



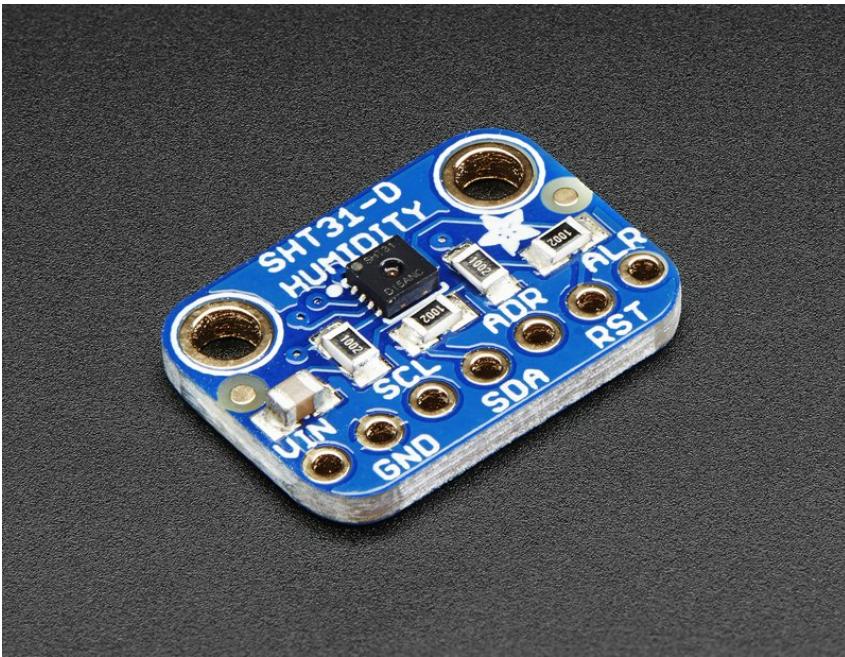
0

- Search Adafruit
- [SHOP](#)
- [BLOG](#)
- [LEARN](#)
- [FORUMS](#)
- [VIDEOS](#)
- [SIGN IN](#)
- [CLOSE MENU](#)

0 Items

[Sign In](#)

- [SHOP](#)
- [BLOG](#)
- [LEARN](#)
- [FORUMS](#)
- [VIDEOS](#)
- [ADABOX](#)



## [Adafruit SHT31-D Temperature & Humidity Sensor Breakout](#)

Ooh, a Sensiron sensor with true I2C!

- [Overview](#)
- [Pinouts](#)
- [Assembly](#)
- [Arduino Code](#)
- [CircuitPython Code](#)
- [Downloads](#)
- 
- [Featured Products](#)
- [Multiple Pages](#)
- [Download PDF](#)

### Contributors

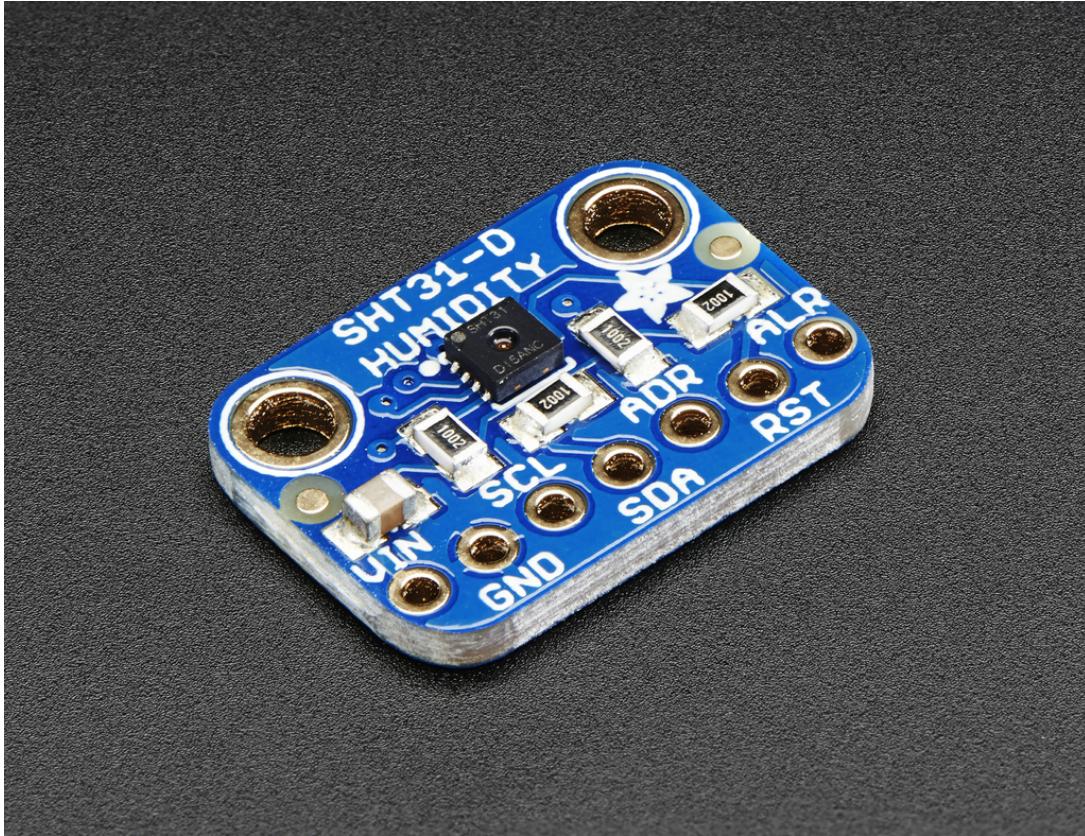
[lady ada](#)  
[Tony DiCola](#)

