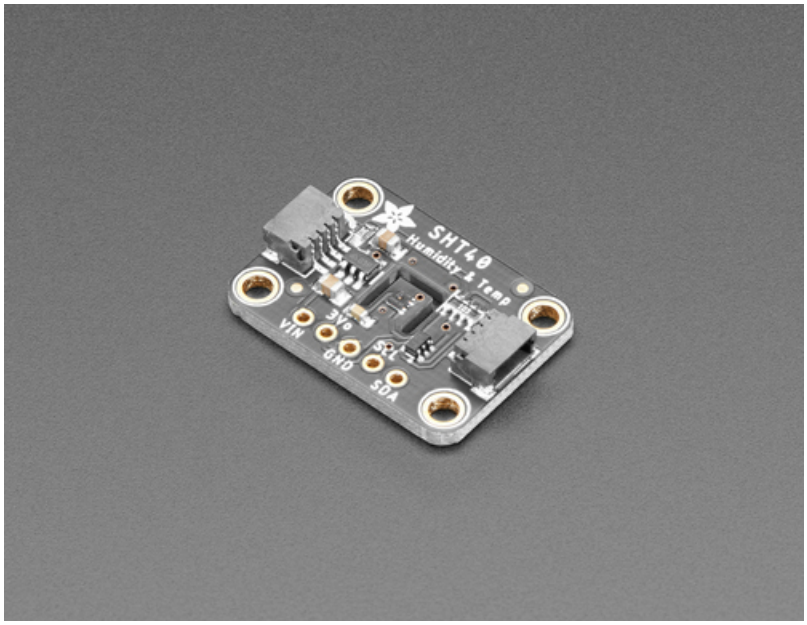
adafruit learning system

Adafruit Sensirion SHT40 Temperature & Humidity Sensor

Created by Kattni Rembor



Last updated on 2021-06-01 04:56:08 PM EDT

Guide Contents

Guide Contents 2
Overview 3
Pinouts 6
- Power Pins 6
- I2C Logic Pins 6

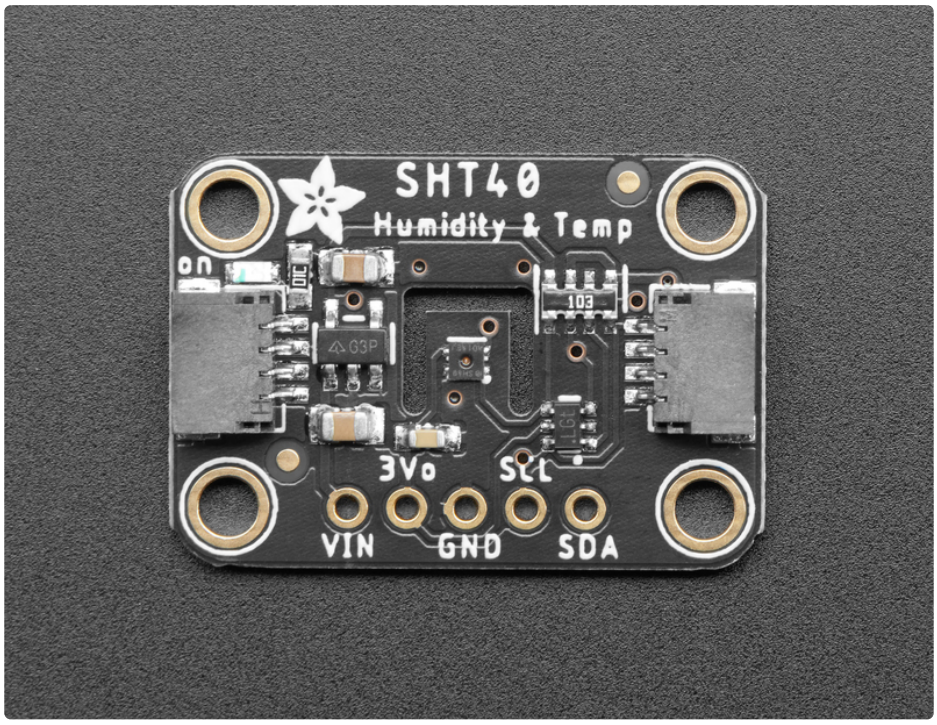
Arduino 7
- Wiring 7

Installation 7
Load Example 8
Example Code 9
Arduino Docs 12
Python & CircuitPython 13
- CircuitPython Microcontroller Wiring 13
- Python Computer Wiring 13
- CircuitPython Installation of SHT4x Library 14
- Python Installation of SHT4x Library 15
- CircuitPython & Python Usage 15
- Full Example Code 16

