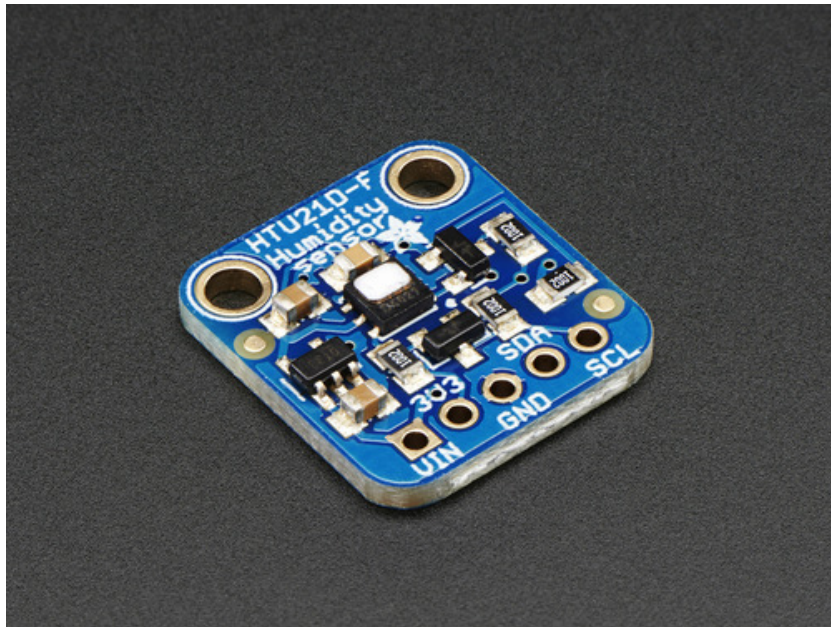




Adafruit HTU21D-F Temperature & Humidity Sensor

Created by lady ada

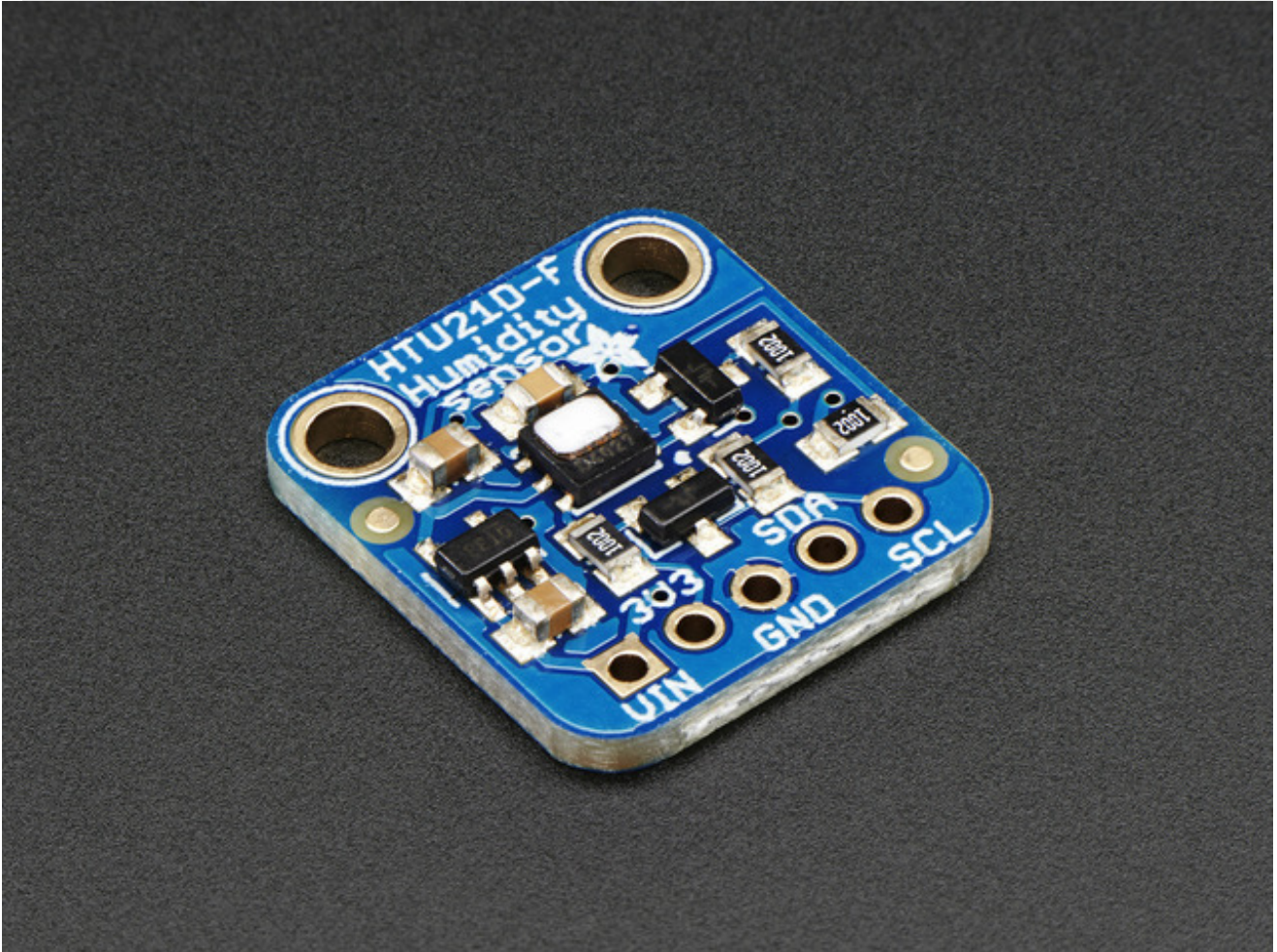


Last updated on 2014-07-26 01:30:08 PM EDT

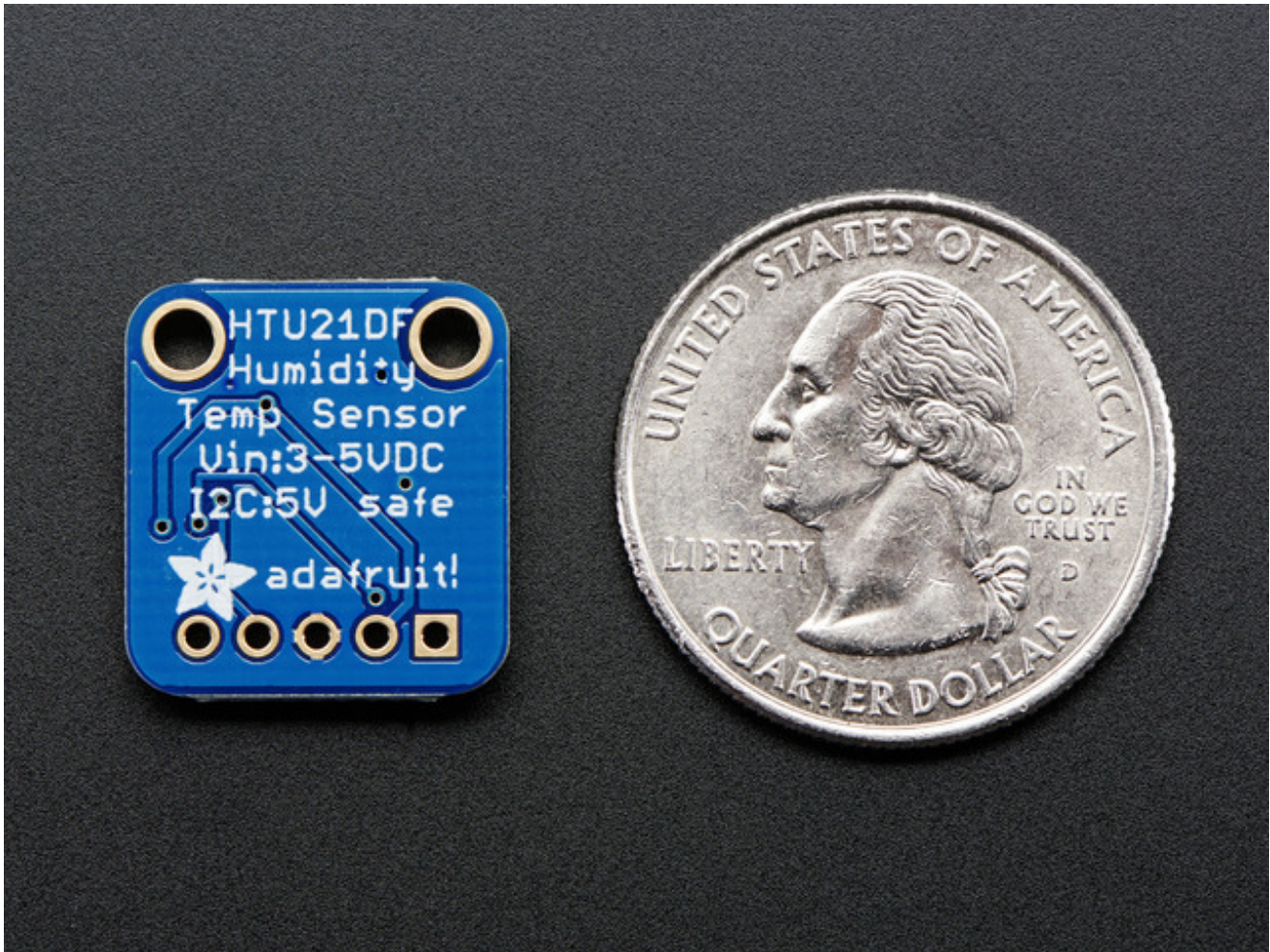
Guide Contents

Guide Contents	2
Overview	3
Pinouts	6
Power Pins:	6
I2C Logic pins:	7
Assembly	8
Prepare the header strip:	8
Add the breakout board:	9
And Solder!	9
Wiring & Test	11
Download Adafruit_HTU21DF	11
Load Demo	12
Library Reference	14
Downloads	16
Schematics	16
PCB Print	16