[Feedback? Corrections?](#)

[ADAFRUIT PRODUCTS](#) [SENSORS](#) / [TEMPERATURE & HUMIDITY BREAKOUT BOARDS](#)

### Overview

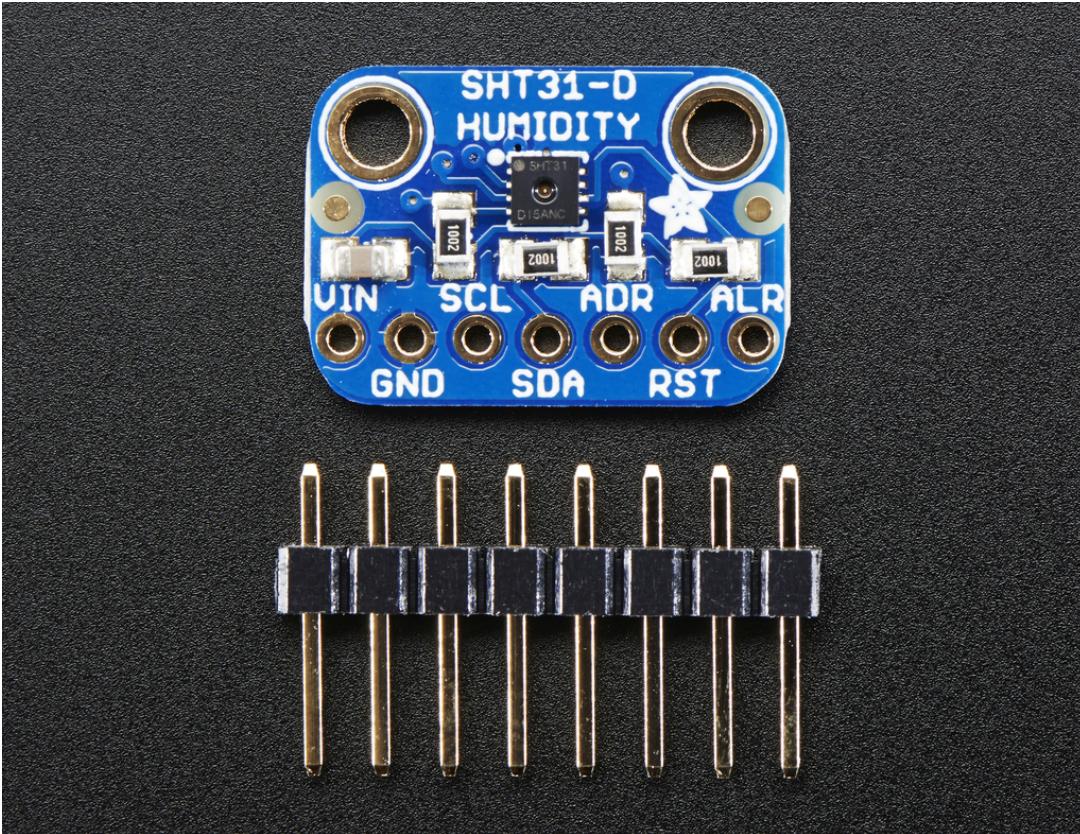
by [lady ada](#)



