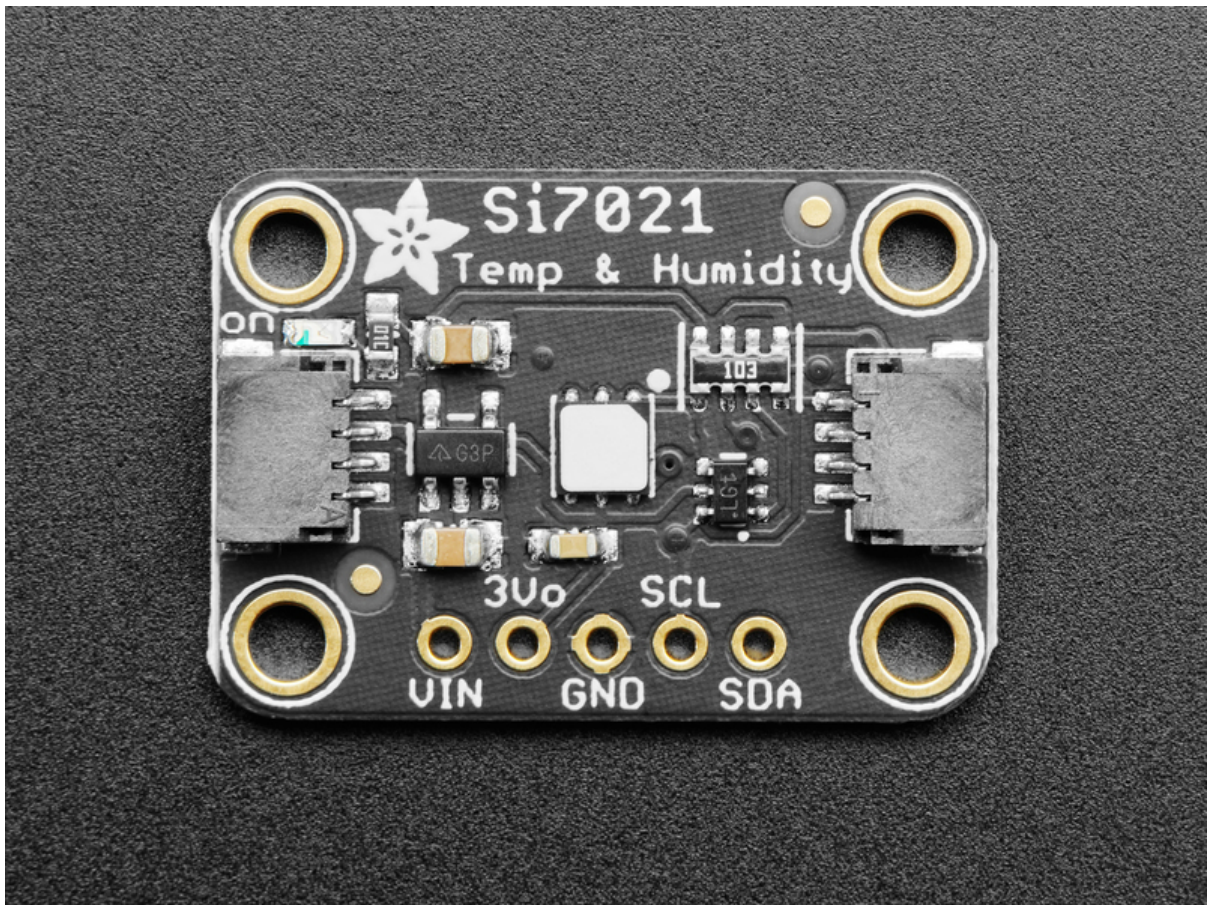




Adafruit Si7021 Temperature + Humidity Sensor

Created by lady ada



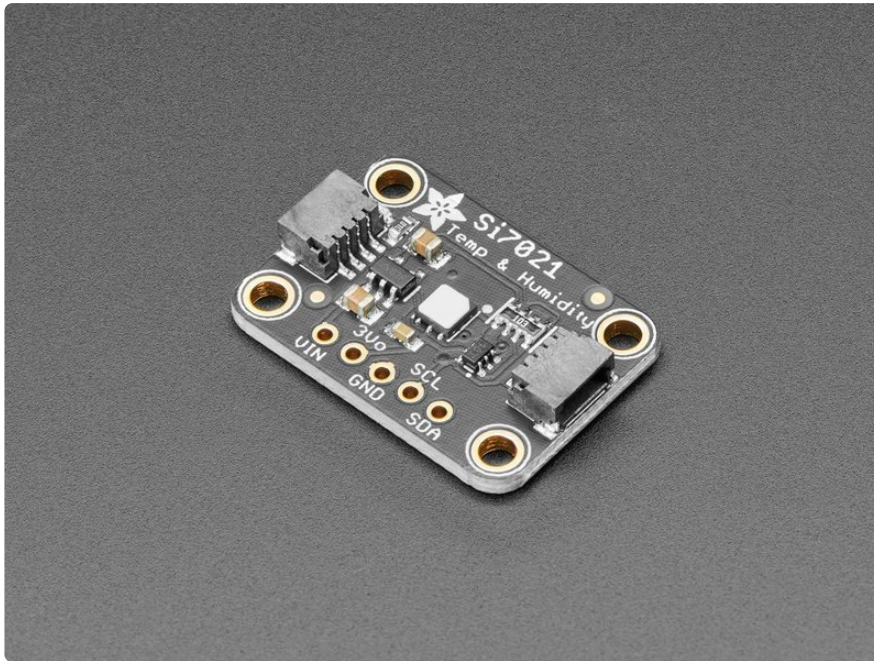
<https://learn.adafruit.com/adafruit-si7021-temperature-plus-humidity-sensor>

Last updated on 2024-06-03 02:00:08 PM EDT

Table of Contents

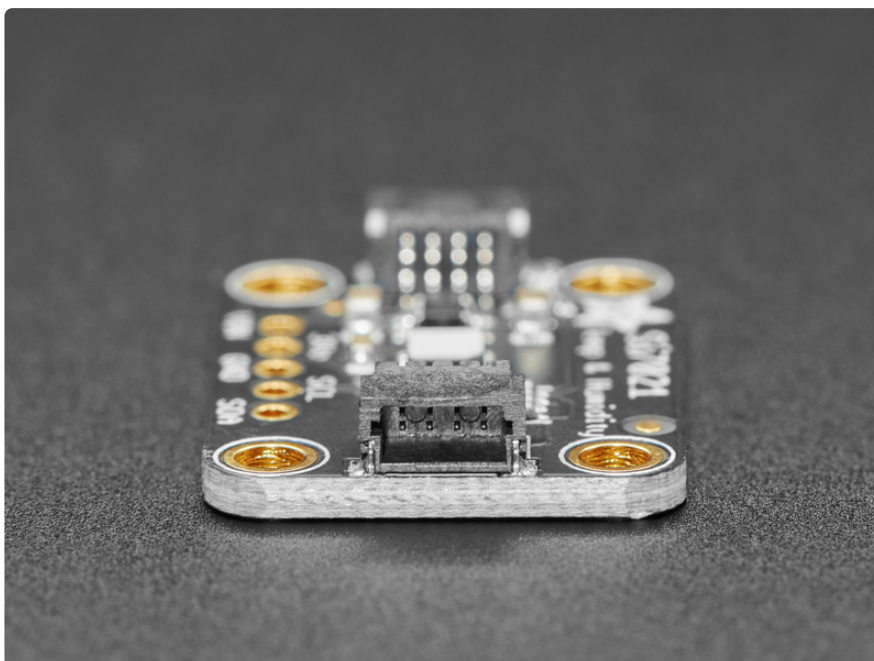
Overview	3
Pinouts	5
<ul style="list-style-type: none">• Power Pins:• I2C Logic pins:	
Assembly	7
<ul style="list-style-type: none">• Prepare the header strip:• Add the breakout board:• And Solder!	
Arduino Code	9
<ul style="list-style-type: none">• Install Adafruit_Si7021 library• Load Demo• Library Reference	
Python & CircuitPython	13
<ul style="list-style-type: none">• CircuitPython Microcontroller Wiring• Python Computer Wiring• CircuitPython Installation of Si7021 Library• Python Installation of Si7021 Library• CircuitPython & Python Usage• Full Example Code	
Python Library Docs	18
WipperSnapper	19
<ul style="list-style-type: none">• What is WipperSnapper• Wiring• Usage	
Downloads	25
<ul style="list-style-type: none">• Files & Datasheets• Schematic and Fabrication Print STEMMA QT Version• Schematic and Fabrication Print Original Version	

