Adafruit HTS221 - Temperature & Humidity Sensor

Created by Bryan Siepert
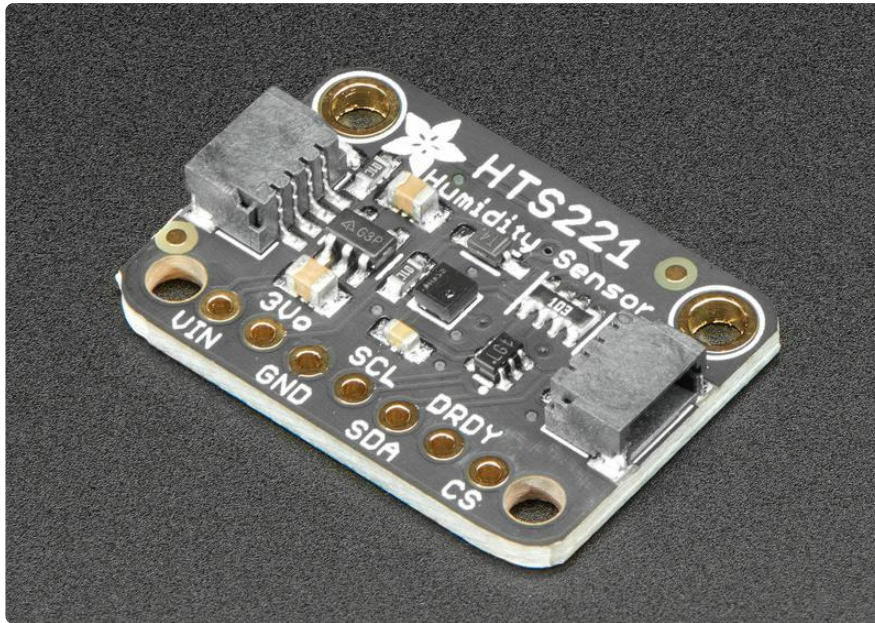

https://learn.adafruit.com/adafruit-hts221-temperature-humidity-sensor

Last updated on 2021-11-15 07:59:22 PM EST

Table of Contents

Overview

Humidity is the concentration of water vapor (water when it's a gas) in the air. Some places have more and make you feel soggy and hot like you're in a sauna. Other places don't have much, which makes it easier to get nose bleeds and static zaps. These are just a few examples of how humidity can have a noticeable impact on things that grow, like mushrooms or humans, as well as things that are sensitive to moisture, like delicate electronics, or humans.
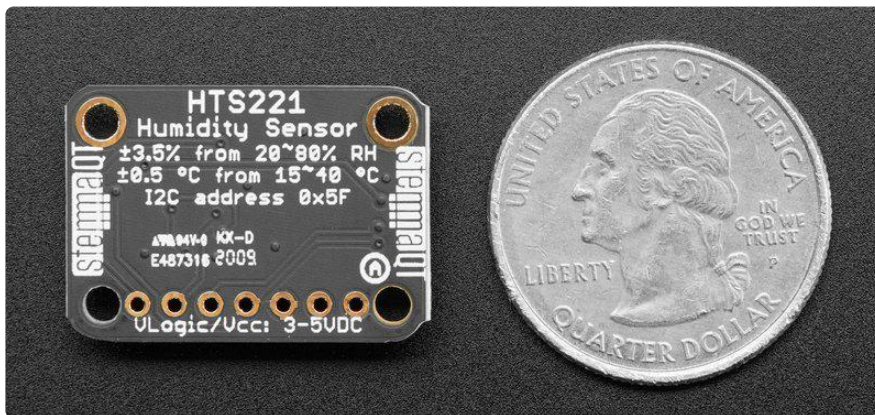
This is where the HTS221 Temperature and Humidity Sensor by ST comes in. With an HTS221, you can take measurements of the relative humidity which will tell you if the environment you're measuring is bone dry or sopping wet. The HTS221 can measure relative humidity from 0%-100% rH with a sensitivity of 0.004% and 3.5% accuracy. It can also measure temperature from -40 to 120 degrees C, with a resolution of 0.016°C with ±0.5 °C accuracy between 15 and +40 °C.