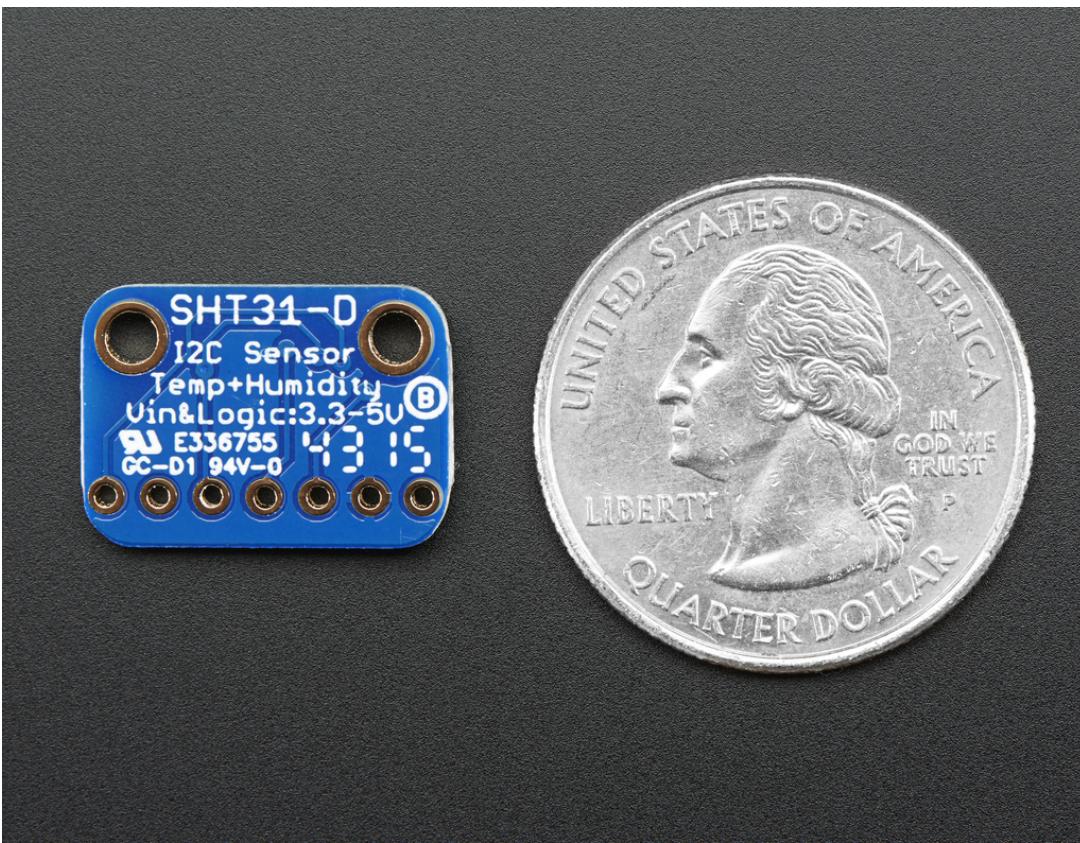
Sensirion Temperature/Humidity sensors are some of the finest & highest-accuracy devices you can get. And, finally we have some that have a true I2C interface for easy reading. The **SHT31-D** sensor has an excellent  $\pm 2\%$  relative humidity and  $\pm 0.3^\circ\text{C}$  accuracy for most uses.



Unlike earlier SHT sensors, this sensor has a *true* I2C interface, with two address options. It also is 3V or 5V compliant, so you can power and communicate with it using any microcontroller or microcomputer.



Such a lovely chip - so we spun up a breakout board with the SHT31-D and some supporting circuitry such as pullup resistors and capacitors. 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.



## Pinouts

by [lady ada](#)

The HTU21D-F 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 default I2C address is **0x44** and you can also select address **0x45** by connecting the **ADDR** pin to a high voltage signal.



### Power Pins:

- **Vin** - this is the power pin. The chip can use 2.5-5VDC for power. 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. For a 3.3V controller like a Raspberry Pi, connect to 3.3V
- **GND** - common ground for power and logic