Overview

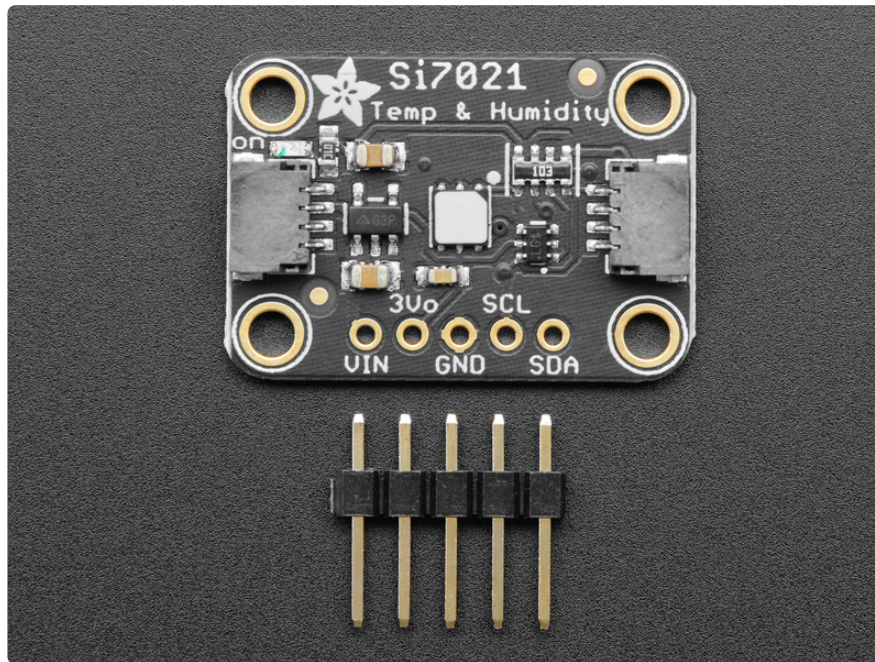


It's summer and you're sweating and your hair's all frizzy and all you really want to know is why the weatherman said this morning that today's relative humidity would max out at a perfectly reasonable 42% when it feels more like 77%. Enter the **Si7021 Temperature + Humidity Sensor** - the best way to prove the weatherman wrong!

This lovely sensor for Silicon labs has $\pm 3\%$ relative humidity measurements with a range of 0–80% RH, and $\pm 0.4\text{ }^\circ\text{C}$ temperature accuracy at a range of -10 to +85 $^\circ\text{C}$. Great for all of your environmental sensing projects. It uses I2C for data transfer so it works with a wide range of microcontrollers.

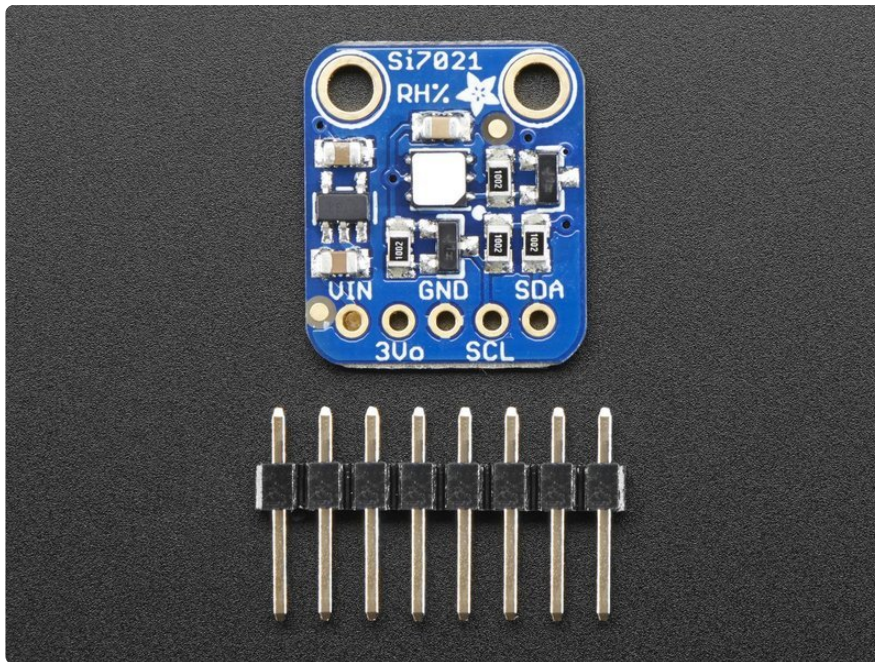


To get you going fast, we spun up a custom made PCB in the [STEMMA QT form factor](https://adafru.it/LBQ) (<https://adafru.it/LBQ>), making them easy to interface with. The [STEMMA QT connectors](https://adafru.it/JqB) (<https://adafru.it/JqB>) on either side are compatible with the [SparkFun Qwiic](https://adafru.it/Fpw) (<https://adafru.it/Fpw>) I2C connectors. This allows you to make solderless connections between your development board and the LIS331s or to chain them with a wide range of other sensors and accessories using a [compatible cable](https://adafru.it/JnB) (<https://adafru.it/JnB>). We've of course broken out all the pins to standard headers and added a voltage regulator and level shifting so allow you to use it with either 3.3V or 5V systems such as the Raspberry Pi + Feather series or Arduino Uno respectively.



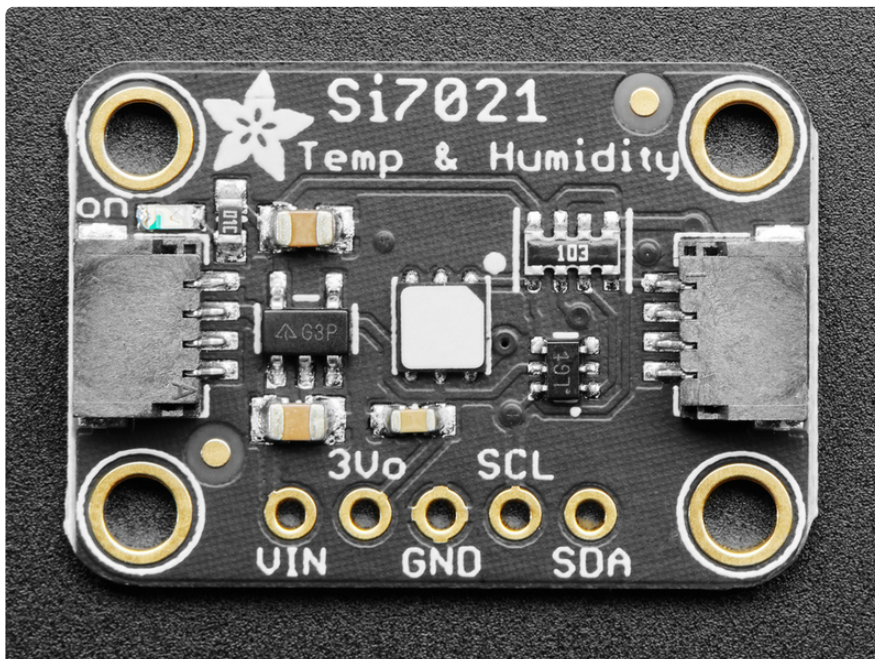
There's a PTFE filter to keep the sensor clean, that's the white flat thing on top. Also comes with some pin header. Some light soldering is required to attach the header for use in a breadboard but it's easy to do.