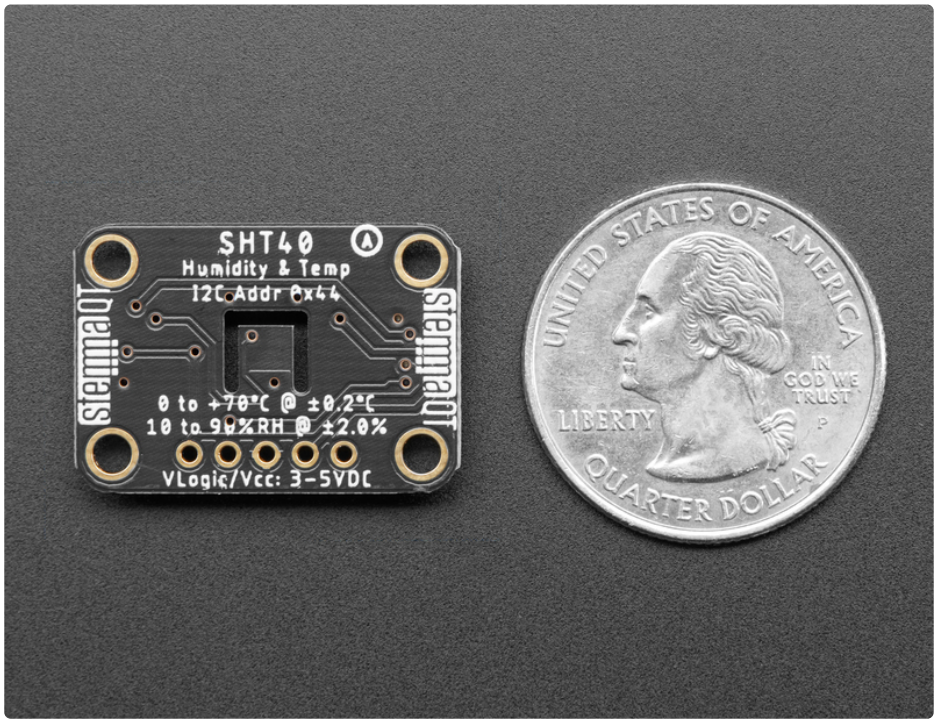
Python Docs 17
Downloads 18
- Files: 18

Schematic 18
Fab Print 18

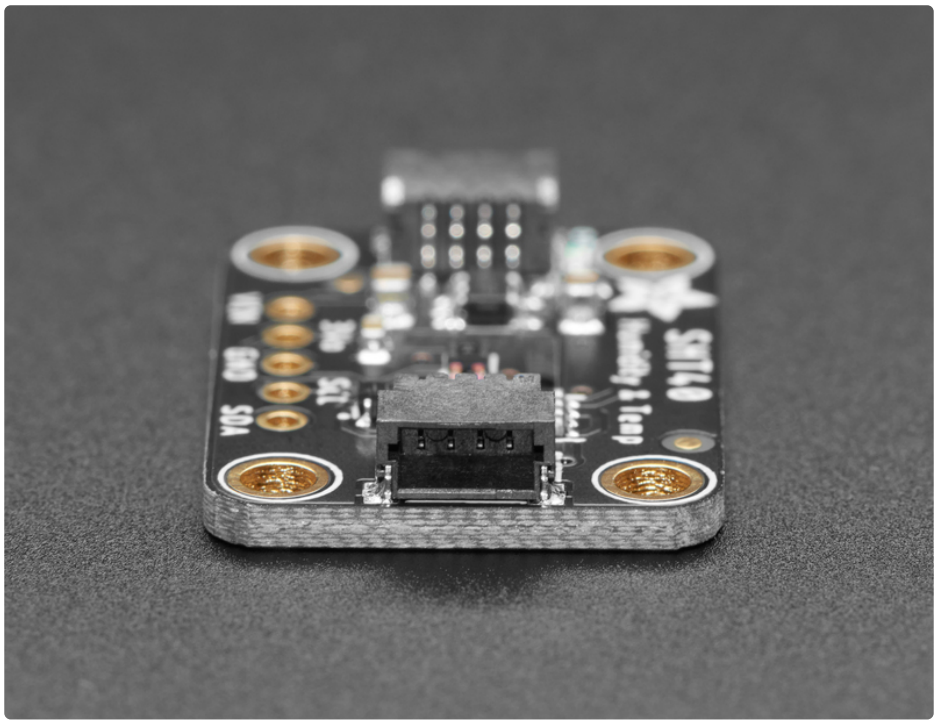
Overview



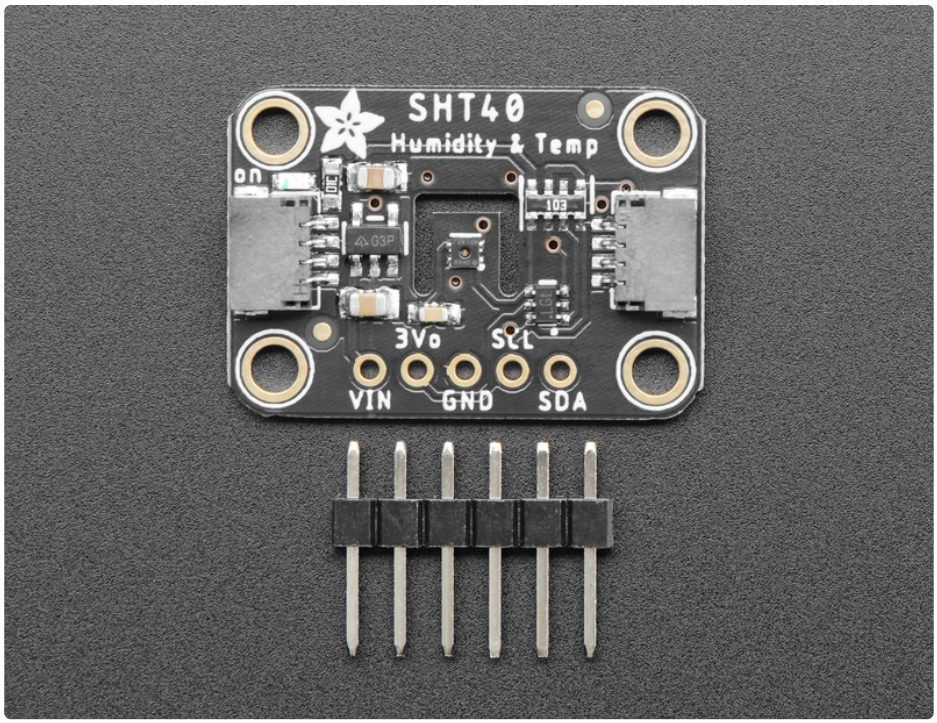
Sensirion Temperature/Humidity sensors are some of the finest & highest-accuracy devices you can get. And finally, we have some that have a true I2C interface for easy reading. The **SHT40** sensor is the fourth generation (started at the SHT10 and worked its way up to the top!). The SHT40 has an excellent ±1.8% typical relative humidity accuracy from 25 to 75% and ±0.2 °C typical accuracy from 0 to 75 °C.



Unlike some earlier SHT sensors, this sensor has a *true* I2C interface for easy interfacing with only two wires (plus power and ground!). Thanks to the voltage regulator and level shifting circuitry we've included on the breakout It is also is 3V or 5V compliant, so you can power and communicate with it using any microcontroller or microcomputer.