### I2C Logic pins:

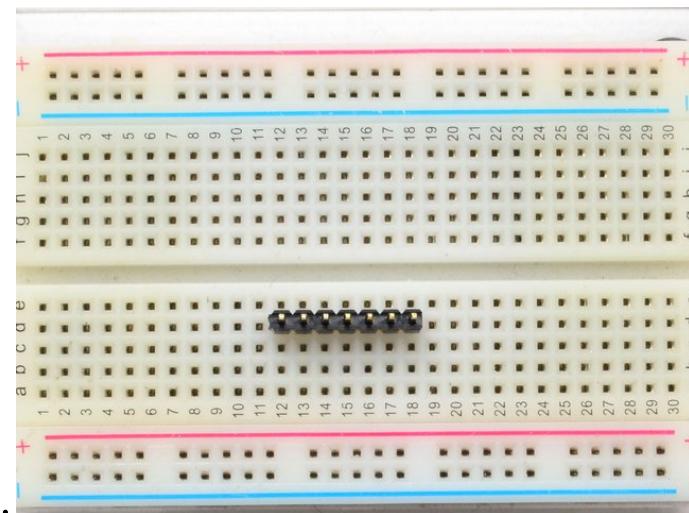
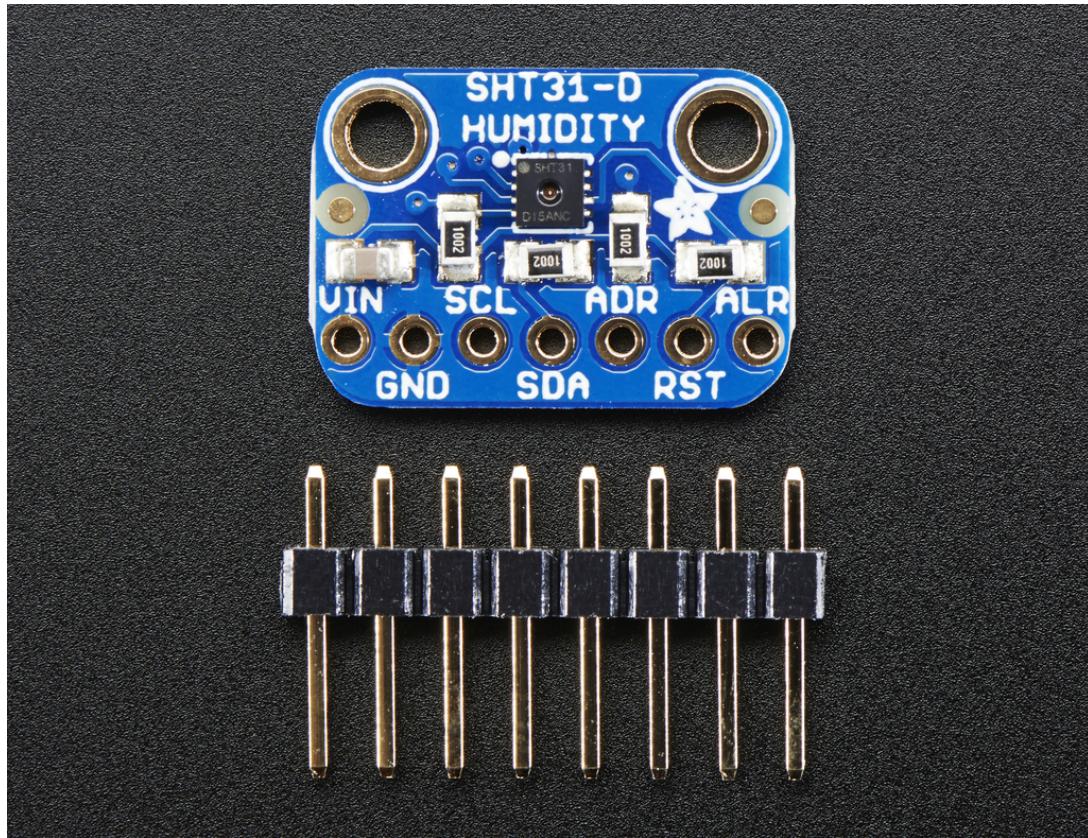
- **SCL** - I2C clock pin, connect to your microcontrollers I2C clock line. This pin has a 10K pullup resistor to Vin
- **SDA** - I2C data pin, connect to your microcontrollers I2C data line. This pin has a 10K pullup resistor to Vin

### Other Pins:

- **ADR** - This is the I2C address selection pin. This pin has a 10K pull down resistor to make the default I2C address **0x44**. You can tie this pin to Vin to make the address **0x45**
- **RST** - Hardware reset pin. Has a 10K pullup on it to make the chip active by default. Connect to ground to do a hardware reset!
- **ALR** - Alert/Interrupt output. You can set up the sensor to alert you when an event has occurred. Check the datasheet for how you can set up the alerts

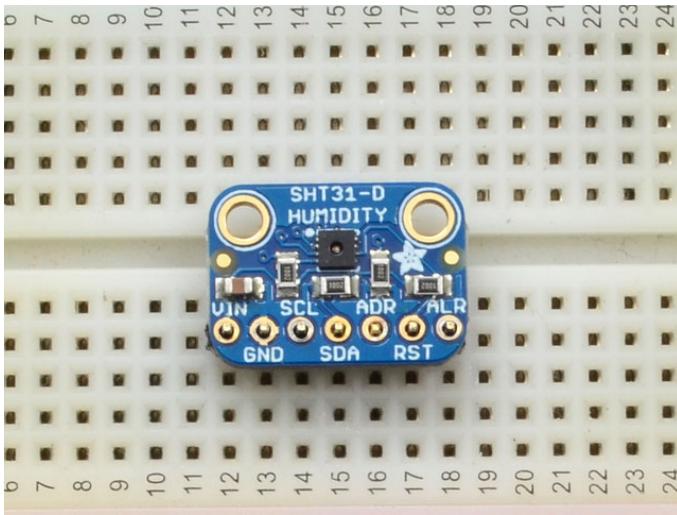
### Assembly

by [lady ada](#)