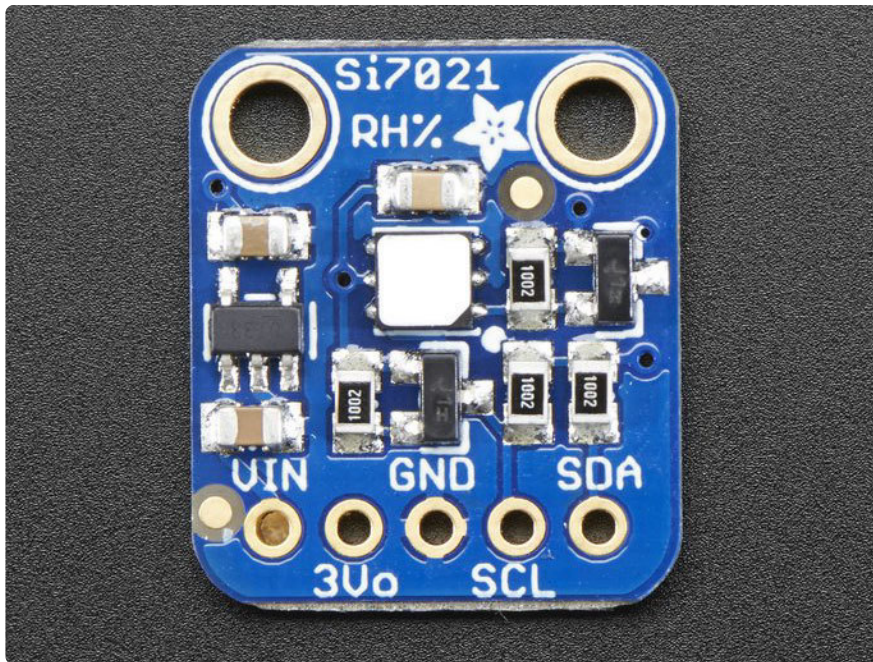
There are two versions of this board - the STEMMA QT version shown above, and the original header-only version shown below. Code works the same on both!



Pinouts

The Si7021 is a I2C sensor. That means it uses the two I2C data/clock wires available on most microcontrollers, and can share those pins with other sensors as long as they don't have an address collision. For future reference, the I2C address is **0x40** and you can't change it!





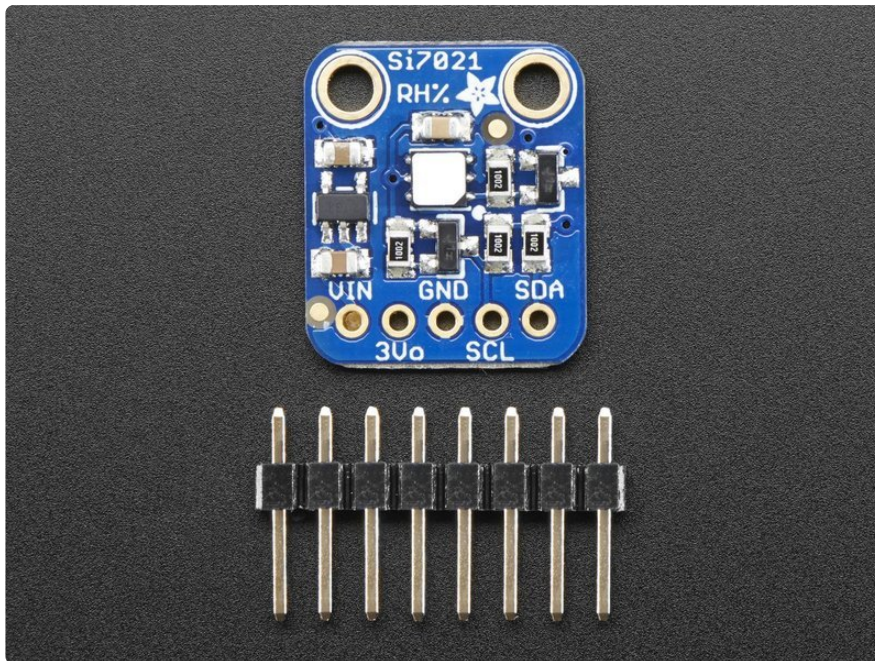
Power Pins:

- **Vin** - this is the power pin. Since the chip uses 3 VDC, we have included a voltage regulator on board that will take 3-5VDC and safely convert it down. 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
- **3v3** - this is the 3.3V output from the voltage regulator, you can grab up to 100mA from this if you like
- **GND** - common ground for power and logic

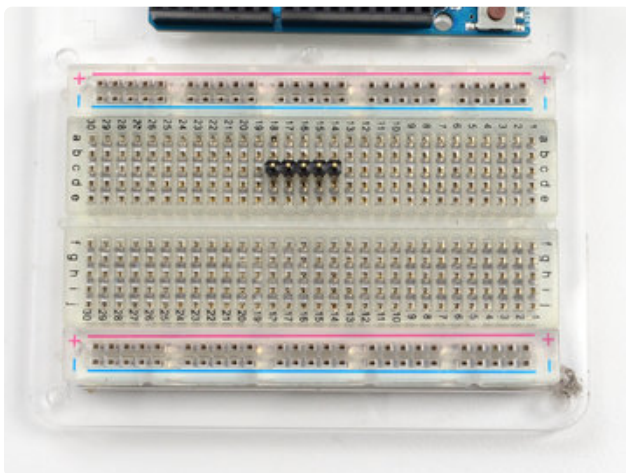
I2C Logic pins:

- **SCL** - I2C clock pin, connect to your microcontrollers I2C clock line.
- **SDA** - I2C data pin, connect to your microcontrollers I2C data line.
- **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) (<https://adafru.it/Ft6>).

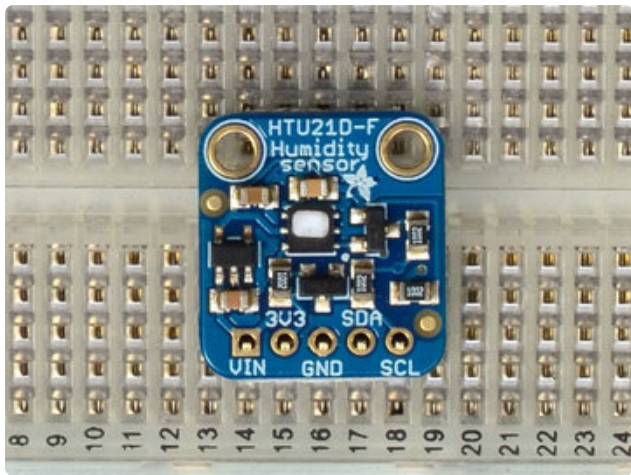
Assembly



The photos below show a an HTUD21 sensor rather than the Si7021 but the soldering procedure is identical!

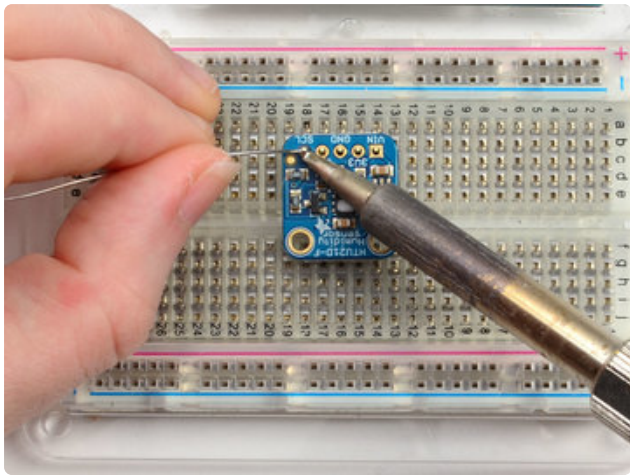


Prepare the header strip:
Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - **long pins down**



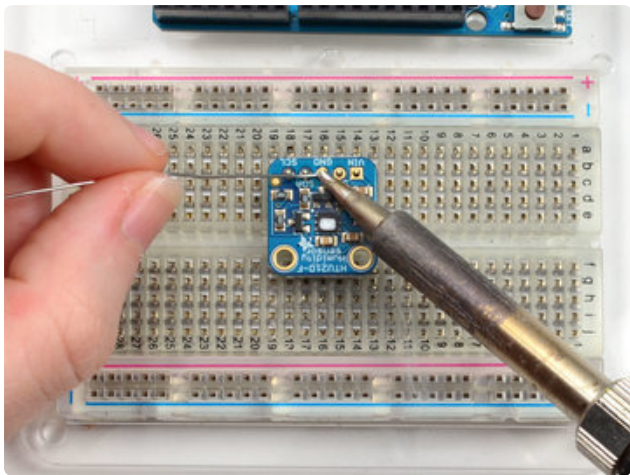
Add the breakout board:

Place the breakout board over the pins so that the short pins poke through the breakout pads

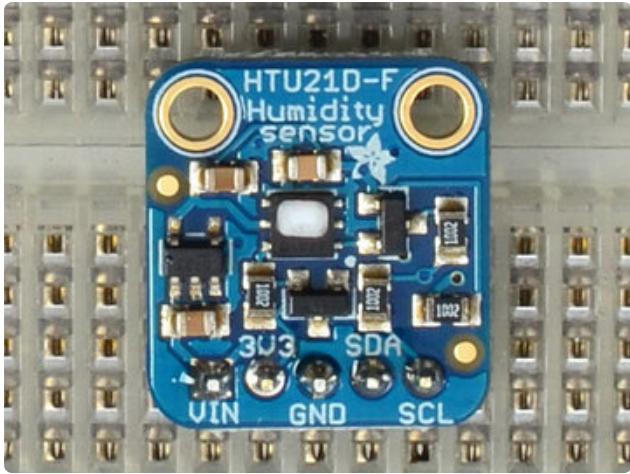


And Solder!

