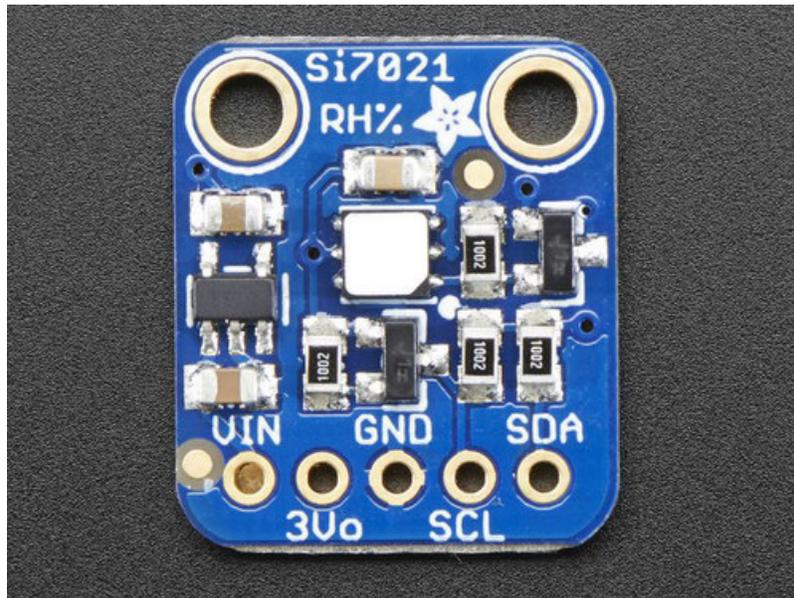


## Adafruit Si7021 Temperature + Humidity Sensor

Created by lady ada

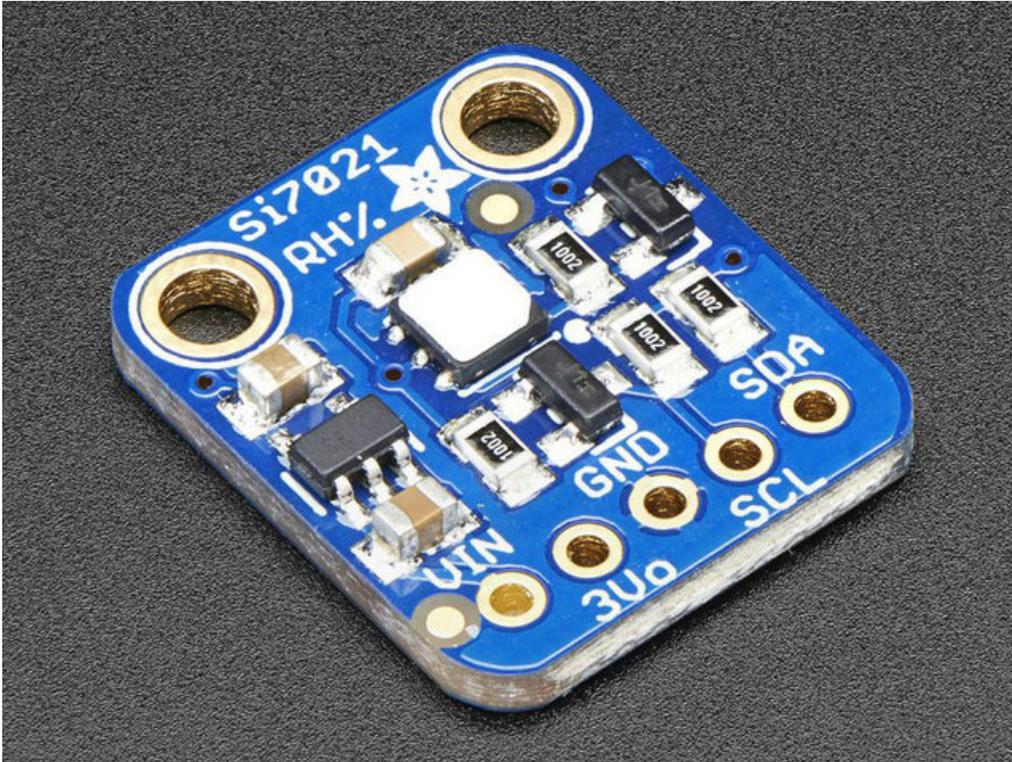


Last updated on 2017-11-12 06:14:07 PM UTC

## Guide Contents

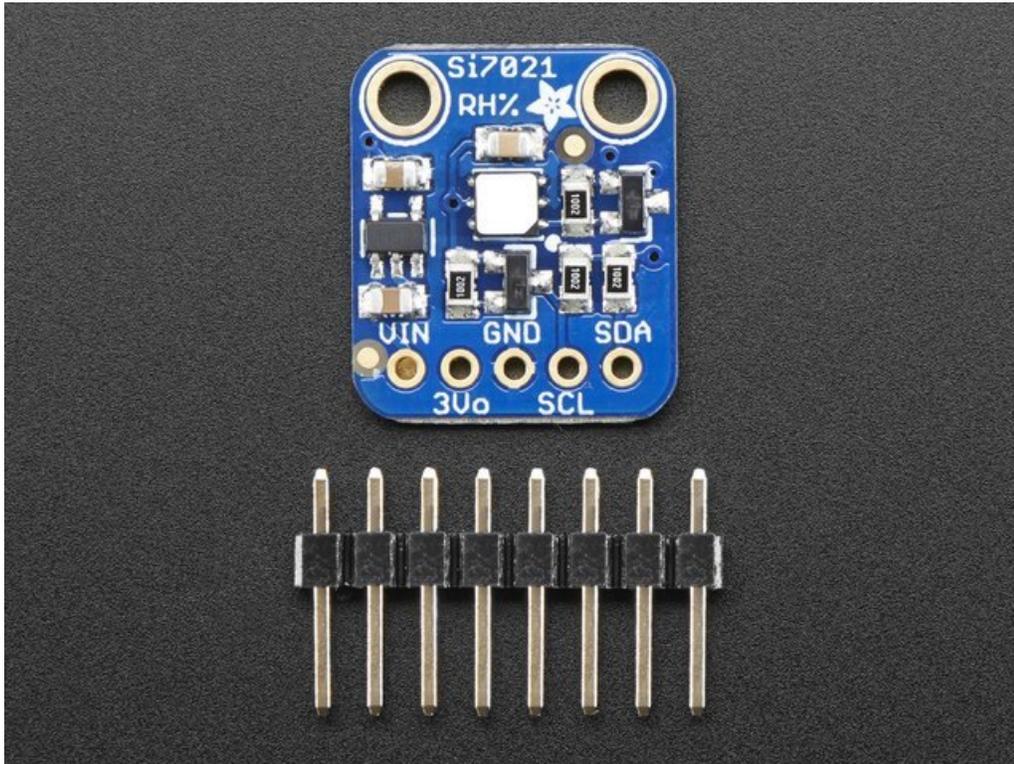
Guide Contents	2
Overview	3
Pinouts	5
Power Pins:	5
I2C Logic pins:	5
Assembly	6
Prepare the header strip:	6
Add the breakout board:	7
And Solder!	8
Arduino Code	10
Download Adafruit_Si7021	10
Load Demo	11
Library Reference	12
CircuitPython Code	14
Usage	15
Downloads	16
Files & Datasheets	16
Schematic	16
Fabrication Print	16