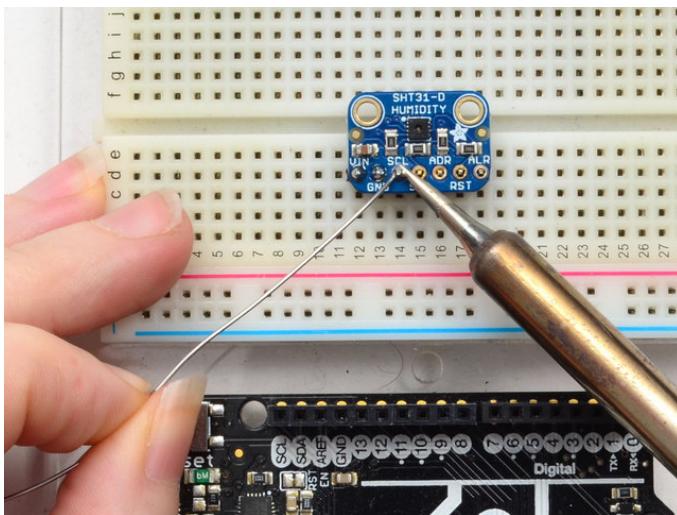
### Prepare the header strip:

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:

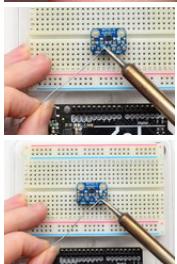
Place the breakout board over the pins so that the short pins poke through the breakout pads



### 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](#)).



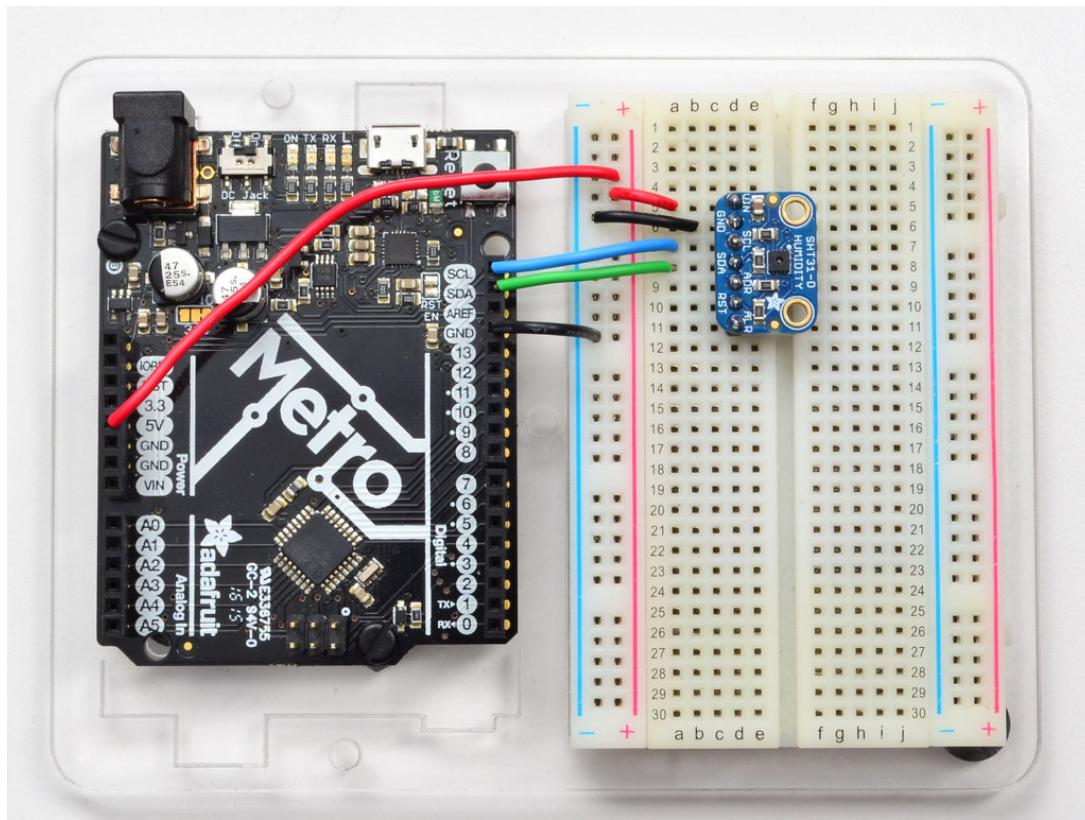


You're done! Check your solder joints visually and continue onto the next steps

## Arduino Code

by [lady ada](#)

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. 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 SHT31-D has a default I2C address of **0x44** which you can change to **0x45** by connecting the **ADR** pin to the **VIN** pin

## Download Adafruit\_SHT31

To begin reading sensor data, you will need to [download Adafruit\\_SHT31 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\\_SHT31 Library](#)

