	--	--	--	--	--	--	--	--	--
50	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
60	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
70	--	--	--	--	--	--	--	--								

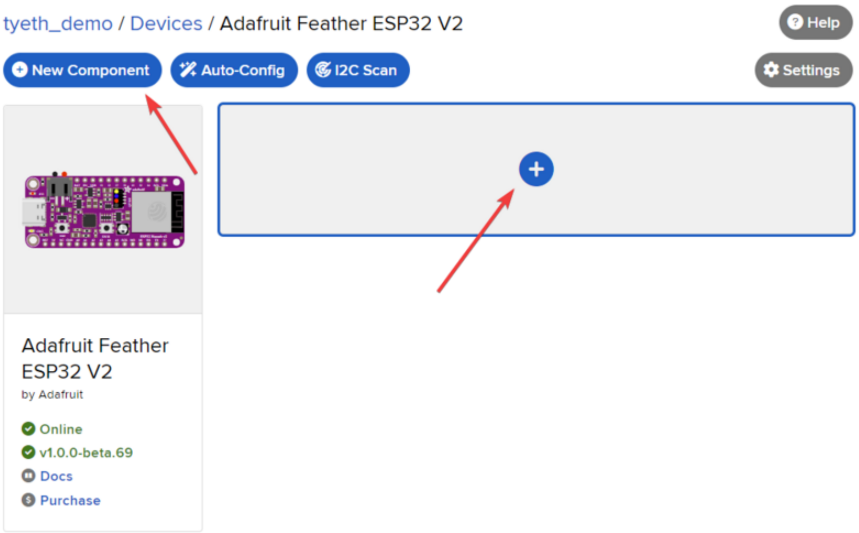
I don't see the sensor's I2C address listed!

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

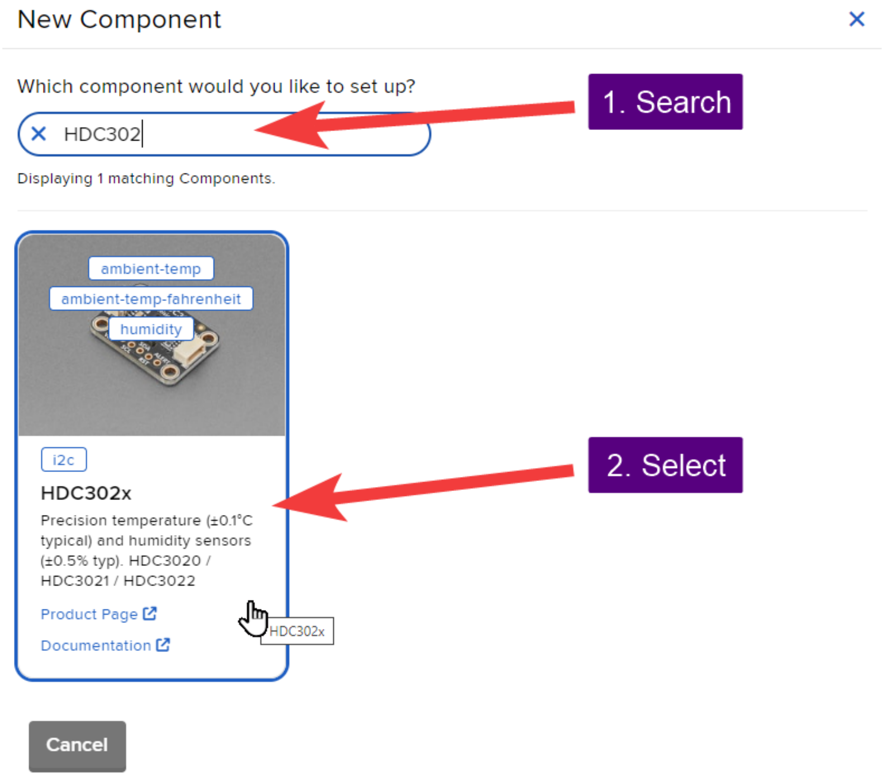
Then, reset the board and let it re-connect to Adafruit IO WipperSnapper.