## Overview

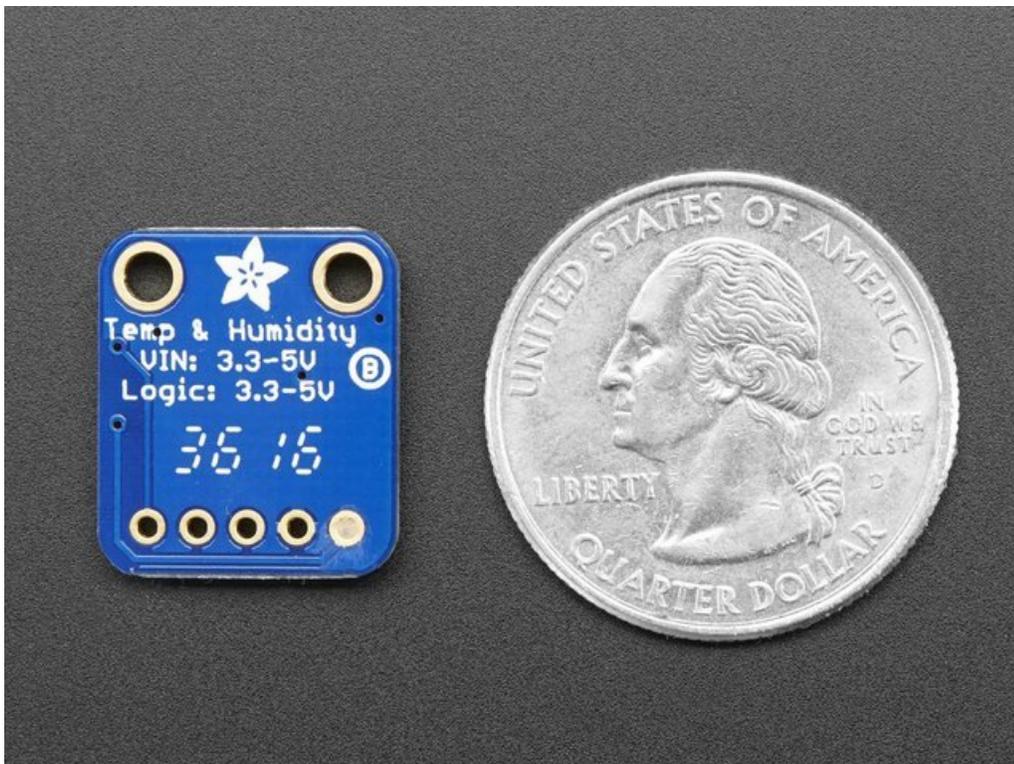


It's summer and you're sweating and your hair's all frizzy and all you really want to know is why the weatherman said this morning that today's relative humidity would max out at a perfectly reasonable 42% when it feels more like 77%. Enter the **Si7021 Temperature + Humidity Sensor** - the best way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