The HTS221 is an inexpensive humidity sensor which makes it great for every day measurement applications. Want to measure the humidity of your bathroom? Sure! How about your mushroom farm? Perfect!
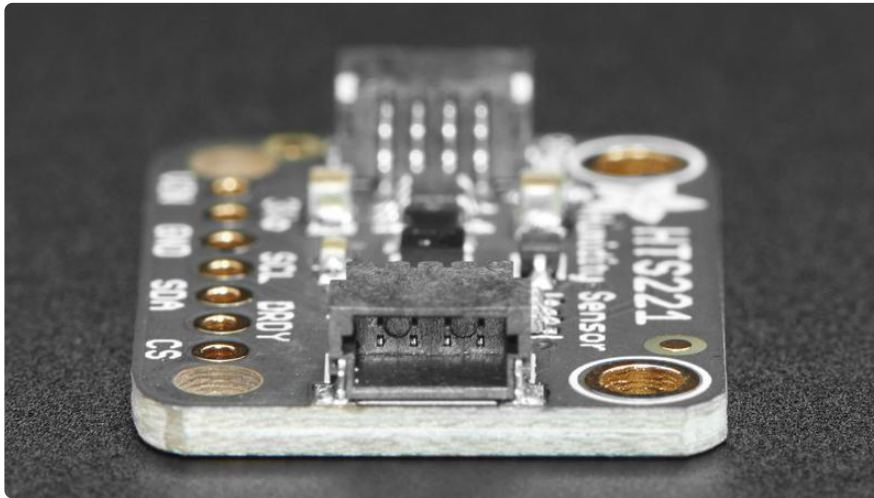
The HTS221 lets you choose between a 3-wire SPI or the ever-popular I2C which only needs two wires for data. If you're an I2C fan (who isn't?), we've included our handy dandy SparkFun Qwiic (https://adafru.it/Fpw) compatible STEMMA QT (https://adafru.it/Ft4) connectors for the I2C bus so you don't even need to solder!

Since the HTS221 is a super small 2mm square, we've put it on a breakout board with level shifting and a power regulator, so it's easy to use on a breadboard with either a 5V logic level device like a Metro 328 or Arduino Uno, or with 3.3V logic level devices like a Feather M4 or Raspberry Pi of your choice.

Lastly, it wouldn't be an Adafruit breakout if it didn't come with libraries for Arduino and CircuitPython or Python. We've also written example code and instructions on the pages that follow (https://adafru.it/JEd), so you'll be ready to tell all your friends the approximate relative humidity level and temperature in only a few minutes.

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 micro 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 - I2C clock pin, connect to your microcontroller's 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's 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 make I2C connections to dev boards with STEMMA QT connectors or to other things with various associated accessories (https://adafru.it/Ft6)

SPI Logic pins:

> Because the HTS221 is only capable of 3-pin SPI, take care when reading the pin descriptions below and consult the wiring diagram on the Arduino page

All pins going into the breakout have level shifting circuitry to make them 3-5V logic level safe. Use whatever logic level is on Vin!

- SCL - This is the SPI Clock pin, its an input to the chip
- SDA - For 3-pin SPI, this pin does double duty and is connected to your device's MISO and MOSI pins:

    - MISO - Connect directly to the HTS221's SDA pin
    - MOSI - Connect to SDA through a 1K resistor

- CS - this is the Chip Select pin, drop it low to start an SPI transaction. Its an input to the chip