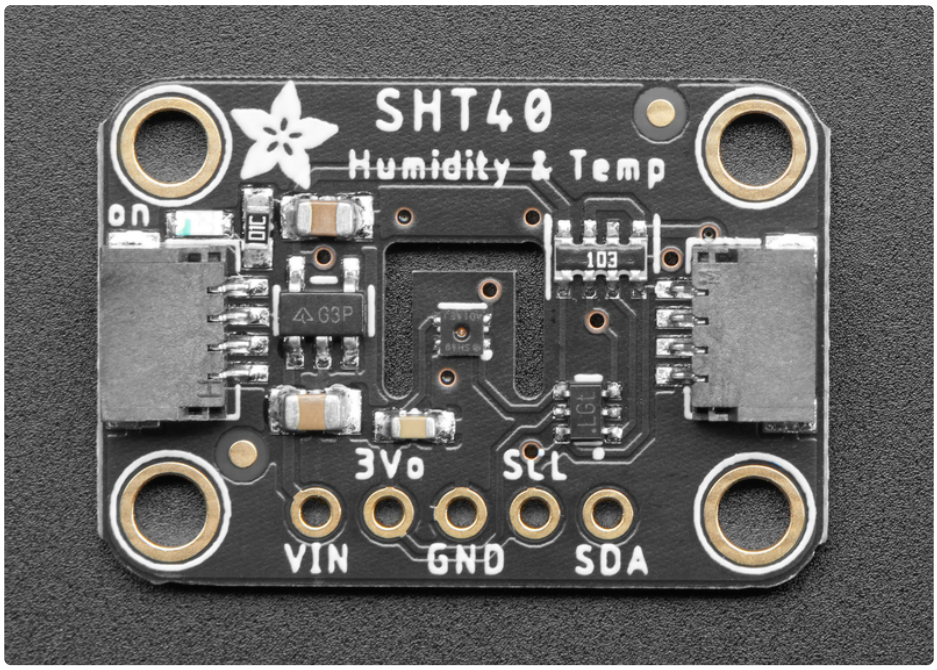
Such a lovely chip - so we spun up a breakout board with the SHT4x and some supporting circuitry such as pullup resistors and capacitors. To make things even easier, we've included SparkFun Qwiic (https://adafru.it/Fpw) compatible STEMMA QT (https://adafru.it/Ft4) connectors for the I2C bus so you don't even need to solder! QT Cable is not included, but we have a variety in the shop (https://adafru.it/JnB).



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.

We've written both Arduino and CircuitPython/Python library code for this chip, so you can use it with just about any microcontroller or single-board computer like Raspberry Pi.

Pinouts



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 microcontroller like Arduino, use 5V
- **3V** - 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

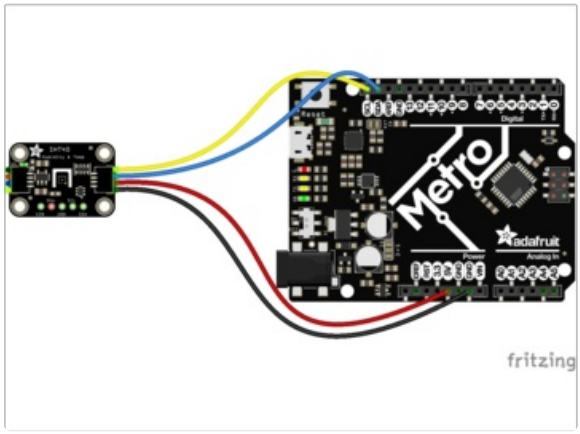
- **SCL** - I2C clock pin, connect to your microcontroller I2C clock line. This pin is level shifted so you can use 3-5V logic, and there's a **10K pullup** on this pin.
- **SDA** - I2C data pin, connect to your microcontroller I2C data line. This pin is level shifted so you can use 3-5V logic, and there's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

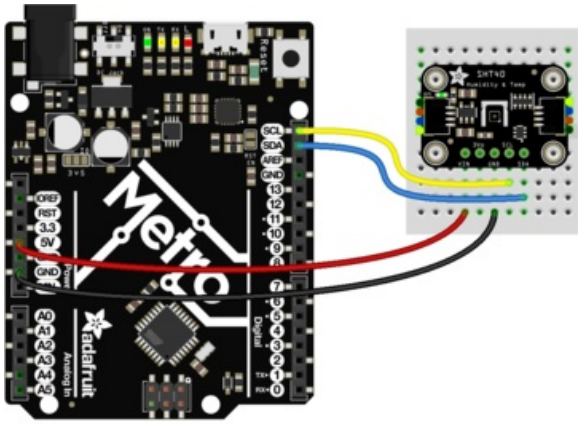
The I2C address is 0x44 and cannot be changed

Arduino

Wiring

Connecting the SHT40 to your Feather or Arduino is easy:



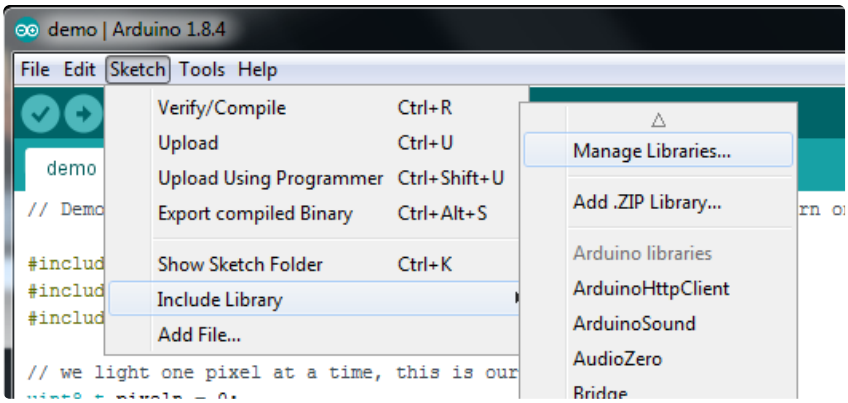


- If you are running a Feather (3.3V), connect **Feather 3V** to **board VIN**
- If you are running a 5V Arduino (Uno, etc.), connect **Arduino 5V** to **board VIN**
- Connect **Feather or Arduino GND** to **board GND**
- Connect **Feather or Arduino SCL** to **board SCL**
- Connect **Feather or Arduino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

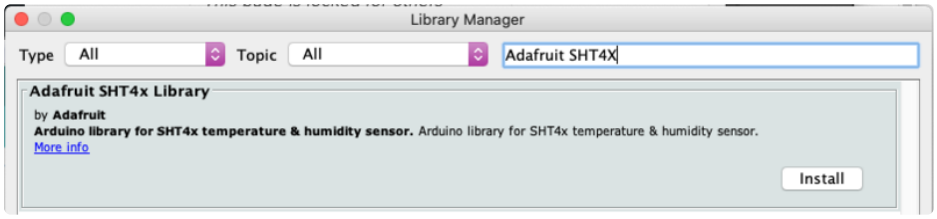
The final results should resemble the illustration above, showing an Adafruit Metro development board.

Installation

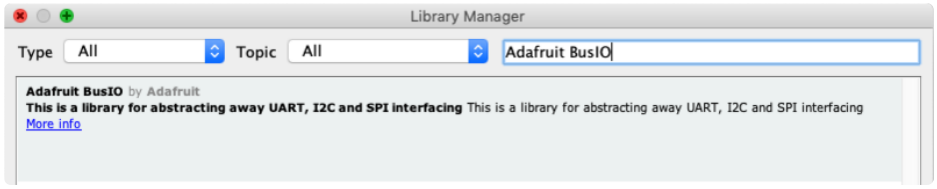
You can install the **Adafruit SHT4X Library** for Arduino using the Library Manager in the Arduino IDE:



Click the **Manage Libraries ...** menu item, search for **Adafruit SHT4X**, and select the **Adafruit SHT4X** library:



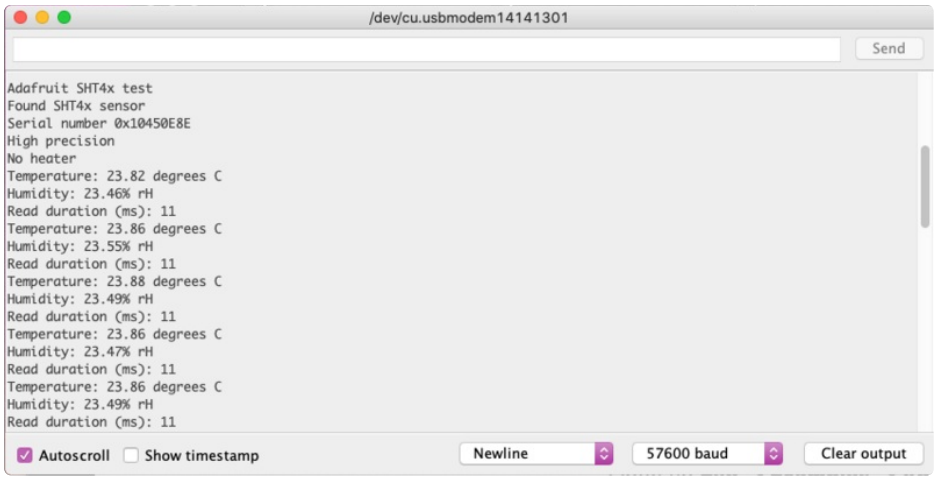
Then follow the same process for the **Adafruit BusIO** library.