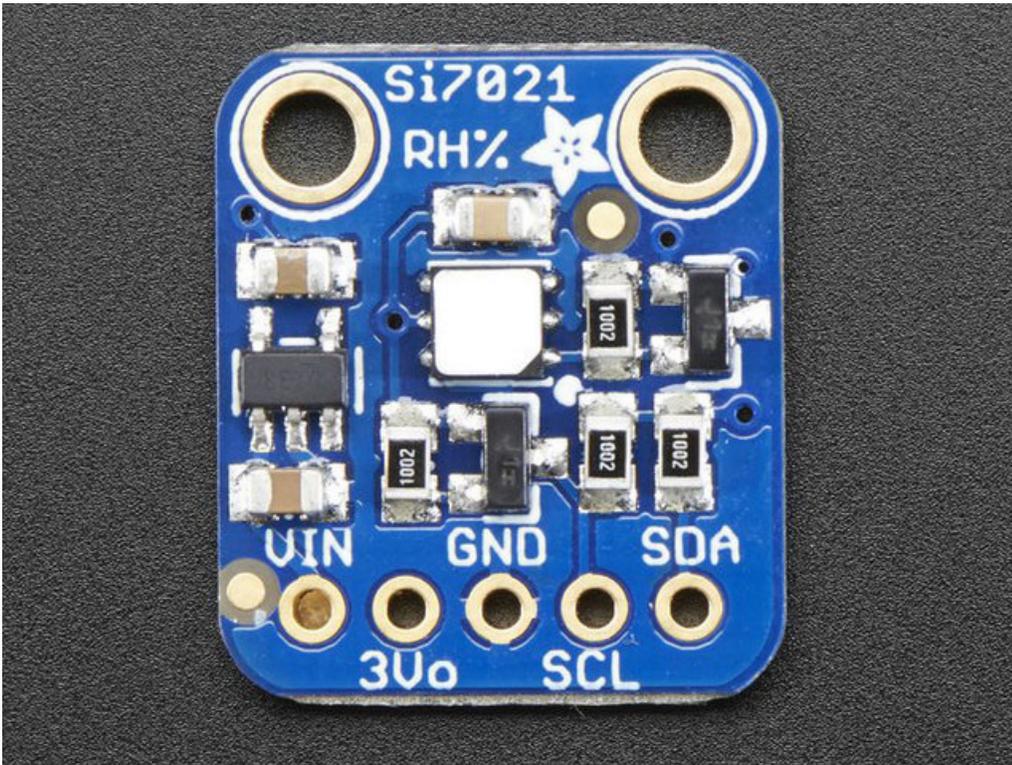


We put this nice sensor on a breakout board with a 3.3V regulator and level shifting so you can use it safely with 3.3V or 5V power & logic. There's a PTFE filter to keep the sensor clean, that's the white flat thing on top. Also comes with some pin header. Some light soldering is required to attach the header but it's easy to do.



## 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!



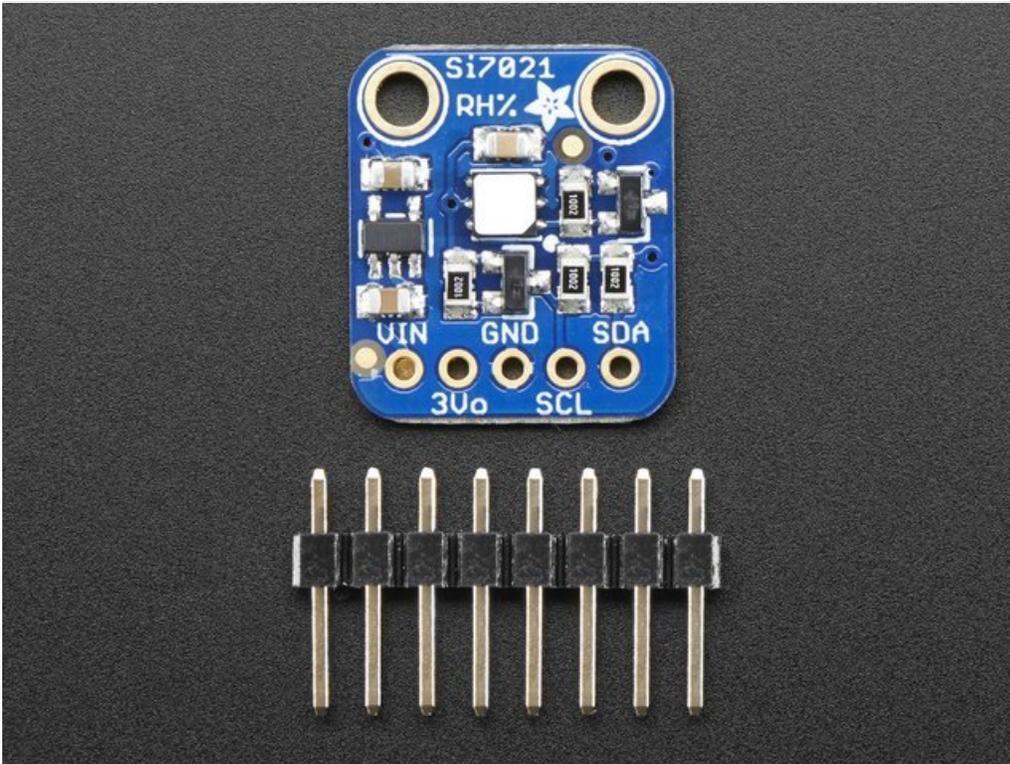
### Power Pins:

- **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

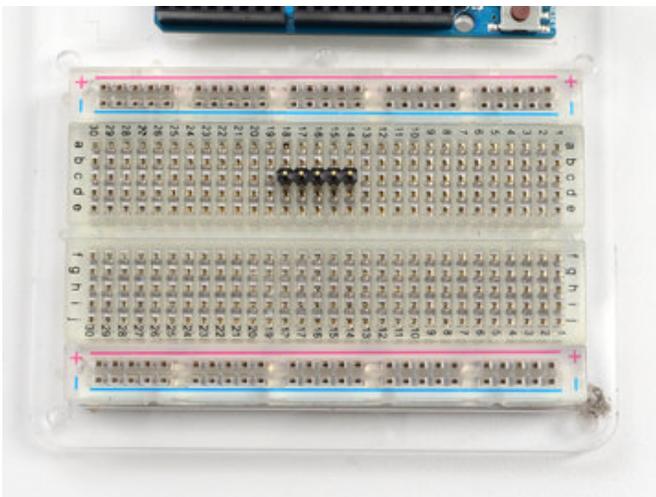
### I2C Logic pins:

- **SCL** - I2C clock pin, connect to your microcontrollers I2C clock line.
- **SDA** - I2C data pin, connect to your microcontrollers I2C data line.

## Assembly

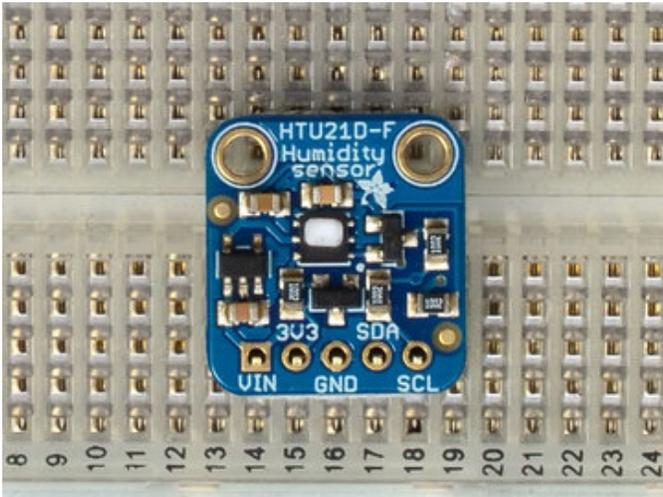


*The photos below show a an HTUD21 sensor rather than the Si7021 but the soldering procedure is identical!*



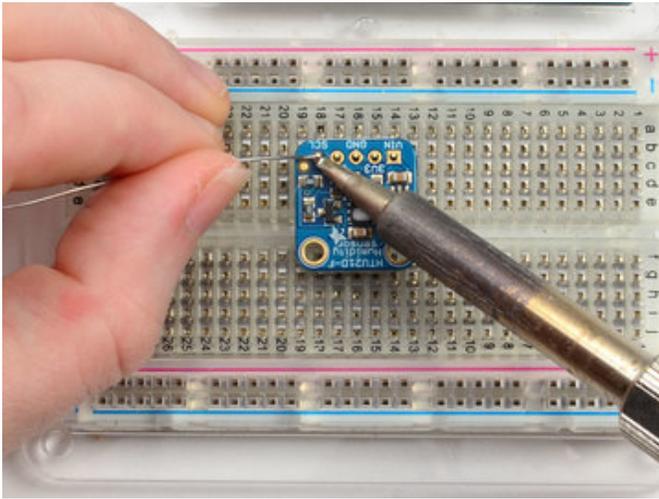
**Prepare the header strip:**

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:

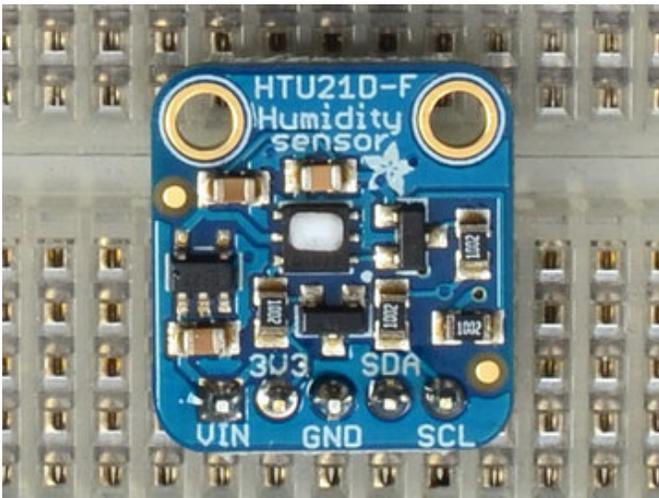
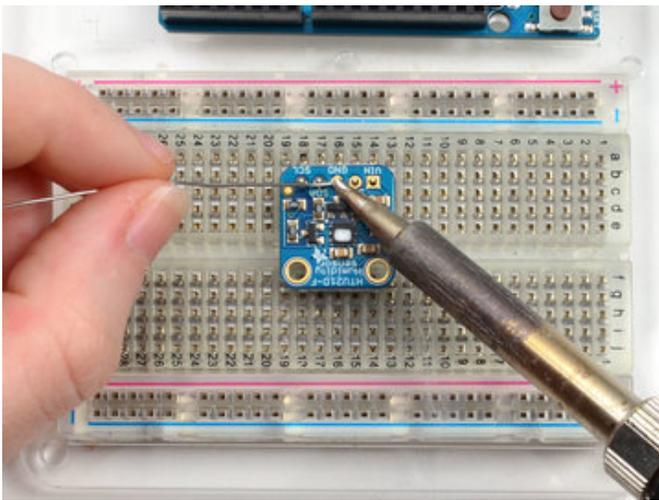
Place the breakout board over the pins so that the short pins poke through the breakout pads



## 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://adafru.it/aTk) (<https://adafru.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!

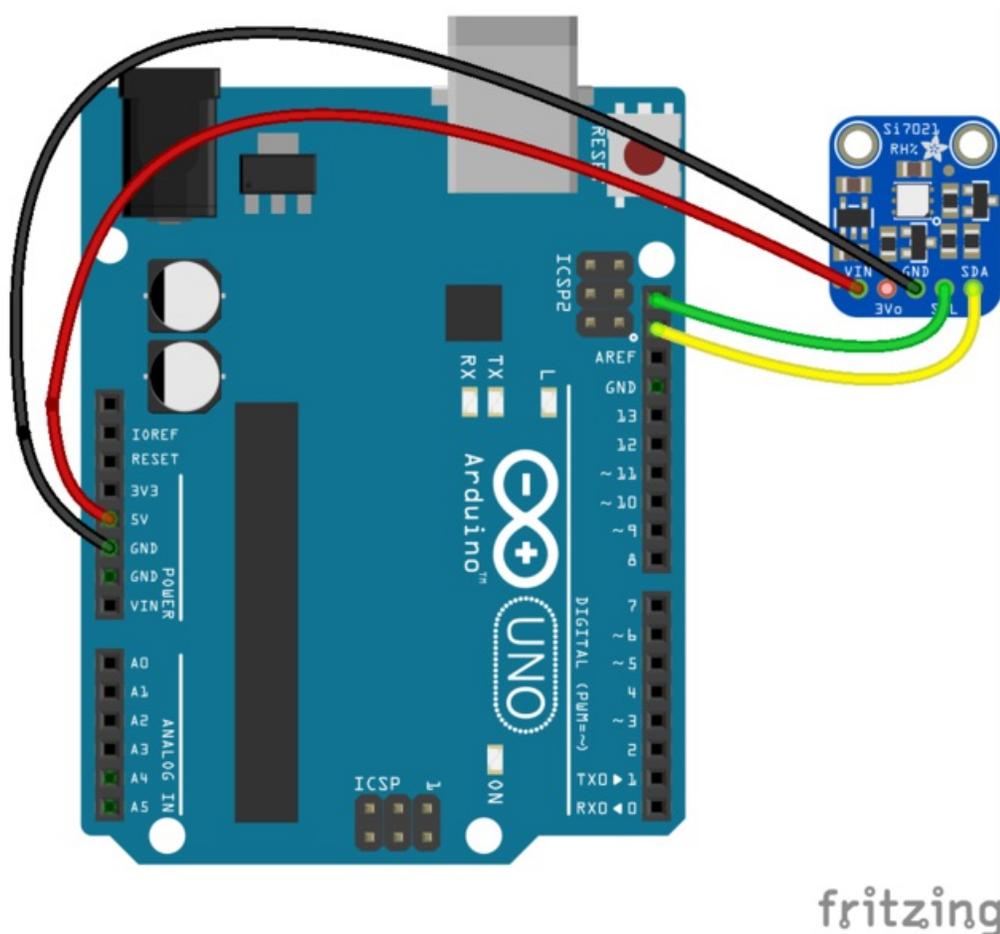


diagram.fzz Fritzing file

<https://adafruit.it/rAA>

- Connect **Vin** to the power supply, 3-5V is fine. 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
- Connect the **SCL** pin to the I2C clock **SCL** pin on your Arduino. 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. 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**