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 `HDC302` into the search bar, then select the **HDC302x** component.



Which component would you like to set up?

HDC302

Displaying 1 matching Components.

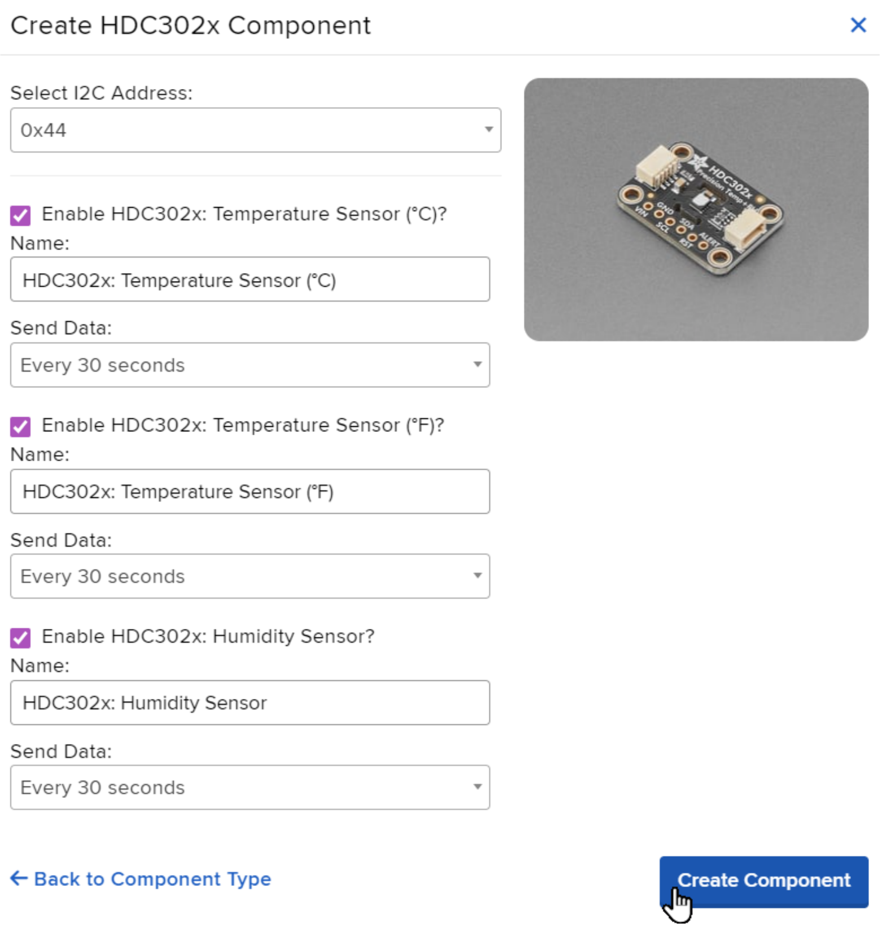
HDC302x

Precision temperature (±0.1°C typical) and humidity sensors (±0.5% typ). HDC3020 / HDC3021 / HDC3022

On the component configuration page, the HDC302x'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 HDC302x 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.



