# Table of Contents

## Overview

- Power Pins
- I2C Logic Pins
- Arduino
  - I2C Wiring
  - Library Installation
  - Load Example
  - Example Code
- Arduino Docs
  - CircuitPython Microcontroller Wiring
  - Python Computer Wiring
  - CircuitPython Installation of SHTC3 Library
  - Python Installation of SHTC3 Library
  - CircuitPython & Python Usage
  - Example Code
- Python Docs
  - Downloads
    - Files
    - Schematic
    - Fab Print
Overview

Sensirion Temperature/Humidity sensors are some of the finest & highest-accuracy devices you can get. And with a true I2C interface, reading the data is for easy. The SHTC3 sensor has an excellent ±2% relative humidity and ±0.2°C accuracy for most uses.

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.

![Image of SHTC3 breakout board]

Such a lovely chip - so we spun up a breakout board with the SHTC3 and some supporting circuitry such as pullup resistors and capacitors. To make things even easier, we've included SparkFun Qwiic compatible STEMMA QT connectors for the I2C bus so you don't even need to solder! 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.
Pinouts

Power Pins

- Vin - this is the power pin. Since the sensor 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 microcontroller like Arduino, use 5V
- 3Vo - 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 - this is the I2C clock pin, connect to your microcontroller's I2C clock line.
- SDA - this is the I2C data pin, connect to your microcontroller's I2C data line
- STEMMA QT () - These connectors allow you to connectors to dev boards with STEMMA QT connectors or to other things with various associated accessories ()

Arduino

Using the SHTC3 with Arduino is a simple matter of wiring up the sensor to your Arduino-compatible microcontroller, installing the Adafruit SHTC3 () library we've written, and running the provided example code.
I2C Wiring

Use this wiring if you want to connect via I2C interface. The I2C address for the SHTC3 is 0x70.

Here is how to wire up the sensor using one of the STEMMA QT connectors. The examples show a Metro but wiring will work the same for an Arduino or other compatible board.

Connect board VIN (red wire) to Arduino 5V if you are running a 5V board Arduino (Uno, etc.). If your board is 3V, connect to that instead.
Connect board GND (black wire) to Arduino GND
Connect board SCL (yellow wire) to Arduino SCL
Connect board SDA (blue wire) to Arduino SDA

Here is how to wire the sensor to a board using a solderless breadboard:

Connect board VIN (red wire) to Arduino 5V if you are running a 5V board Arduino (Uno, etc.). If your board is 3V, connect to that instead.
Connect board GND (black wire) to Arduino GND
Connect board SCL (yellow wire) to Arduino SCL
Connect board SDA (blue wire) to Arduino SDA

Library Installation

You can install the Adafruit SHTC3 library for Arduino using the Library Manager in the Arduino IDE.
Click the Manage Libraries ... menu item, search for Adafruit SHTC3, and select the Adafruit SHTC3 library:

Then follow the same process for the Adafruit BusIO library.

Finally follow the same process for the Adafruit Unified Sensor library:

Load Example

Open up File -> Examples -> Adafruit SHTC3 -> SHTC3test

After opening the demo file, upload to your Arduino wired up to the sensor. Once you upload the code, you will see the Temperature and Humidity values being printed when you open the Serial Monitor (Tools->Serial Monitor) at 115200 baud, similar to this:

```
SHTC3 test
Found SHTC3 sensor
Temperature: 31.53 degrees C
Humidity: 31.60% RH
Temperature: 31.54 degrees C
Humidity: 31.18% RH
Temperature: 31.50 degrees C
Humidity: 30.42% RH
```
Example Code

```c
#include "Adafruit_SHTC3.h"

Adafruit_SHTC3 shtc3 = Adafruit_SHTC3();

void setup() {
  Serial.begin(115200);
  while (!Serial)
    delay(10); // will pause Zero, Leonardo, etc until serial console opens
  Serial.println("SHTC3 test");
  if (! shtc3.begin()) {
    Serial.println("Couldn't find SHTC3");
    while (1) delay(1);
  }
  Serial.println("Found SHTC3 sensor");
}

void loop() {
  sensors_event_t humidity, temp;
  shtc3.getEvent(&humidity, &temp); // populate temp and humidity objects with fresh data
  Serial.print("Temperature: "); Serial.print(temp.temperature); Serial.println(" degrees C");
  Serial.print("Humidity: "); Serial.print(humidity.relative_humidity); Serial.println("% rH");
  delay(1000);
}
```