Load Example

Open up **File -> Examples -> Adafruit SHT4X -> SHT4test** 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 SHT40 for the temperature and humidity values:

```
/***************************************************
  This is an example for the SHT4x Humidity & Temp Sensor

  Designed specifically to work with the SHT4x sensor from Adafruit
  ----> https://www.adafruit.com/products/4885

  These sensors use I2C to communicate, 2 pins are required to
  interface
 ***************************************************/

#include "Adafruit_SHT4x.h"

Adafruit_SHT4x sht4 = Adafruit_SHT4x();

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

  while (!Serial)
    delay(10);     // will pause Zero, Leonardo, etc until serial console opens

  Serial.println("Adafruit SHT4x test");
  if (! sht4.begin()) {
    Serial.println("Couldn't find SHT4x");
    while (1) delay(1);
  }
  Serial.println("Found SHT4x sensor");
  Serial.print("Serial number 0x");
  Serial.println(sht4.readSerial(), HEX);

  // You can have 3 different precisions, higher precision takes longer
  sht4.setPrecision(SHT4X_HIGH_PRECISION);
  switch (sht4.getPrecision()) {
     case SHT4X_HIGH_PRECISION:
```

```
      Serial.println("High precision");
      break;
    case SHT4X_MED_PRECISION:
      Serial.println("Med precision");
      break;
    case SHT4X_LOW_PRECISION:
      Serial.println("Low precision");
      break;
  }

  // You can have 6 different heater settings
  // higher heat and longer times uses more power
  // and reads will take longer too!
  sht4.setHeater(SHT4X_NO_HEATER);
  switch (sht4.getHeater()) {
    case SHT4X_NO_HEATER:
      Serial.println("No heater");
      break;
    case SHT4X_HIGH_HEATER_1S:
      Serial.println("High heat for 1 second");
      break;
    case SHT4X_HIGH_HEATER_100MS:
      Serial.println("High heat for 0.1 second");
      break;
    case SHT4X_MED_HEATER_1S:
      Serial.println("Medium heat for 1 second");
      break;
    case SHT4X_MED_HEATER_100MS:
      Serial.println("Medium heat for 0.1 second");
      break;
    case SHT4X_LOW_HEATER_1S:
      Serial.println("Low heat for 1 second");
      break;
    case SHT4X_LOW_HEATER_100MS:
      Serial.println("Low heat for 0.1 second");
      break;
  }

}


void loop() {
  sensors_event_t humidity, temp;

  uint32_t timestamp = millis();
  sht4.getEvent(&humidity, &temp);// populate temp and humidity objects with fresh data
  timestamp = millis() - timestamp;

  Serial.print("Temperature: "); Serial.print(temp.temperature); Serial.println(" degrees C");
  Serial.print("Humidity: "); Serial.print(humidity.relative_humidity); Serial.println("% rH");

  Serial.print("Read duration (ms): ");
  Serial.println(timestamp);

  delay(1000);
}
```

Arduino Docs

Arduino Docs (https://adafru.it/Qib)

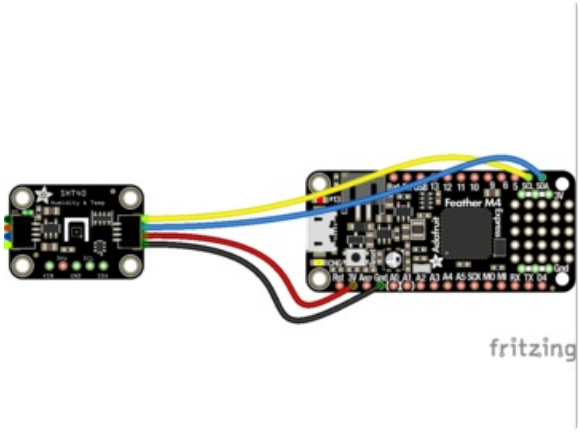
Python & CircuitPython

It's easy to use the Adafruit Sensirion SHT40 Temperature & Humidity Sensor with CircuitPython and the Adafruit CircuitPython SHT4x (https://adafru.it/Qid) module. This module allows you to easily write Python code that reads temperature and humidity data from the sensor.

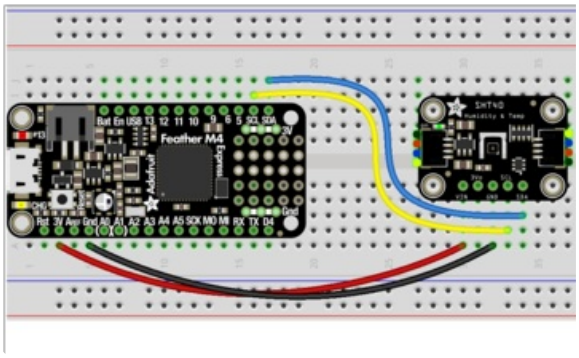
You can use this sensor with any CircuitPython microcontroller board or with a computer that has GPIO and Python thanks to Adafruit_Blinka, our CircuitPython-for-Python compatibility library (https://adafru.it/BSN).