Create HDC302x Component

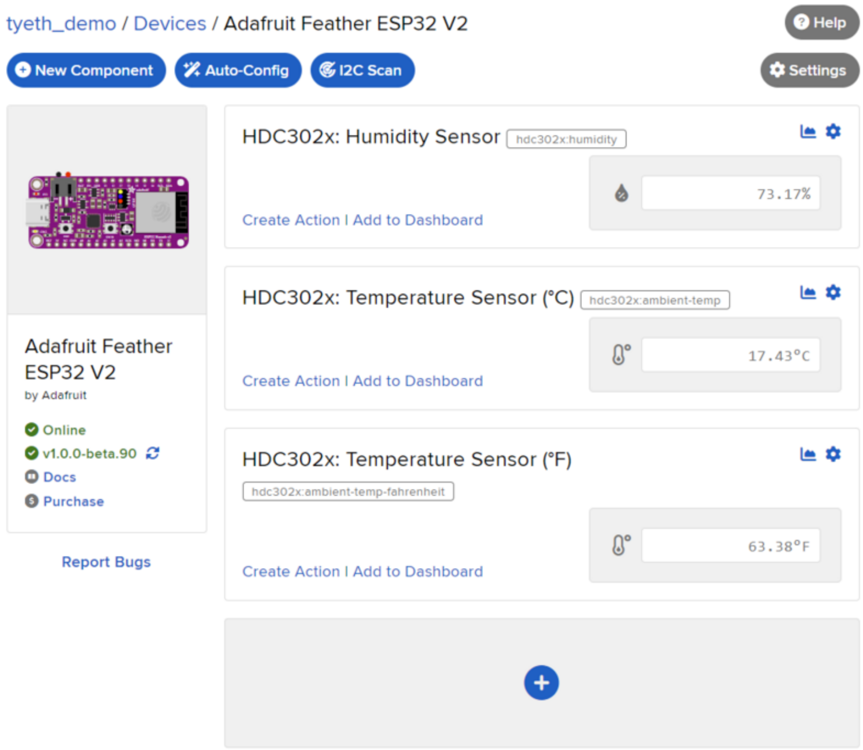
Select I2C Address:
0x44

☑ Enable HDC302x: Temperature Sensor (°C)?
Name:
HDC302x: Temperature Sensor (°C)
Send Data:
Every 30 seconds

☑ Enable HDC302x: Temperature Sensor (°F)?
Name:
HDC302x: Temperature Sensor (°F)
Send Data:
Every 30 seconds

☑ Enable HDC302x: Humidity Sensor?
Name:
HDC302x: Humidity Sensor
Send Data:
Every 30 seconds

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.



tyeth_demo / Devices / Adafruit Feather ESP32 V2

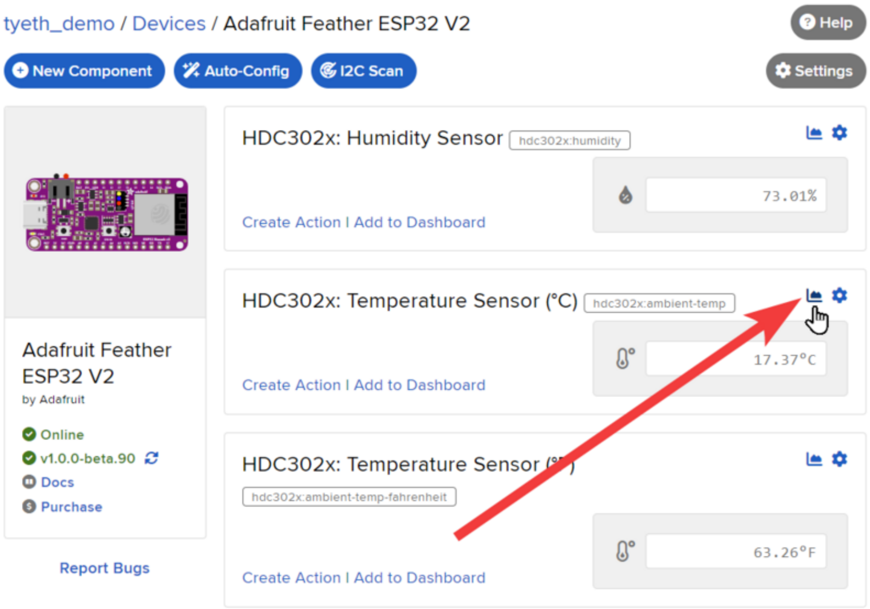
Adafruit Feather ESP32 V2
by Adafruit
Online
v1.0.0-beta.90
Docs
Purchase

HDC302x: Humidity Sensor hdc302x:humidity
73.17%
Create Action | Add to Dashboard

HDC302x: Temperature Sensor (°C) hdc302x:ambient-temp
17.43°C
Create Action | Add to Dashboard

HDC302x: Temperature Sensor (°F) hdc302x:ambient-temp-fahrenheit
63.38°F
Create Action | Add to Dashboard