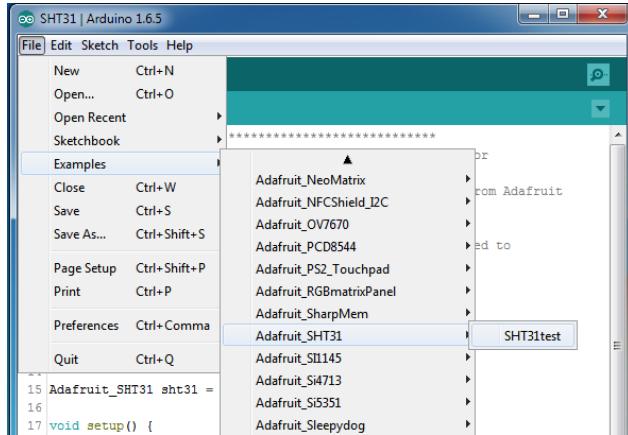
Rename the uncompressed folder **Adafruit\_SHT31** and check that the **Adafruit\_SHT31** folder contains **Adafruit\_SHT31.cpp** and **Adafruit\_SHT31.h**

Place the **Adafruit\_SHT31** library folder your **arduinosketchfolder/libraries/** folder.  
You may need to create the **libraries** subfolder if its your first library. Restart the IDE.

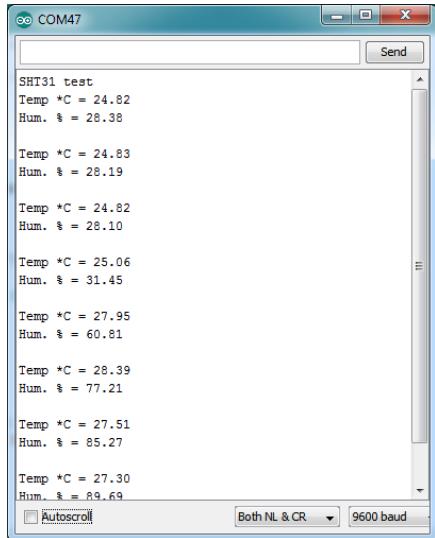
We also have a great tutorial on Arduino library installation at:  
<http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use>

## Load Demo

Open up **File->Examples->Adafruit\_SHT31->SHT31test** and upload to your Arduino wired up to the sensor



Thats it! Now open up the serial terminal window at 9600 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\_SHT31** object with:

[Download file](#)  
[Copy Code](#)

```
1. Adafruit_SHT31 sht31 = Adafruit_SHT31();
```

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

Then initialize the sensor with:

[Download file](#)  
[Copy Code](#)

```
1. sht31.begin(0x44)
```

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

The **0x44** is the i2c address you have the sensor set up for. By default its 0x44, you can also adjust the sensor for **0x45** and then pass that value in

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

[Download file](#)  
[Copy Code](#)

```
1. sht31.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

[Download file](#)

[Copy Code](#)

```
1. sht31.readHumidity()
```

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

We also have a few helper functions. Want to soft-reset the sensor? Use

[Download file](#)

[Copy Code](#)

```
1. sht31.reset()
```

There's also a heater built into the sensor, used to heat/evaporate any condensation. You can turn it on or off with

[Download file](#)

[Copy Code](#)

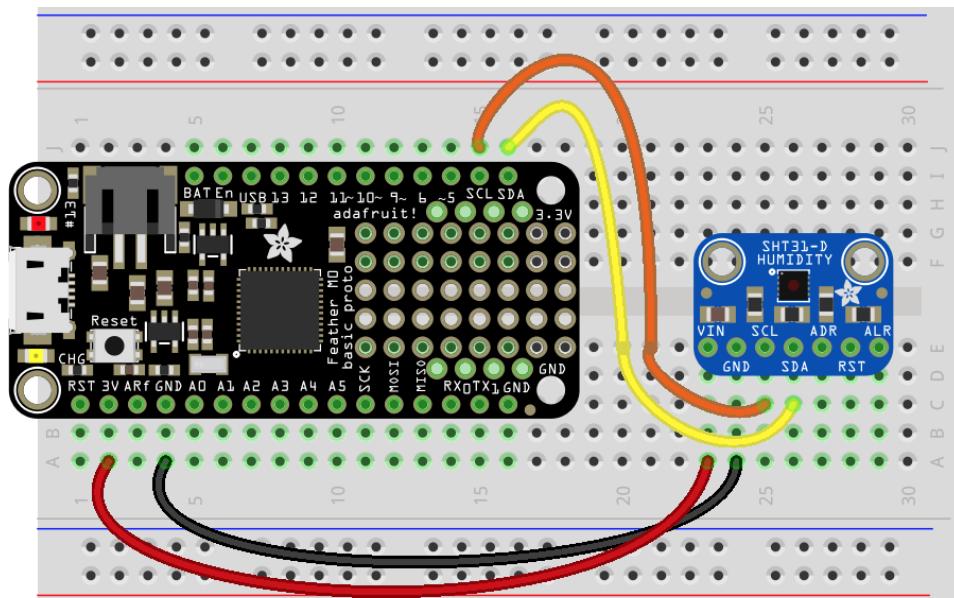
```
1. sht31.heater(true)  
2. sht31.heater(false)
```

## CircuitPython Code

by [Tony DiCola](#)

It's easy to use the SHT31-D sensor with CircuitPython and the [Adafruit CircuitPython SHT31D](#) module. This module allows you to easily write Python code that reads the humidity and temperature from the sensor.

First wire up a SHT31-D to your board exactly as shown on the previous pages for Arduino using an I2C connection. Here's an example of wiring a Feather M0 to the sensor with I2C:



fritzing

- 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 SHT31D](#) library on your CircuitPython board.

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](#). Our introduction 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, you'll need to manually install the necessary libraries from the bundle:

- `adafruit_sht31d.mpy`
- `adafruit_bus_device`

You can also download the `adafruit_sht31d.mpy` from [its releases page on Github](#).

Before continuing make sure your board's lib folder or root filesystem has the `adafruit_sht31d.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 humidity and temperature from the board's Python REPL.

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

[Download file](#)  
[Copy Code](#)