CircuitPython Microcontroller Wiring

First wire up an SHT40 to your board exactly as follows. Here is an example of the SHT40 wired to a Feather using I2C:



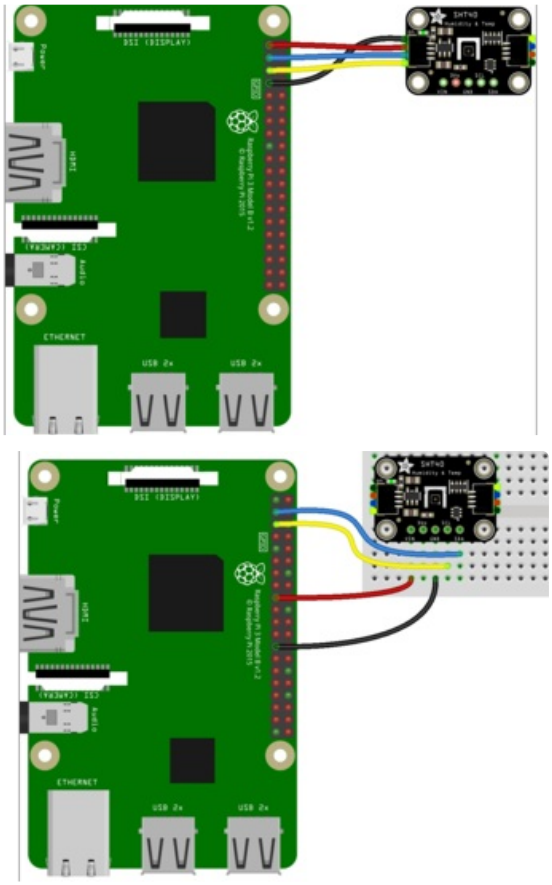
- **Board 3V** to **sensor VIN**
- **Board GND** to **sensor GND**
- **Board SCL** to **sensor SCL**
- **Board SDA** to **sensor SDA**



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 SHT4x Library

You'll need to install the Adafruit CircuitPython SHT4x (https://adafru.it/Qid) 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/ENC). Our CircuitPython starter guide has a great page on how to install libraries from the library bundle (https://adafru.it/ABU).

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

- **adafruit_sht4x.mpy**

Before continuing make sure your board's **lib** folder or root filesystem has the **adafruit_sht4x.mpy** file copied over.

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

Python Installation of SHT4x 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-sht4x`

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 from the sensor in 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_sht4x

sht = adafruit_sht4x.SHT4x(board.I2C())
```

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

- **relative_humidity** - The current relative humidity in % rH
- **temperature** - The current temperature in degrees celsius

```
print(sht.temperature)
print(sht.relative_humidity)
```

```
>>> print(sht.temperature)
23.8758
>>> print(sht.relative_humidity)
25.384
```

That's all there is to reading temperature and humidity from the SHT40 sensor!

Full Example Code

```
# SPDX-FileCopyrightText: Copyright (c) 2020 ladyada for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import board
import adafruit_sht4x

i2c = board.I2C()  # uses board.SCL and board.SDA
sht = adafruit_sht4x.SHT4x(i2c)
print("Found SHT4x with serial number", hex(sht.serial_number))

sht.mode = adafruit_sht4x.Mode.NOHEAT_HIGHPRECISION
# Can also set the mode to enable heater
# sht.mode = adafruit_sht4x.Mode.LOWHEAT_100MS
print("Current mode is: ", adafruit_sht4x.Mode.string[sht.mode])