Be sure to solder all pins for reliable electrical contact.



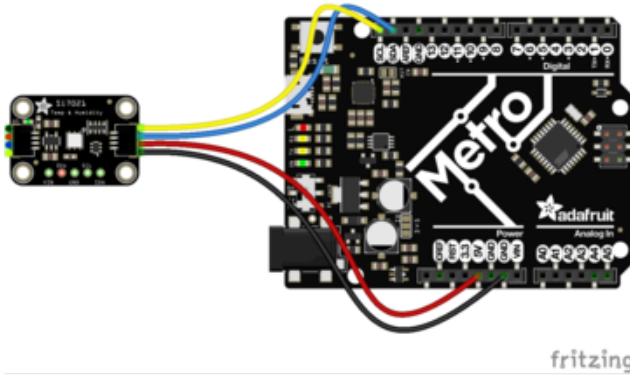
(For tips on soldering, be sure to check out our [Guide to Excellent Soldering \(https://adafruit.it/aTk\)](https://adafruit.it/aTk)).



You're done! Check your solder joints visually and continue onto the next steps

Arduino Code

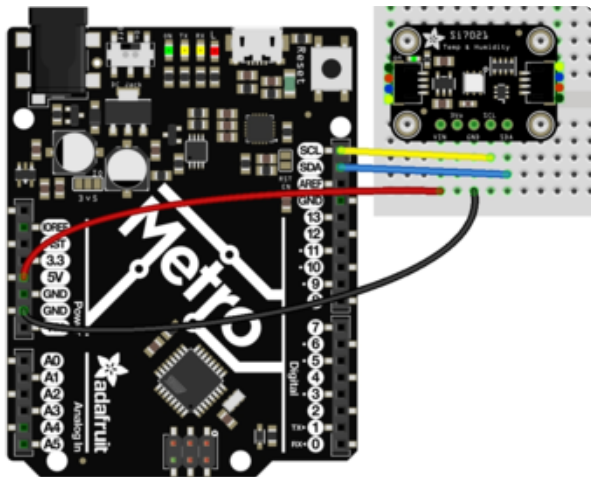
You can easily wire this breakout to any microcontroller, we'll be using an Arduino. For another kind of microcontroller, just make sure it has I2C, then port the code - its pretty simple stuff!



Connect **Vin** to the power supply, 3-5V is fine. (red wire on **STEMMA QT** version)

Use the same voltage that the microcontroller logic is based off of. For most Arduinos, that is 5V

Connect **GND** to common power/data ground (black wire on **STEMMA QT** version)



Connect the **SCL** pin to the I2C clock **SCL** pin on your Arduino. (yellow wire on **STEMMA QT** version) On an UNO & '328 based Arduino, this is also known as **A5**,

on a Mega it is also known as **digital 21** and on a Leonardo/Micro, **digital 3**

Connect the **SDA** pin to the I2C data **SDA** pin on your Arduino. (blue wire on **STEMMA QT** version) On an UNO & '328 based Arduino, this is also known as **A4**, on a Mega it is also known as **digital 20** and on a Leonardo/Micro, **digital 2**

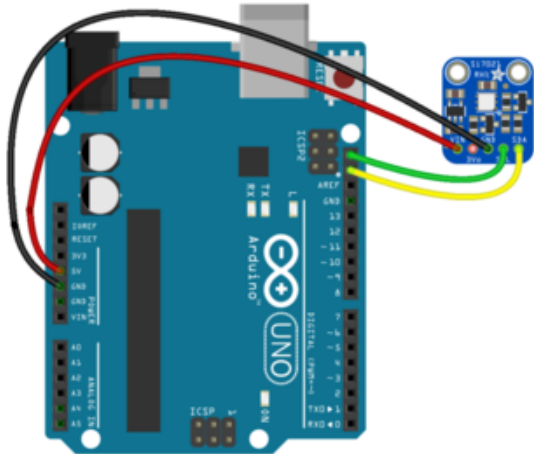


diagram.fzz Fritzing file

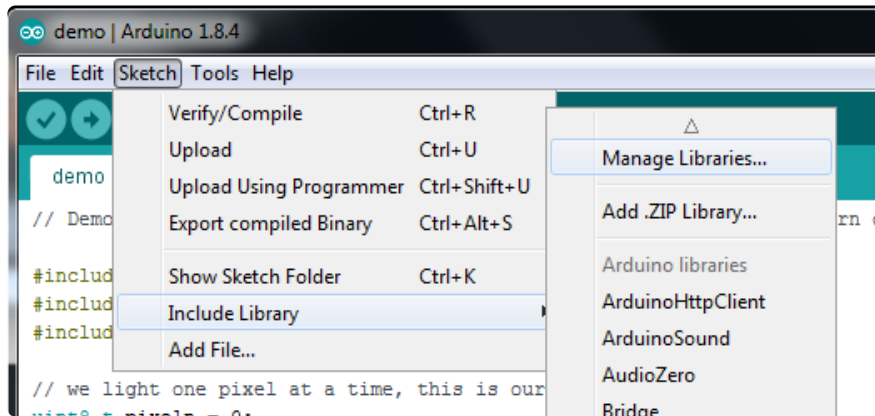
<https://adafru.it/rAA>

The Si7021 has a default I2C address of **0x40** and cannot be changed!

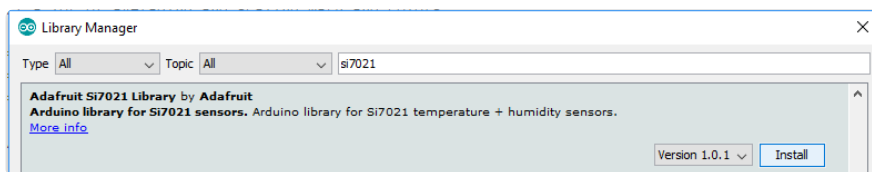
Install Adafruit_Si7021 library

To begin reading sensor data, you will need to [install the Adafruit_Si7021 library \(code on our github repository\)](https://adafru.it/rAw) (<https://adafru.it/rAw>). It is available from the Arduino library manager so we recommend using that.

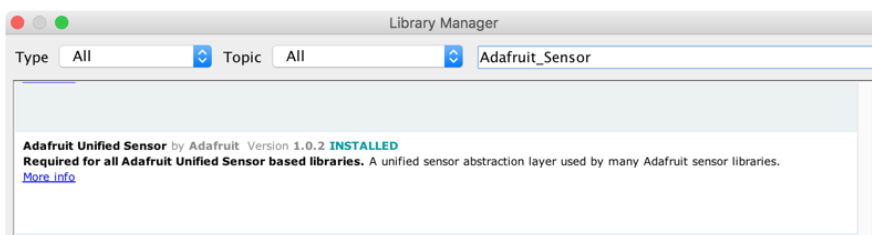
From the IDE open up the library manager...