Arduino Docs

Python & CircuitPython

It's easy to use the SHTC3 with Python or CircuitPython, and the Adafruit CircuitPython SHTC3 module. This module allows you to easily write Python code that reads humidity and temperature from the SHTC3 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://circuitpython.org/docs/).

**CircuitPython Microcontroller Wiring**

First wire up a SHTC3 to your board exactly as shown below. Here's an example of wiring a Feather M4 to the sensor with I2C using one of the handy [STEMMA QT](https://www.adafruit.com/product/4747) connectors:

- 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)

You can also use the standard 0.100" pitch headers to wire it up on a 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'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.
Here's the Raspberry Pi wired to the sensor using I2C and a STEMMA QT (connector):

![Diagram of Raspberry Pi wired to sensor](image1)

- Pi 3V to sensor VCC (red wire)
- Pi GND to sensor GND (black wire)
- Pi SCL to sensor SCL (yellow wire)
- Pi SDA to sensor SDA (blue wire)

Finally here is an example of how to wire up a Raspberry Pi to the sensor using a solderless breadboard:

![Diagram of Raspberry Pi wired to sensor with breadboard](image2)

- Pi 3V to sensor VCC (red wire)
- Pi GND to sensor GND (black wire)
- Pi SCL to sensor SCL (yellow wire)
- Pi SDA to sensor SDA (blue wire)

CircuitPython Installation of SHTC3 Library

You'll need to install the Adafruit CircuitPython SHTC3 () 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 CircuitPython starter guide has a great page on how to install the library bundle ().
Before continuing make sure your board's lib folder or root filesystem has the adafruit_shtc3.mpy file and adafruit_bus_device folder copied over.

Next connect to the board's serial REPL so you are at the CircuitPython >>> prompt.

Python Installation of SHTC3 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!

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

- `sudo pip3 install adafruit-circuitpython-shtc3`

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 measurements from the board's Python REPL.

The SHTC3 does not like being restarted without having the power disconnected and reconnected and will raise an OSError if that happens. We are investigating a better solution, however for now you will need to disconnect and reconnect power to the SHTC3 each time you wish to re-run your script.

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

```python
import board
import adafruit_shtc3

i2c = board.I2C()

sht = adafruit_shtc3.SHTC3(i2c)
```
Now you’re ready to read values from the sensor using 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.

```python
print("Temperature: %0.1f C" % sht.temperature)
print("Humidity: %0.1f %%fH" % sht.relative_humidity)
```

We’ve also added a `measurements` property that simultaneously reads the temperature and relative humidity properties and returns them as a `(temperature, relative_humidity)` tuple:

```python
print("Temperature: %0.1f C Humidity: %0.1f %%%rH" % sht.measurements)
```

Example Code

```python
# SPDX-FileCopyrightText: Copyright (c) 2020 Bryan Siepert for Adafruit Industries
# SPDX-License-Identifier: MIT
import time
import board
import adafruit_shtc3

i2c = board.I2C()  # uses board.SCL and board.SDA
# i2c = board.STEMMA_I2C()  # For using the built-in STEMMA QT connector on a
# microcontroller
sht = adafruit_shtc3.SHTC3(i2c)

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

Downloads

Files

- SHTC3 Datasheet ()
- EagleCAD files on GitHub ()
- 3D models on GitHub ()
- Fritzing object in the Adafruit Fritzing Library ()

Schematic
Fab Print