If you want to connect multiple HTS221's to one microcontroller, have them share the SCL and SDA pins. Then assign each one a unique CS pin.
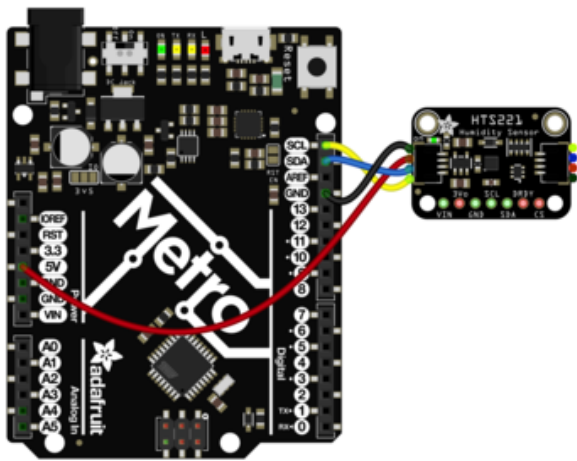
Other Pins

- DRDY - Data ready pin. The HTS221 can be configured to change the state of this pin to signify that new pressure or temperature measurements are available
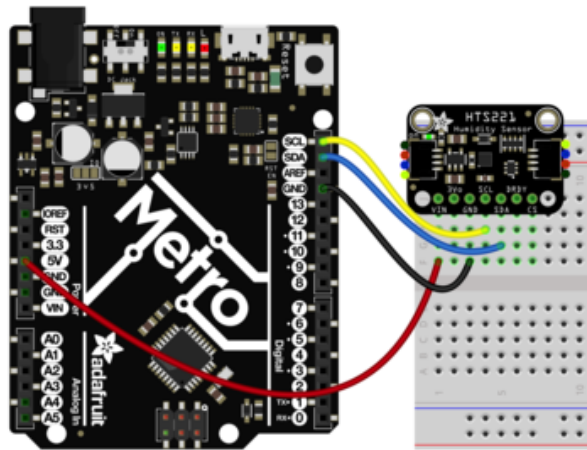

Arduino

I2C Wiring

Use this wiring if you want to connect via I2C interface

The HTS221's I2C address is 0x5F.

- 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

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

Because the HTS221 is only capable of 3-pin SPI, take care when reading the wiring instructions below and consult the wiring diagram to verify your connections
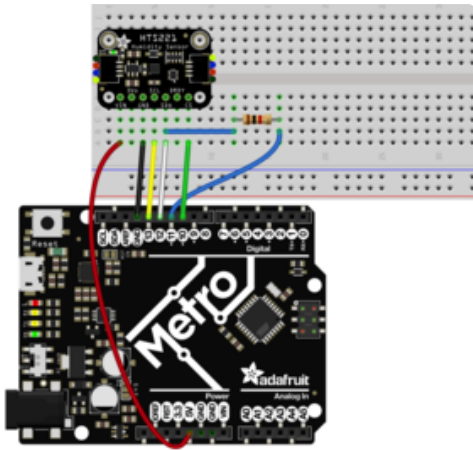
- Connect Vin to the power supply, 3V or 5V is fine. Use the same voltage that the microcontroller logic is based off of
- Connect GND to common power/ data ground
- Connect the SCL pin to Digital #13 but any pin can be used later
- Connect the SDA pin to Digital #12 but any pin can be used later
- Make a second connection to the SDA pin, through a 1K resistor from Digital #11. Any pin can be used later but it must be connected to SDA through a 1K resistor
- Connect the CS pin Digital #10 but any pin can be used later

Later on, once we get it working, we can adjust the library to use hardware SPI if you desire, or change the pins to others.
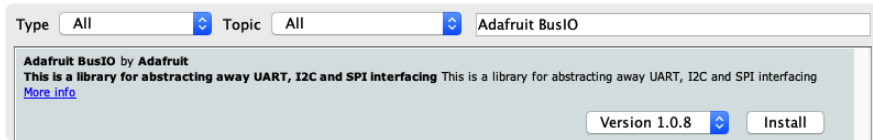
Library Installation