And type in **adafruit si7021** to locate the library. Click **Install**



Also install the **Adafruit Unified Sensor** library in the same way:

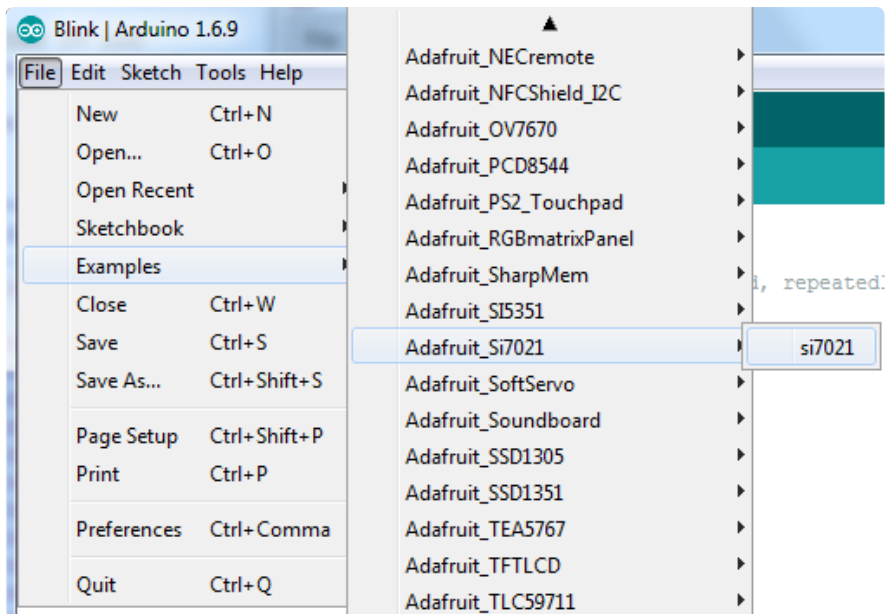


We also have a great tutorial on Arduino library installation at:

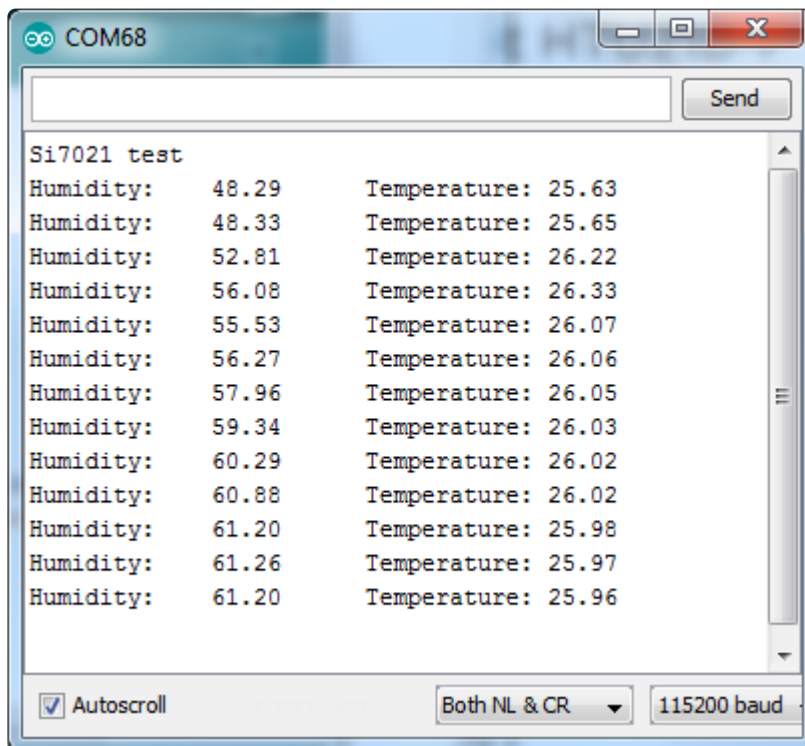
<http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use> (<https://adafru.it/aYM>)

Load Demo

Open up **File->Examples->Adafruit_Si7021->si7021** and upload to your Arduino wired up to the sensor



That's it! Now open up the serial terminal window at 115200 speed to begin the test.



You can try breathing on the sensor to increase the humidity. The sensor reacts very fast!

Library Reference

The library we have is simple and easy to use

You can create the **Adafruit_Si7021** object with:

```
Adafruit_Si7021 sensor = Adafruit_Si7021();
```

There are no pins to set since you must use the I2C bus!

Then initialize the sensor with:

```
sensor.begin();
```

this function returns **True** if the sensor was found and responded correctly and **False** if it was not found

Once initialized, you can query the temperature in °C with

```
sensor.readTemperature()
```

Which will return floating point (decimal + fractional) temperature. You can convert to Fahrenheit by multiplying by 1.8 and adding 32 as you have learned in grade school!

Reading the humidity is equally simple. Call

```
sensor.readHumidity()
```

to read the humidity also as a floating point value between 0 and 100 (this reads % humidity)

This sensor also has a serial number which you might find handy! Call `sensor.readSerialNumber()` to read out the 8 bytes of unique ID. Then you can access them from `sensor.sernum_a` and `sensor.sernum_b`

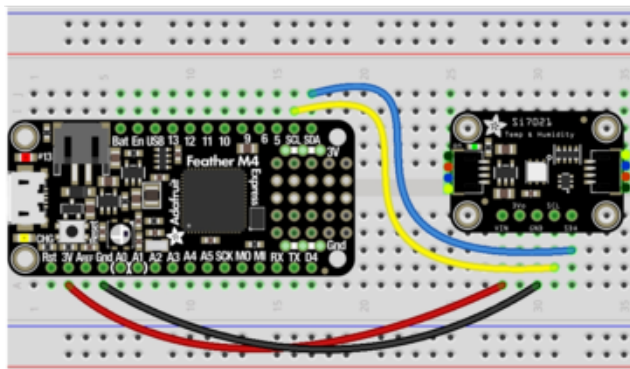
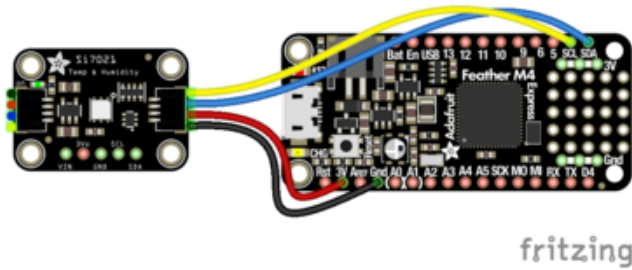
Python & CircuitPython

It's easy to use the Si7021 sensor with Python or CircuitPython and the [Adafruit CircuitPython SI7021 \(https://adafru.it/BfV\)](https://adafru.it/BfV) module. This module allows you to easily write Python code that reads the humidity and temperature, pressure 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 SI7021 to your board exactly as shown on the previous pages for Arduino. Here's an example of wiring a Feather M0 or M4 to the sensor with I2C:

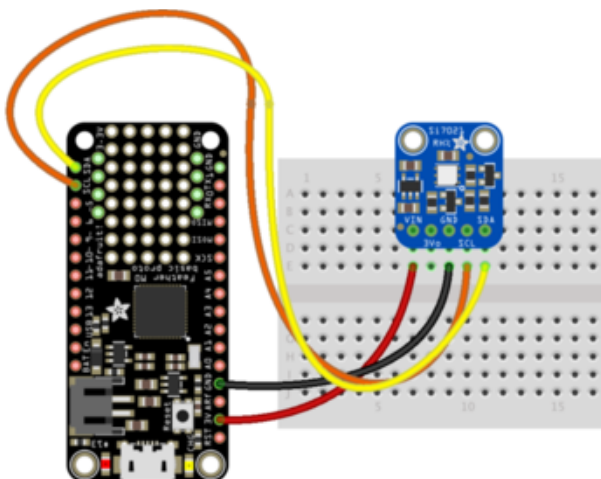


Board 3V to sensor VIN (red wire on STEMMA QT version)

Board GND to sensor GND (black wire on STEMMA QT version)

Board SCL to sensor SCL (yellow wire on STEMMA QT version)

Board SDA to sensor SDA (blue wire on STEMMA QT version)