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

## Download Adafruit\_Si7021

To begin reading sensor data, you will need to [download Adafruit\\_Si7021 Library from our github repository](#). You can do that by visiting the github repo and manually downloading or, easier, just click this button to download the zip

## Download Adafruit Si7021 Library

<https://adafru.it/rAx>

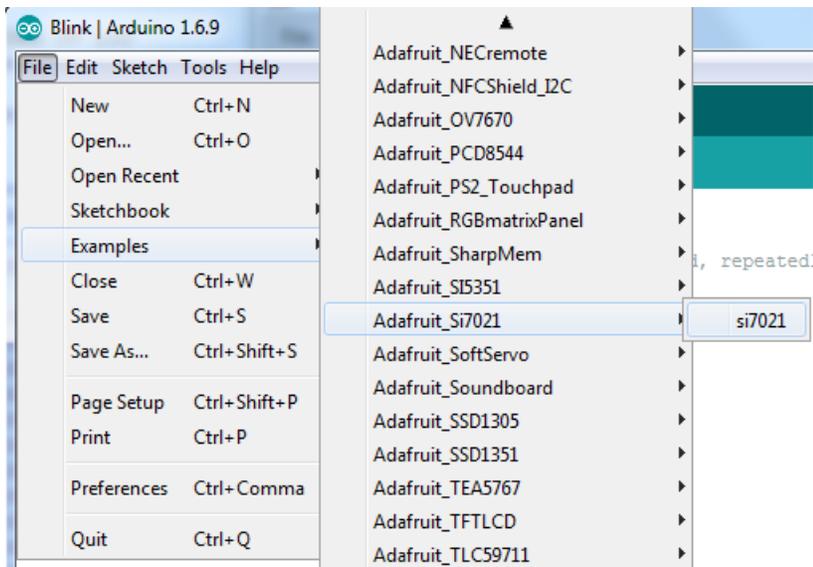
Rename the uncompressed folder **Adafruit\_Si7021** and check that the **Adafruit\_Si7021** folder contains **Adafruit\_Si7021.cpp** and **Adafruit\_Si7021.h**

Place the **Adafruit\_Si7021** library folder your **arduinodesketchfolder/libraries/** folder.  
You may need to create the **libraries** subfolder if its your first library. Restart the IDE.

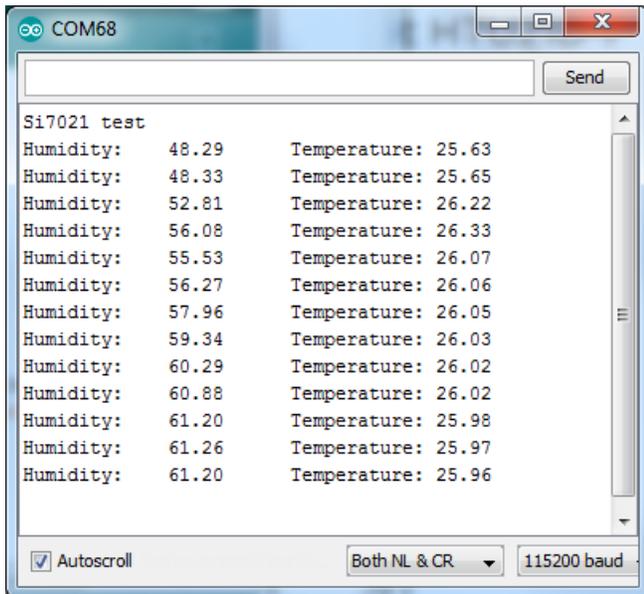
We also have a great tutorial on Arduino library installation at:  
<http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use>