```
1. import board
2. import busio
3. import adafruit_sht31
4. i2c = busio.I2C(board.SCL, board.SDA)
5. sensor = adafruit_sht31.SHT31(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:

[Download file](#)  
[Copy Code](#)

```
1. import board
2. import bitbangio
3. import adafruit_sht31
4. i2c = bitbangio.I2C(board.SCL, board.SDA)
5. sensor = adafruit_sht31.SHT31(i2c)
```

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

- **relative\_humidity** - The relative humidity measured by the sensor, this is a value from 0-100%.
- **temperature** - The temperature measured by the sensor, a value in degrees Celsius.

[Download file](#)  
[Copy Code](#)

```
1. print('Humidity: {0}'.format(sensor.relative_humidity))
2. print('Temperature: {0}C'.format(sensor.temperature))

>>> print('Humidity: {0}'.format(sensor.relative_humidity))
Humidity: 38.8291%
>>> print('Temperature: {0}C'.format(sensor.temperature))
Temperature: 22.9545C
>>> █
```

That's all there is to using the SHT31D with CircuitPython!

Below is a complete example that measures the sensor readings and prints them every two seconds. Save this as **main.py** on your board and open the REPL to see the output. Remember to change to the **bitbangio** module if necessary for your board!

[Download sht31\\_simpletest.py](#) | [View on GitHub](#)  
[Copy Code](#)

```
1. import time
2. import board
3. import busio
4. import adafruit_sht31
5.
6. # Create library object using our Bus I2C port
7. i2c = busio.I2C(board.SCL, board.SDA)
8. sensor = adafruit_sht31.SHT31(i2c)
9.
10. loopcount = 0
11. while True:
12.     print("\nTemperature: {0:1f} C".format(sensor.temperature))
13.     print("Humidity: {0:1f} %".format(sensor.relative_humidity))
14.     loopcount += 1
15.     time.sleep(2)
16.     # every 10 passes turn on the heater for 1 second
17.     if loopcount == 10:
18.         loopcount = 0
19.         sensor.heater = True
20.         print("Sensor Heater status =", sensor.heater)
21.         time.sleep(1)
22.         sensor.heater = False
23.         print("Sensor Heater status =", sensor.heater)
```

## Downloads

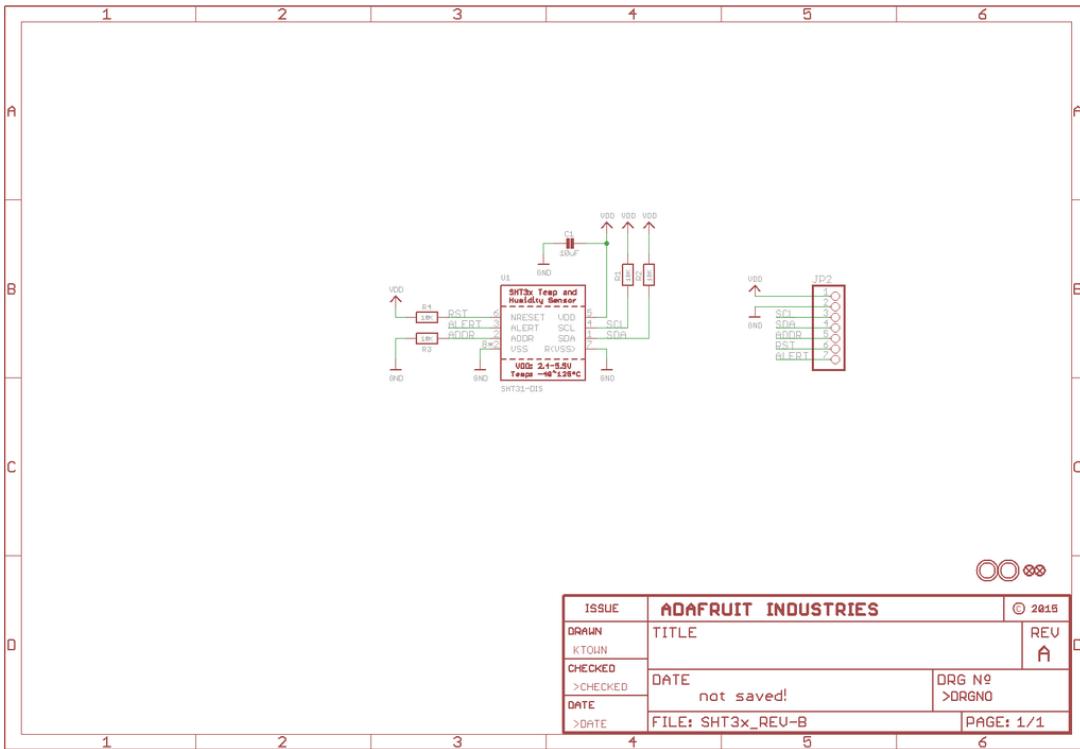
by [lady ada](#)