Python Computer Wiring

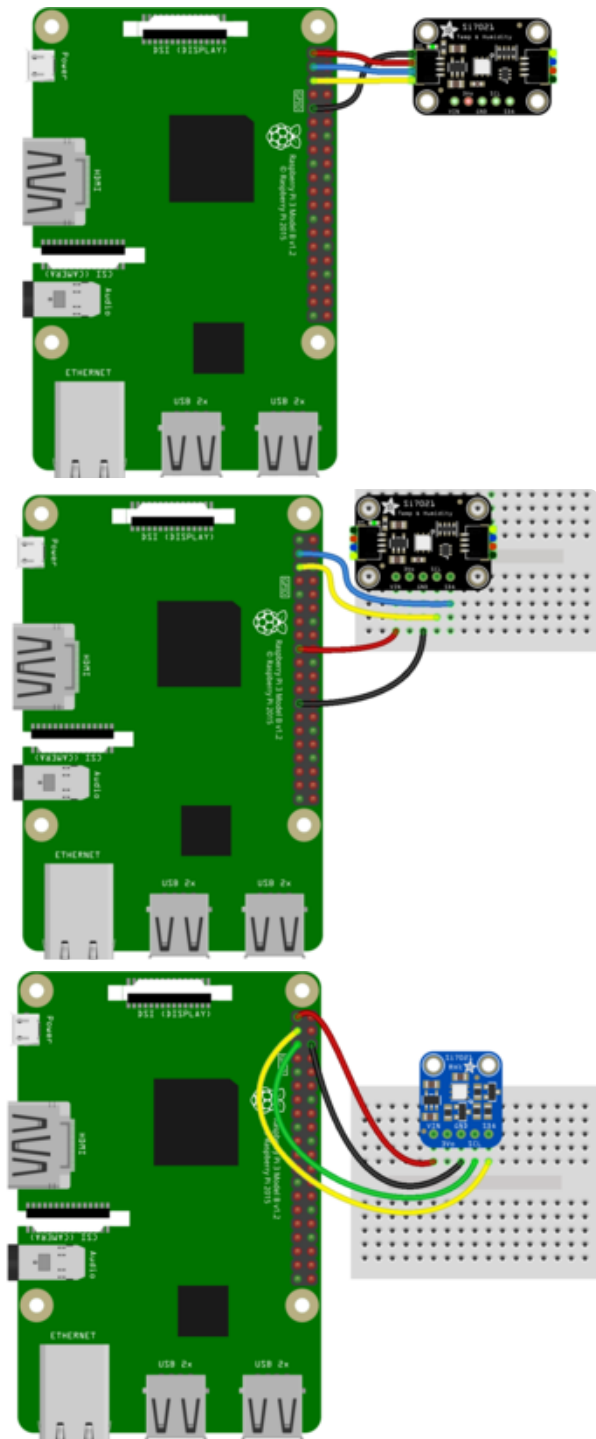
The Si7021 can potentially clock stretch the I2C signal, which can cause issues on Raspberry Pi's. See here for suggested fix:

I2C Clock Stretching on Raspberry Pi

<https://adafru.it/C7x>

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) (<https://adafru.it/BSN>).

Here's the Raspberry Pi wired with I2C:



- Pi 3V3 to sensor VIN (red wire on STEMMA QT version)
- Pi GND to sensor GND (black wire on STEMMA QT version)
- Pi SCL to sensor SCL (yellow wire on STEMMA QT version)
- Pi SDA to sensor SDA (blue wire on STEMMA QT version)

CircuitPython Installation of Si7021 Library

You'll need to install the [Adafruit CircuitPython Si7021 \(https://adafru.it/BfV\)](https://adafru.it/BfV) library on your CircuitPython board.

First make sure you are running the [latest version of Adafruit CircuitPython \(https://adafru.it/tBa\)](https://adafru.it/tBa) 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 \(https://adafru.it/zdx\)](https://adafru.it/zdx). For example the Circuit Playground Express guide has [a great page on how to install the library bundle \(https://adafru.it/Bf2\)](https://adafru.it/Bf2) for both express and non-express boards.

Load the following libraries into your **lib** folder on your **CIRCUITPY** drive:

- `adafruit_si7021/`
- `adafruit_bus_device/`

Before continuing make sure your board's lib folder or root filesystem has the `adafruit_si7021/`, and `adafruit_bus_device/` folders copied over.

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

Python Installation of Si7021 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 \(https://adafru.it/BSN\)](https://adafru.it/BSN)!

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

- `sudo pip3 install adafruit-circuitpython-si7021`

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. First run the following code to import the necessary modules and initialize the I2C bus:

```
import board
import adafruit_si7021
sensor = adafruit_si7021.SI7021(board.I2C())
```

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

- **temperature** - The sensor temperature in degrees Celsius.
- **relative_humidity** - The percent humidity as a value from 0 to 100%.

```
print('Temperature: {} degrees C'.format(sensor.temperature))
print('Humidity: {}'.format(sensor.relative_humidity))
```

```
>>> print('Temperature: {} degrees C'.format(sensor.temperature))
Temperature: 21.6142 degrees C
>>> print('Humidity: {}'.format(sensor.relative_humidity))
Humidity: 48.0605%
>>> █
```

That's all there is to using the Si7021 sensor with CircuitPython!

Full Example Code

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

"""
Initializes the sensor, gets and prints readings every two seconds.
"""
import time
import board
import adafruit_si7021

# Create library object using our Bus I2C port
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_si7021.SI7021(i2c)

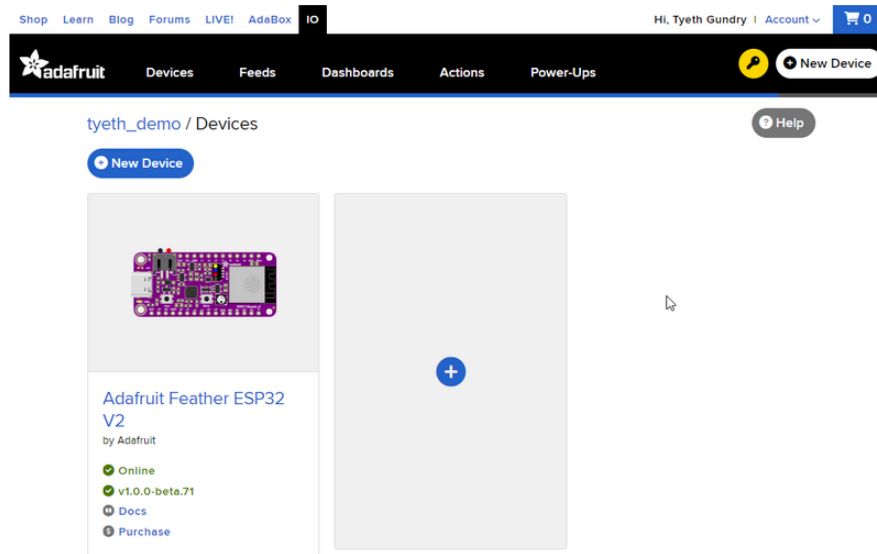
# If you'd like to use the heater, you can uncomment the code below
# and pick a heater level that works for your purposes
#
# sensor.heater_enable = True
# sensor.heater_level = 0 # Use any level from 0 to 15 inclusive

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

Python Library Docs

[Python Library Docs \(https://adafru.it/C3f\)](https://adafru.it/C3f)

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 \(https://adafru.it/fsU\)](https://adafru.it/fsU), a web platform designed ([by Adafruit! \(https://adafru.it/Bo5\)](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.

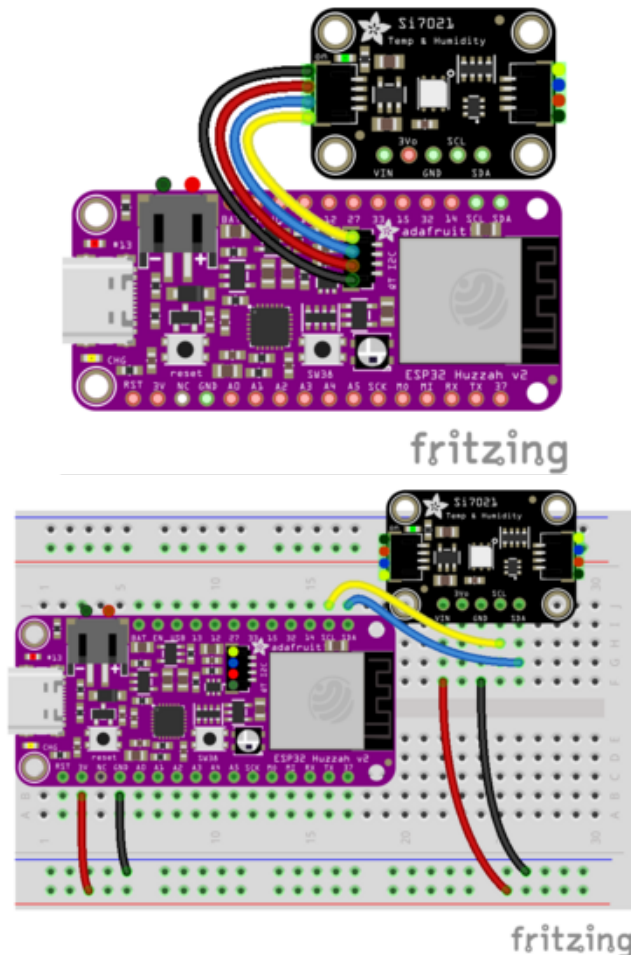
If you've never used WipperSnapper, click below to read through the quick start guide before continuing.