You can install the Adafruit HTS221 Library for Arduino using the Library Manager in the Arduino IDE.
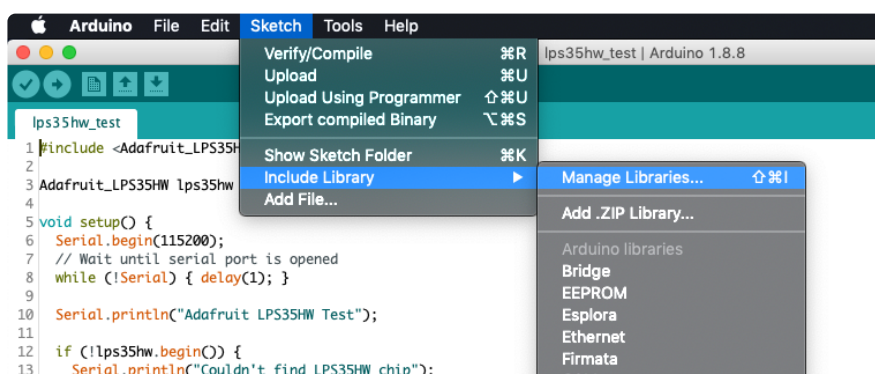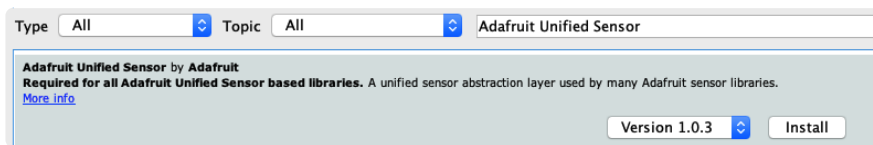

Click the Manage Libraries ... menu item, search for Adafruit HTS221 and select the Adafruit HTS221 library:
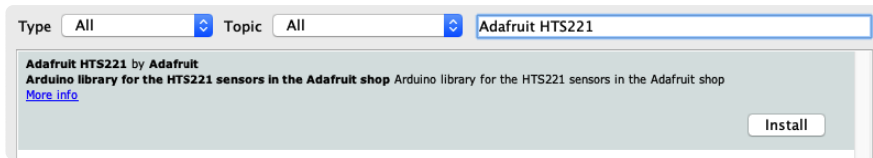
Then follow the same process for the Adafruit BusIO library.

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

Load Example

Open up File -> Examples -> Adafruit HTS221 -> adafruit_hts221_test and upload to your Arduino wired up to the sensor.

Depending on whether you are using I2C or SPI, change the pin names and comment or uncomment the following lines.

```
if (!hts.begin_I2C()) {
//  if (!hts.begin_SPI(HTS_CS)) {
//  if (!hts.begin_SPI(HTS_CS, HTS_SCK, HTS_MISO, HTS_MOSI)) {
```

Once you upload the code and open the Serial Monitor (Tools->Serial Monitor) at 1152 00 baud, you will see the current configuration printed, followed by the humidity and temperature measurements. You should see something similar to this:

```
Adafruit HTS221 test!
HTS221 Found!
Data rate set to:  12.5 Hz
Temperature: 26.19 degrees C
Humidity: 32.03% rH
Temperature: 26.21 degrees C
Humidity: 32.06% rH
Temperature: 26.23 degrees C
Humidity: 32.02% rH
```