## Datasheets & Files

- [SHT31-DIS dataheet](#)
- [EagleCAD PCB Files on GitHub](#)
- [Fritzing object available in the Adafruit Fritzing Library](#)

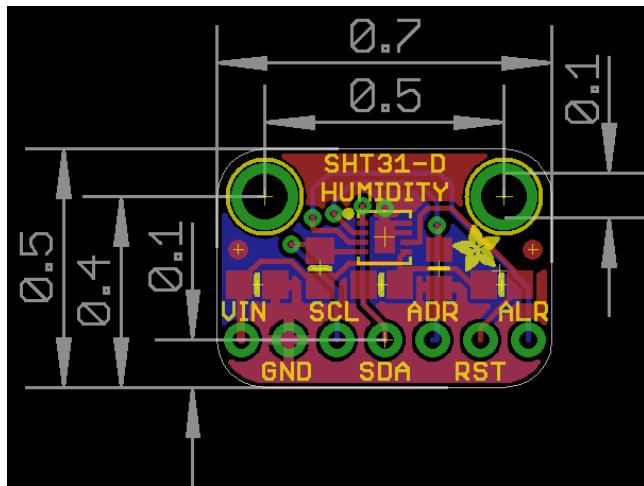
## Schematic

Click to enlarge



## Fabrication Print

Dimensions in inches



Adafruit Sensiron SHT31-D Temperature & Humidity Sensor Breakout  
\$13.95 [Add to Cart](#)

## RELATED GUIDES

[POPULAR](#)

[TTL Serial Camera](#)

Snap, Snap!  
by [lady ada](#)



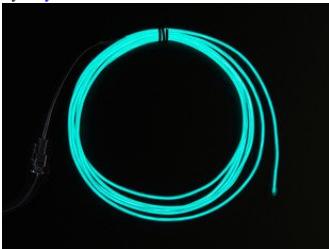
This guide is for our new TTL serial camera module with NTSC video output. These modules are a nice addition to a microcontroller project when you want to take a photo or control a video stream. The modules have a few features built in, such as the ability to change the brightness/saturation/hue of images, auto-contrast and auto-brightness adjustment, and motion detection.

[POPULAR](#)

[EL Wire](#)

[Working with Electroluminescent Wire](#)

by [lady ada](#)



EL Wire, also known as Electroluminescent wire, is a stiff wire core coated with phosphor and then covered with a protective PVC sheath. When an AC signal is applied to it, it glows a cool neon color.

[Find out how to solder, power, and work with EL Wire in your next project.](#)

[POPULAR](#)

[Hacking the Kinect](#)

[Reverse engineering the Microsoft Kinect](#)

by [lady ada](#)



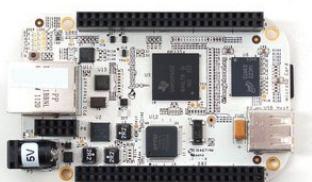
[Here's a step by step guide on how you can reverse engineer a Microsoft Kinect for the Xbox 360.](#)

[POPULAR](#)

[BeagleBone](#)

[Tutorials for the TI embedded Linux board](#)

by [lady ada](#)



New from the fine people who have brought us the Beagle Board, we now have a smaller, lighter, but powerful single board linux computer, Beagle Bone! We like this move to a more compact and integrated SBC. For example, there is onboard Ethernet and USB host, as well as a USB client interface (a FTDI chip for shell access). It even comes preloaded with Angstrom Linux on the 4 GB microSD card! Here are some tips and tricks to get your BeagleBone up and running.

X

**OUT OF STOCK NOTIFICATION**

YOUR NAME

YOUR EMAIL

[NOTIFY ME](#)

- [CONTACT](#)
- [SUPPORT](#)
- [DISTRIBUTORS](#)

- [EDUCATORS](#)
- [JOBS](#)
- [FAQ](#)
- [SHIPPING & RETURNS](#)
- [TERMS OF SERVICE](#)
- [PRIVACY & LEGAL](#)
- [ABOUT US](#)

[ENGINEERED IN NYC](#) Adafruit ®

"Be so good they can't ignore you" - [Steve Martin](#)