## Load Demo

Open up **File->Examples->Adafruit\_Si7021->si7021** and upload to your Arduino wired up to the sensor



Thats 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()
```

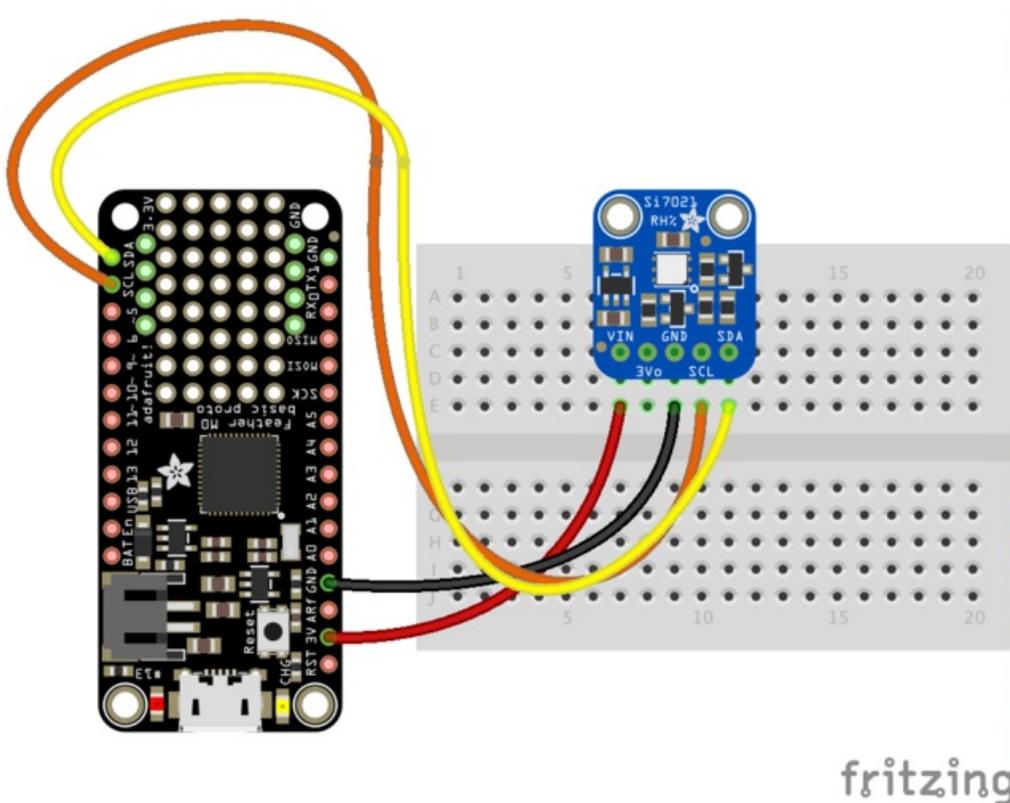
to read the humidity also as a floating point value between 0 and 100 (this reads % humidity)

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`

## CircuitPython Code

It's easy to use the SI7021 sensor with CircuitPython and the [Adafruit CircuitPython SI7021](#) module. This module allows you to easily write Python code that reads the humidity and temperature, pressure from the sensor.

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 to the sensor:



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

Next you'll need to install the [Adafruit CircuitPython SI7021](#) library on your CircuitPython board. **Remember this module is for Adafruit CircuitPython firmware and not MicroPython.org firmware!**

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

Remember for non-express boards like the Trinket M0, Gemma M0, and Feather/Metro M0 basic you'll need to manually install the necessary libraries from the bundle:

- `adafruit_si7021.mpy`
- `adafruit_bus_device`

You can also download the `adafruit_si7021.mpy` from [its releases page on Github](#).

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

Next [connect to the board's serial REPL](#) so you are at the CircuitPython `>>>` prompt.

## 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 busio
import adafruit_si7021
i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_si7021.SI7021(i2c)
```

Remember if you're using a board that doesn't support hardware I2C (like the ESP8266) you need to use the `bitbangio` module instead:

```
import board
import bitbangio
import adafruit_bmp280
i2c = bitbangio.I2C(board.SCL, board.SDA)
sensor = adafruit_si7021.SI7021(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!