The moisture from your breath should make the readings change, so give it a try by breathing on the sensor (the tiny black square in the center of the breakout) and seeing what happens!

```cpp
// Basic demo for reading Humidity and Temperature
#include <Wire.h>
#include <Adafruit_HTS221.h>
#include <Adafruit_Sensor.h>

// For SPI mode, we need a CS pin
#define HTS_CS 10
// For software-SPI mode we need SCK/MOSI/MISO pins
#define HTS_SCK 13
#define HTS_MISO 12
#define HTS_MOSI 11

Adafruit_HTS221 hts;
void setup(void) {
  Serial.begin(115200);
  while (!Serial) delay(10);     // will pause Zero, Leonardo, etc until serial
console opens

  Serial.println("Adafruit HTS221 test!");

  // Try to initialize!
  if (!hts.begin_I2C()) {
//  if (!hts.begin_SPI(HTS_CS)) {
//  if (!hts.begin_SPI(HTS_CS, HTS_SCK, HTS_MISO, HTS_MOSI)) {
    Serial.println("Failed to find HTS221 chip");
    while (1) { delay(10); }
  }
  Serial.println("HTS221 Found!");

//  hts.setDataRate(HTS221_RATE_1_HZ);
  Serial.print("Data rate set to: ");
  switch (hts.getDataRate()) {
   case HTS221_RATE_ONE_SHOT: Serial.println("One Shot"); break;
   case HTS221_RATE_1_HZ: Serial.println("1 Hz"); break;
   case HTS221_RATE_7_HZ: Serial.println("7 Hz"); break;
   case HTS221_RATE_12_5_HZ: Serial.println("12.5 Hz"); break;
  }

}

void loop() {

  sensors_event_t temp;
  sensors_event_t humidity;
  hts.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(500);
}
```

# Arduino Docs

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

# Python & CircuitPython

It's easy to use the HTS221 sensor with Python and CircuitPython, and the Adafruit CircuitPython HTS221 (https://adafru.it/JDV) module. This module allows you to easily write Python code that reads humidity and temperature measurements.

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 a HTS221 to your board for an I2C connection, exactly as shown below. Here's an example of wiring a Feather M4 to the sensor with I2C:
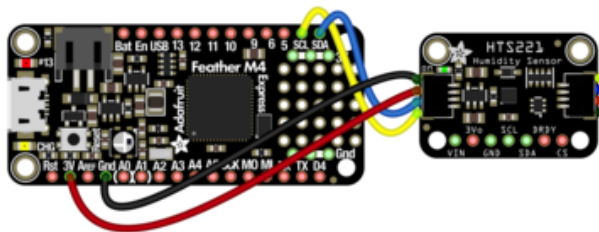


- Board 3V to sensor VIN (red wire)
- Board GND to sensor GND (black wire)
- Board SCL to sensor SCL (yellow wire)
- Board SDA to sensor SDA (blue wire)
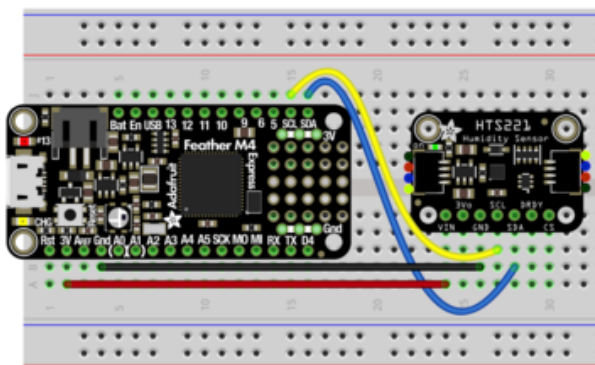
# Python Computer Wiring

Since there are 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:
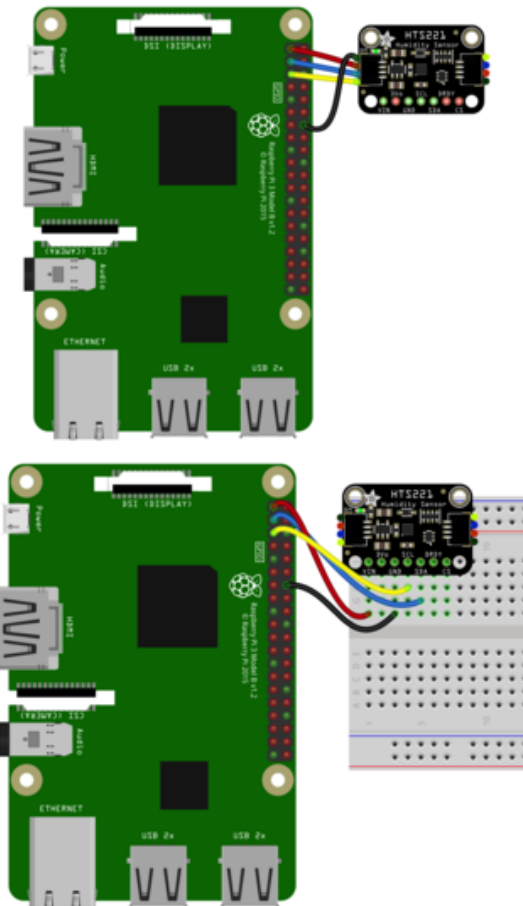