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

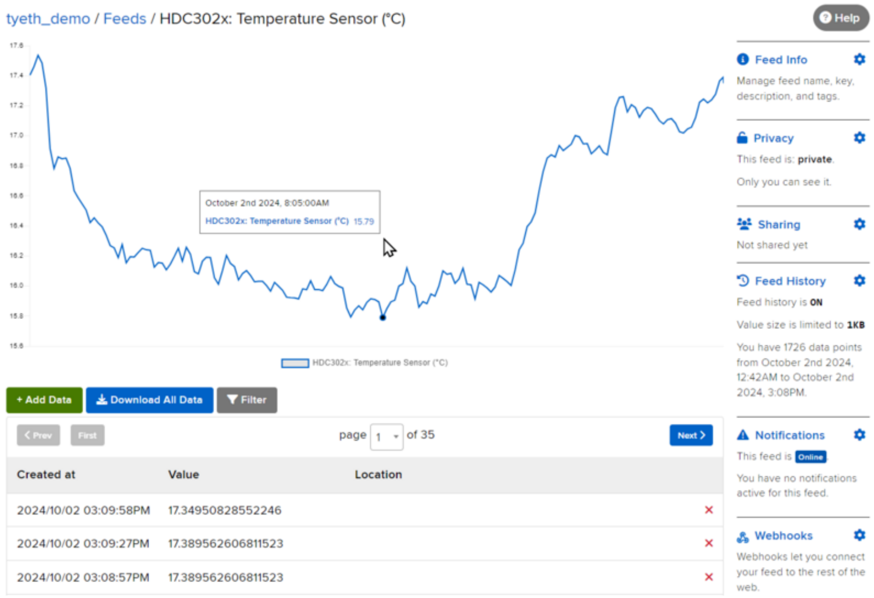

tyeth_demo / Devices / Adafruit Feather ESP32 V2
Help
New Component
Auto-Config
I2C Scan
Settings
Adafruit Feather ESP32 V2
by Adafruit
Online
v1.0.0-beta.90
Docs
Purchase
Report Bugs
Create Action | Add to Dashboard

HDC302x: Humidity Sensor hdc302x:humidity — 73.01%

HDC302x: Temperature Sensor (°C) hdc302x:ambient-temp — 17.37°C

HDC302x: Temperature Sensor (°F) hdc302x:ambient-temp-fahrenheit — 63.26°F

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).


tyeth_demo / Feeds / HDC302x: Temperature Sensor (°C)
Help
17.6
17.4
17.2
17.0
16.8
16.6
16.4
16.2
16.0
15.8
15.6
October 2nd 2024, 8:05:00AM
HDC302x: Temperature Sensor (°C) 15.79
HDC302x: Temperature Sensor (°C)
+ Add Data
Download All Data
Filter
Prev
First
Next
Feed Info
Privacy
Sharing
Feed History
Notifications
Webhooks

Manage feed name, key, description, and tags.

This feed is: **private**. Only you can see it.

Not shared yet

Feed history is **ON**

Value size is limited to **1KB**

You have 1726 data points from October 2nd 2024, 12:42AM to October 2nd 2024, 3:08PM.

This feed is Online.

You have no notifications active for this feed.

Webhooks let you connect your feed to the rest of the web.