while True:
    temperature, relative_humidity = sht.measurements
    print("Temperature: %0.1f C" % temperature)
    print("Humidity: %0.1f %%" % relative_humidity)
    print("")
    time.sleep(1)
```

Python Docs

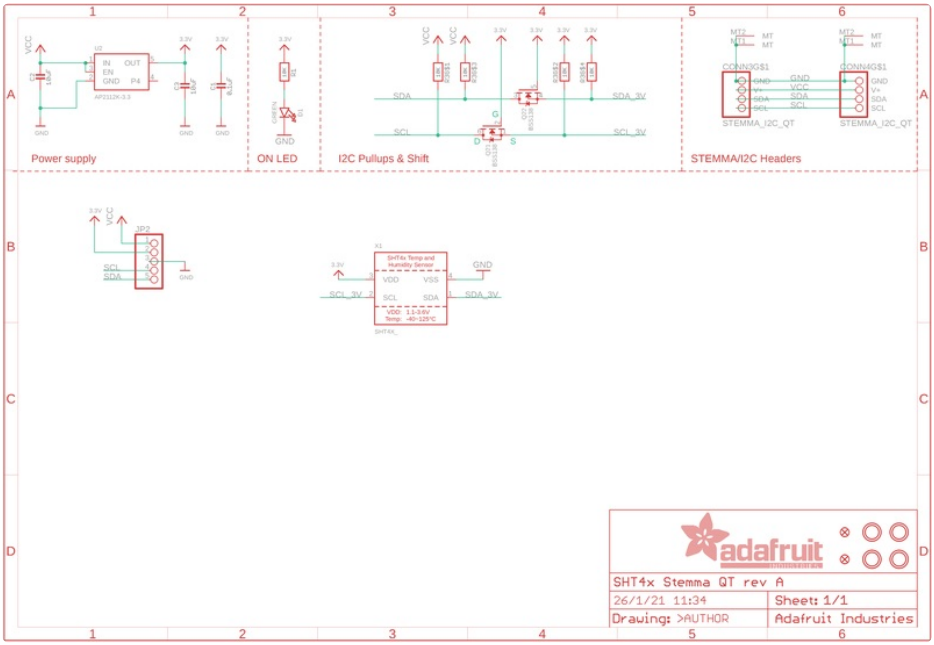
Python Docs (https://adafru.it/Qic)

Downloads

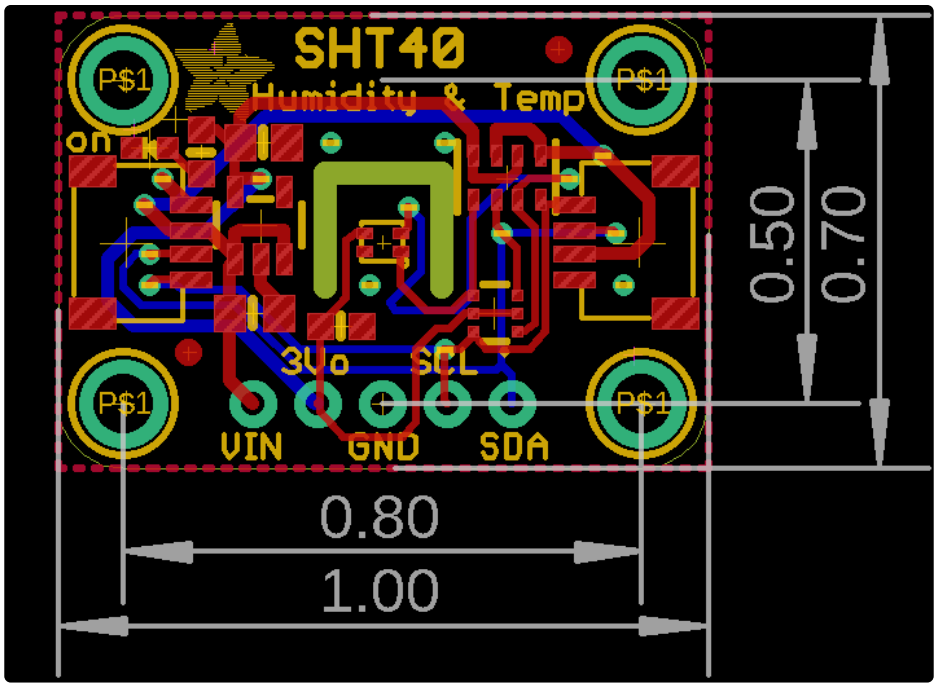
Files:

- SHT40 Datasheet (https://adafru.it/QhD)
- Fritzing object in the Adafruit Fritzing Library (https://adafru.it/QhE)
- EagleCAD PCB files on GitHub (https://adafru.it/QhF)

Schematic



Fab Print


SHT40
Humidity & Temp
P$1
P$1
P$1
P$1
on
3Vo
SCL
VIN
GND
SDA
0.50
0.70
0.80
1.00