**Quickstart: Adafruit IO
WipperSnapper**

<https://adafru.it/Vfd>

Wiring

First, wire up an Si7021 to your board exactly as follows. Here is an example of the Si7021 wired to an [Adafruit ESP32 Feather V2](http://adafru.it/5400) (<http://adafru.it/5400>) using I2C [with a STEMMA QT cable](http://adafru.it/4210) (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)

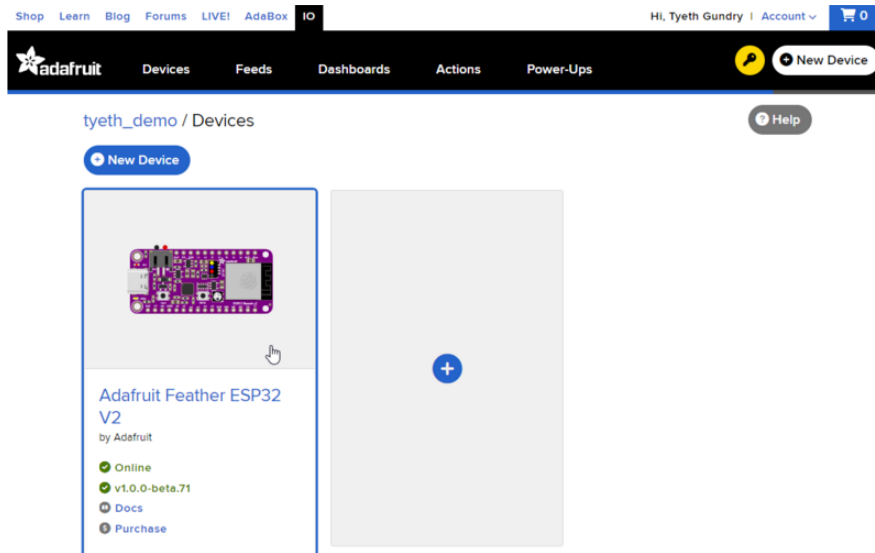
Board SCL to sensor SCL (yellow wire on STEMMA QT)

Board SDA to sensor SDA (blue wire on STEMMA QT)

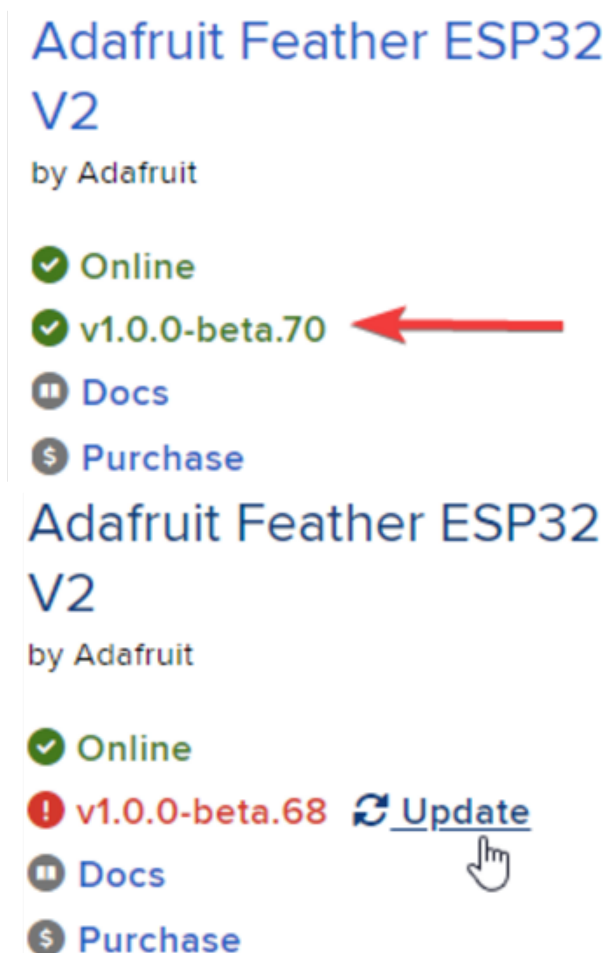
Usage

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

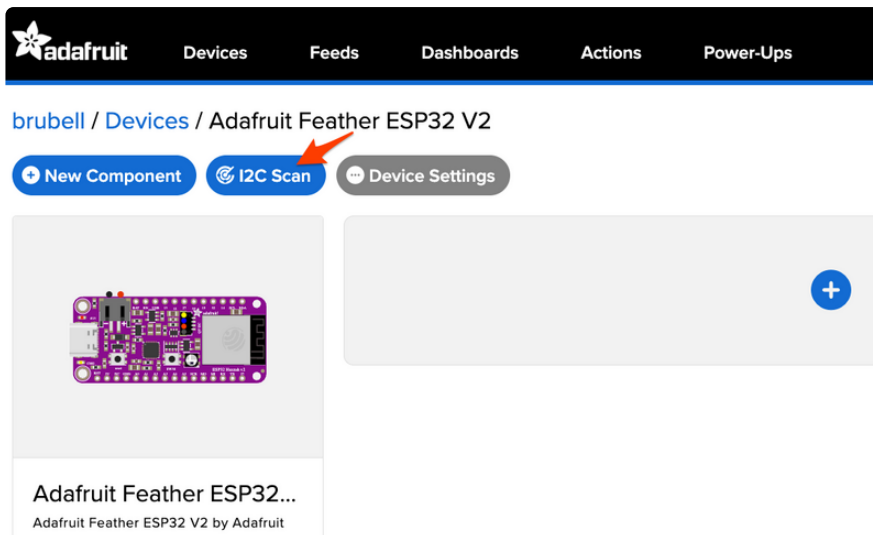


On the device page, quickly check that you're running the latest version of the WipperSnapper firmware.

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 exclamation mark "!" - [update to the latest WipperSnapper firmware](https://adafru.it/Vfd) before continuing.

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



You should see the Si7021's default I2C address of `0x40` pop-up in the I2C scan list.

I2C Scan Complete ✕

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

Close Scan Again

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

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

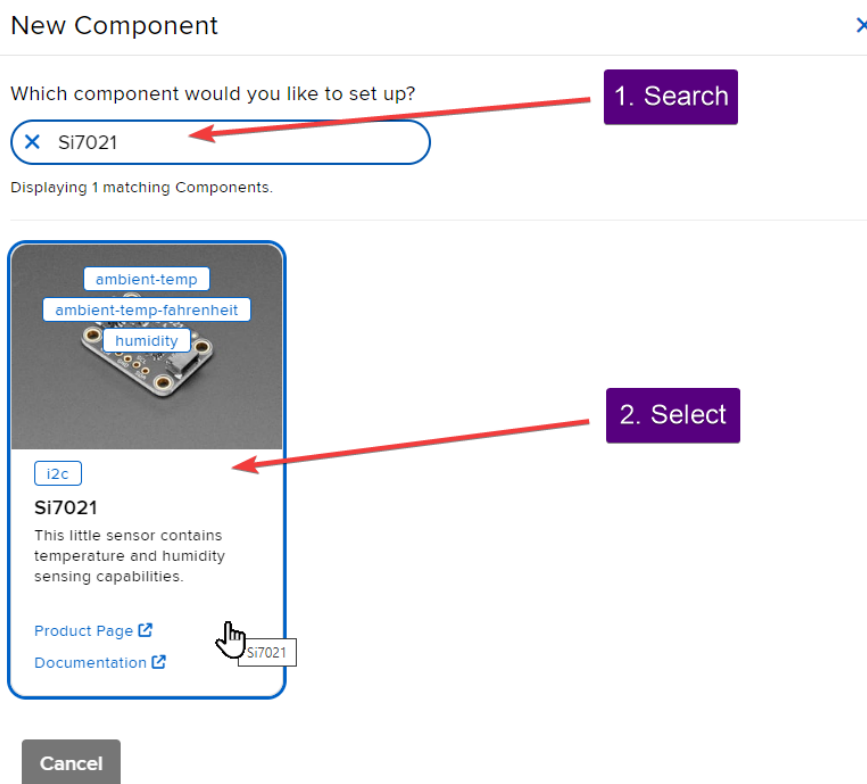
Then, reset the board and let it re-connect to Adafruit IO WipperSnapper.