page 1 of 35

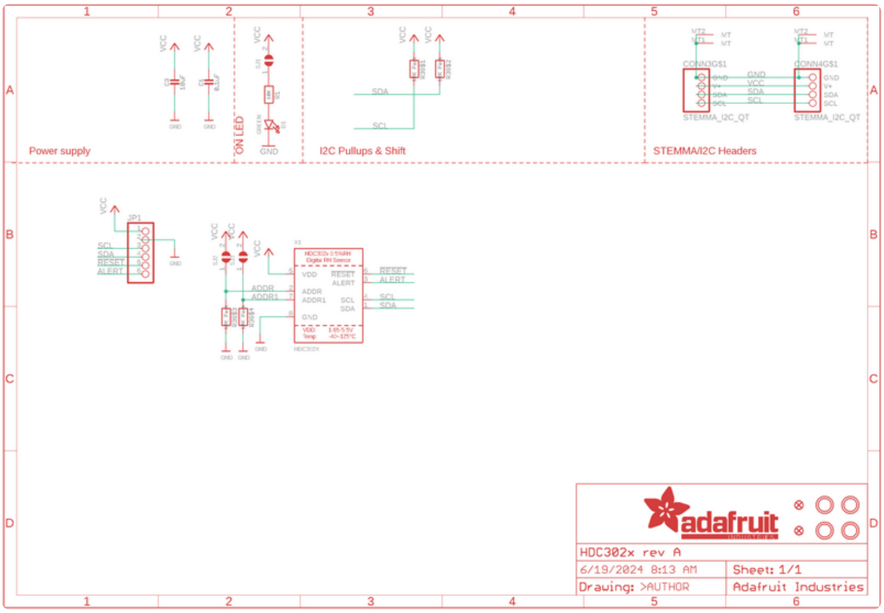
Created at	Value	Location	
2024/10/02 03:09:58PM	17.34950828552246		×
2024/10/02 03:09:27PM	17.389562606811523		×
2024/10/02 03:08:57PM	17.389562606811523		×

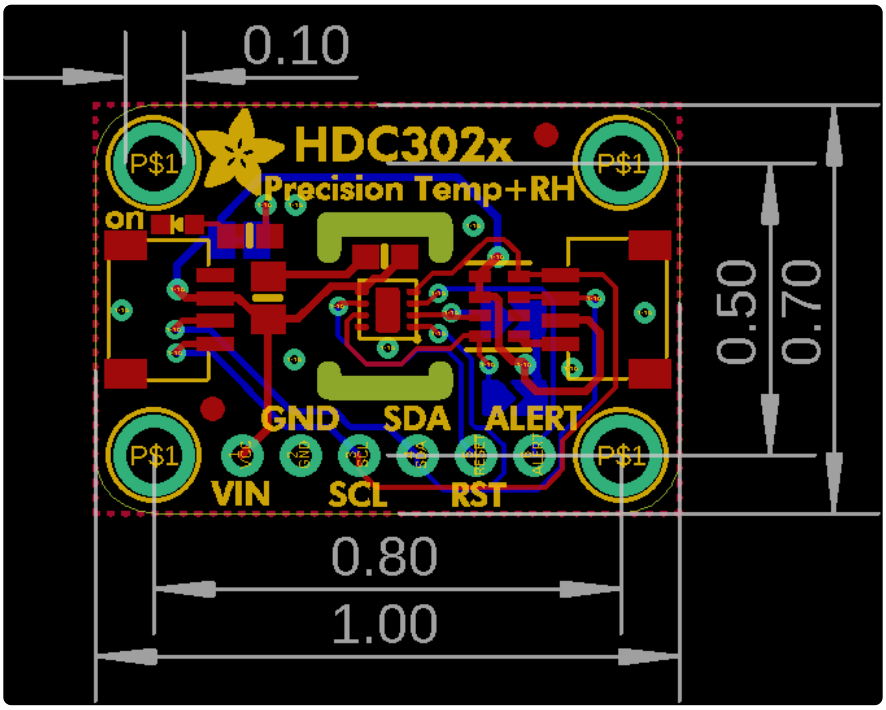
Downloads

Files

- HDC3021 Datasheet (https://adafru.it/1a4j)
- EagleCAD PCB Files on GitHub (https://adafru.it/1a4k)
- Fritzing object in the Adafruit Fritzing Library (https://adafru.it/1a4l)

Schematic and Fab Print


Power supply
ON LED
I2C Pullups & Shift
STEMMA/I2C Headers
SDA
SCL
VCC
GND
JP1
SCL
SDA
RESET
ALERT
ADDR
ADDR1
RESET
ALERT
SCL
SDA
VDD
GND
HDC302x
STEMMA_I2C_QT
adafruit
HDC302x rev A
6/19/2024 8:13 AM
Sheet: 1/1
Drawing: >AUTHOR
Adafruit Industries


0.10
HDC302x
Precision Temp+RH
on
P$1
P$1
P$1
P$1
GND
SDA
ALERT
VIN
SCL
RST
0.50
0.70
0.80
1.00