- Pi 3V to sensor VCC (red wire)
- Pi GND to sensor GND (black wire)
- Pi SCL to sensor SCL (yellow wire)
- Pi SDA to sensor SDA (blue wire)



## CircuitPython Installation of HTS221 Library

You'll need to install the Adafruit CircuitPython HTS221 (https://adafru.it/JDV) 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 the library bundle (https://adafru.it/ABU).

For non-express boards like the Trinket M0 or Gemma M0, you'll need to manually install the necessary libraries from the bundle:

- adafruit_hts221.mpy
- adafruit_bus_device
- adafruit_register

Before continuing make sure your board's lib folder or root filesystem has the adafruit_hts221.mpy, adafruit_bus_device, and adafruit_register files and folders copied over.

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

## Python Installation of HTS221 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:

- `sudo pip3 install adafruit-circuitpython-hts221`

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 humidity and temperature measurements from the board's Python REPL.

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

```python
import time
import board
import busio
import adafruit_hts221

i2c = busio.I2C(board.SCL, board.SDA)
hts = adafruit_hts221.HTS221(i2c)
```

```
>>> import time
>>> import board
>>> import busio
>>> import adafruit_hts221
>>> i2c = busio.I2C(board.SCL, board.SDA)
>>> hts = adafruit_hts221.HTS221(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("Relative Humidity: %.2f %% rH" % hts.relative_humidity)
print("Temperature: %.2f C" % hts.temperature)
```

```
>>> print("Relative Humidity: %.2f %% rH" % hts.relative_humidity)
Relative Humidity: 44.02 % rH
>>> print("Temperature: %.2f C" % hts.temperature)
Temperature: 22.88 C
```

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

import time
import board
import adafruit_hts221

i2c = board.I2C()
hts = adafruit_hts221.HTS221(i2c)