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 **Si7021** into the search bar, then select the **Si7021** component.



On the component configuration page, the Si7021'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 Si7021 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 Si7021 Component



Select I2C Address:

0x40

Enable Si7021: Temperature Sensor (°C)?

Name:

Si7021: Temperature Sensor (°C)

Send Data:

Every 30 seconds

Enable Si7021: Temperature Sensor (°F)?

Name:

Si7021: Temperature Sensor (°F)

Send Data:

Every 30 seconds

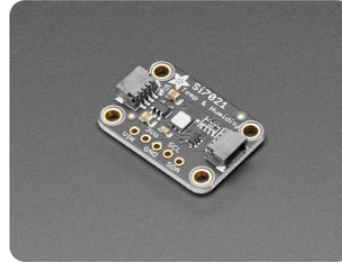
Enable Si7021: Humidity Sensor?

Name:

Si7021: Humidity Sensor

Send Data:

Every 30 seconds



[← Back to Component Type](#)

[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.

The screenshot shows the Adafruit IO dashboard for a device named 'tyeth_demo'. The dashboard displays three sensor components:

- Si7021: Humidity Sensor** (si7021:humidity) showing 52.84%.
- Si7021: Temperature Sensor (°C)** (si7021:ambient-temp) showing 19.99°C.
- Si7021: Temperature Sensor (°F)** (si7021:ambient-temp-fahrenheit) showing 68.00°F.

Each component has a 'Create Action | Add to Dashboard' link. The dashboard also includes a sidebar for the device 'Adafruit Feather ESP32 V2' with status 'Online', version 'v1.0.0-beta.76', and links for 'Docs' and 'Purchase'. A 'Report Bugs' link is also present.

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

New Component Auto-Config I2C Scan Settings Help

Adafruit Feather ESP32 V2 by Adafruit

- Online
- v1.0.0-beta.76
- Docs
- Purchase

Report Bugs

Si7021: Humidity Sensor (si7021:humidity) 52.59%

Create Action | Add to Dashboard

Si7021: Temperature Sensor (°C) (si7021:ambient-temp) 19.66°C

Create Action | Add to Dashboard

Si7021: Temperature Sensor (°F) (si7021:ambient-temp-fahrenheit) 67.38°F

Create Action | Add to Dashboard

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 \(https://adafru.it/10aZ\)](https://adafru.it/10aZ).

tyeth_demo / Feeds / Si7021: Humidity Sensor

Feed Info: Manage feed name, key, description, and tags.

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

Sharing: Not shared yet.

Feed History: Feed history is **ON**. Value size is limited to **1KB**. You have 14 data points from February 29th 2024, 8:02PM to February 29th 2024, 8:09PM.

Notifications: This feed is **Online**. You have no notifications active for this feed.

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

Disable Feed: Disabling a feed will remove it from your feed count and prevent you from adding new data to it.

License: No Default License.

Created at Value Location

2024/02/29 08:09:42PM	54.214996337890625	
2024/02/29 08:09:12PM	54.810089111328125	
2024/02/29 08:08:42PM	55.092376708984375	
2024/02/29 08:08:12PM	55.824798583984375	
2024/02/29 08:07:42PM	55.725616455078125	
2024/02/29 08:07:12PM	56.297821044921875	
2024/02/29 08:06:42PM	56.633514404296875	
2024/02/29 08:06:11PM	55.145782470703125	
2024/02/29 08:05:41PM	55.229705810546875	

Downloads

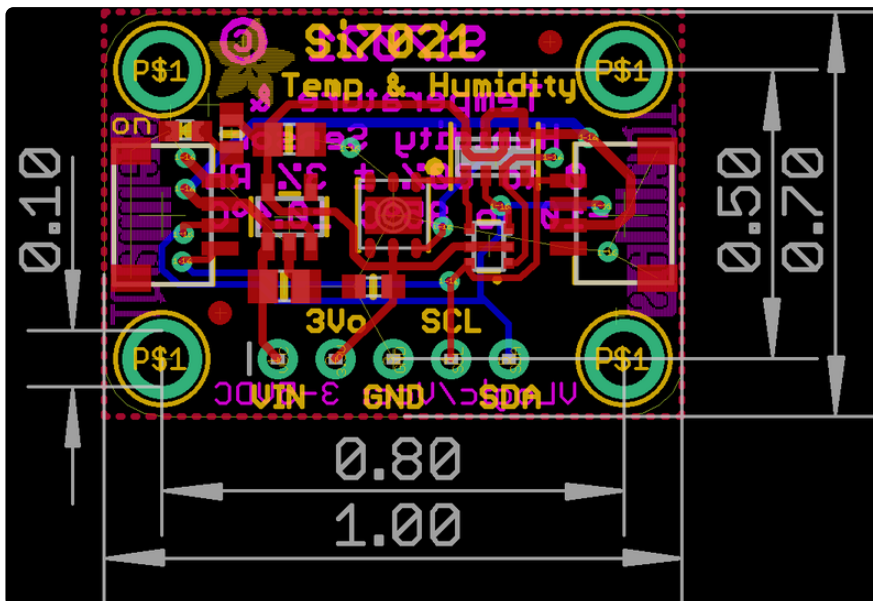
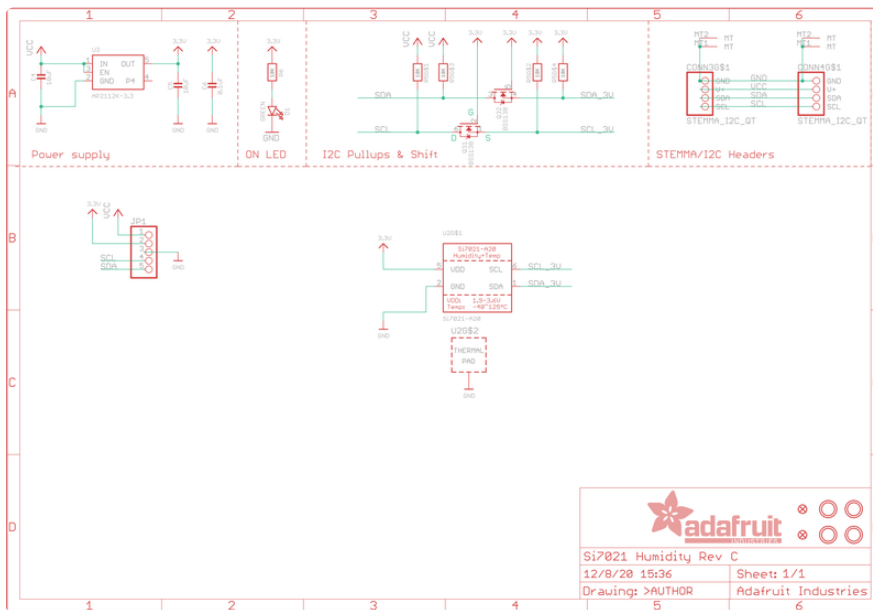
Files & Datasheets

- [Fritzing object in Adafruit Fritzing library \(https://adafru.it/aP3\)](https://adafru.it/aP3)
- [Arduino library on GitHub \(https://adafru.it/rAw\)](https://adafru.it/rAw)
- [EagleCAD PCB files on GitHub \(https://adafru.it/rAB\)](https://adafru.it/rAB)
- [Si7021-A20 datasheet \(https://adafru.it/rAC\)](https://adafru.it/rAC)

- [K&R Smith calibration notes \(https://adafru.it/BfU\)](https://adafru.it/BfU)

If you try writing your own I2C driver, please be aware it is possible to brick a Si7021 by having it erase its calibration memory! https://www.silabs.com/community/sensors/forum.topic.html/si7021_loses_serial-Lkb7

Schematic and Fabrication Print STEMMA QT Version



Schematic and Fabrication Print Original Version