while True:
    print("Relative Humidity: %.2f %% rH" % hts.relative_humidity)
    print("Temperature: %.2f C" % hts.temperature)
    print("")
    time.sleep(1)
```
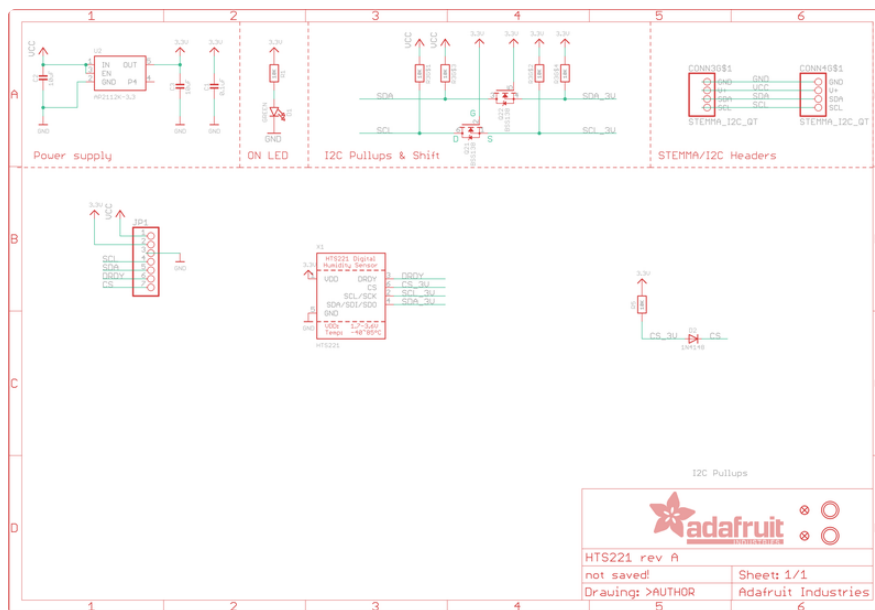
Python Docs

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

Downloads

Files

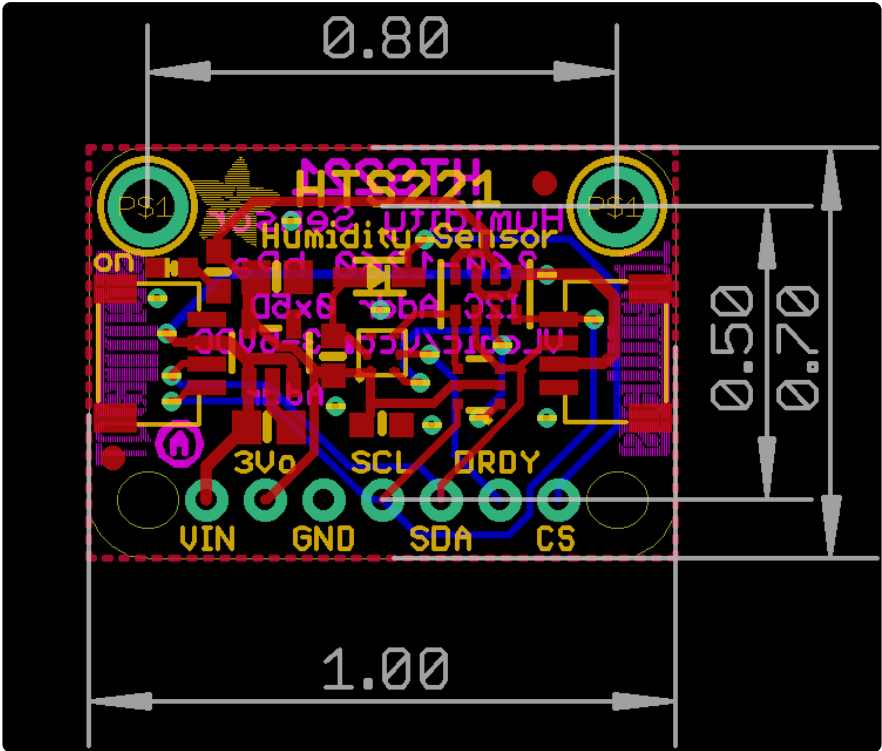- HTS221 Datasheet (https://adafru.it/JDW)
- EagleCAD files on GitHub (https://adafru.it/JDX)
- Fritzing object in Adafruit Fritzing Library (https://adafru.it/JDY)

Schematic

Fab Print

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Multiple Function Sensor Development Tools *category:*

Click to view products by Adafruit *manufacturer:*

Other Similar products are found below :

MAXWSNENV# STEVAL-MKIT01V1 KT-O2-25%-TB200A-E KT-TVOC-200-TB200A KT-NmHc-200-TB200A SEN0344 PIM520 PIM518 PIM519 PIM502 AS7022-EVALKIT ALTEHTG2SMIP MAX30101WING# OB1203SD-U-EVK MIKROE-4265 A000070 EV_ICG-20660L GX-F12A-P GX-F15A GX-F8AI-P GX-H15AI-P GX-H6A-P GX-HL15B-P 1093 MIKROE-2455 MIKROE-2458 MIKROE-2507 MIKROE-2508 MIKROE-2516 MIKROE-2529 1458 MIKROE-1628 176 189 1893 2106 ATQT4-XPRO GP30-DEMO MODULE GX-F12AI-P GX-F15A-P GX-FL15B-P GX-H12AI-P GX-H15A-P GX-H6AI-P GX-H8A-P GX-F15AI-P GX-FL15A-P AAS-AQS-UNO DFR0018 DFR0131