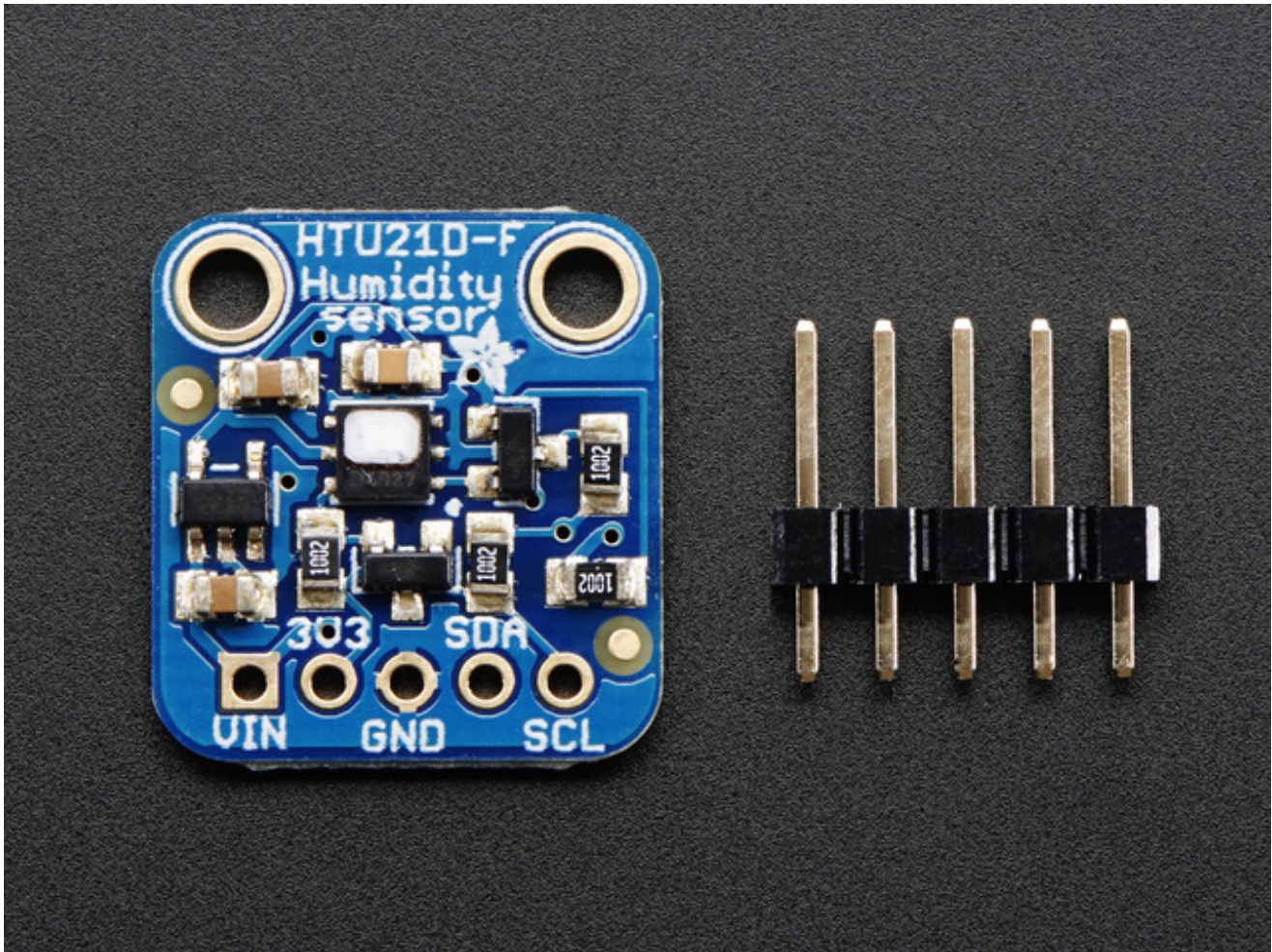
Overview



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 52% when it feels more like 77%. Enter the **HTU21D-F Temperature + Humidity Sensor** - the best way to prove the weatherman wrong!



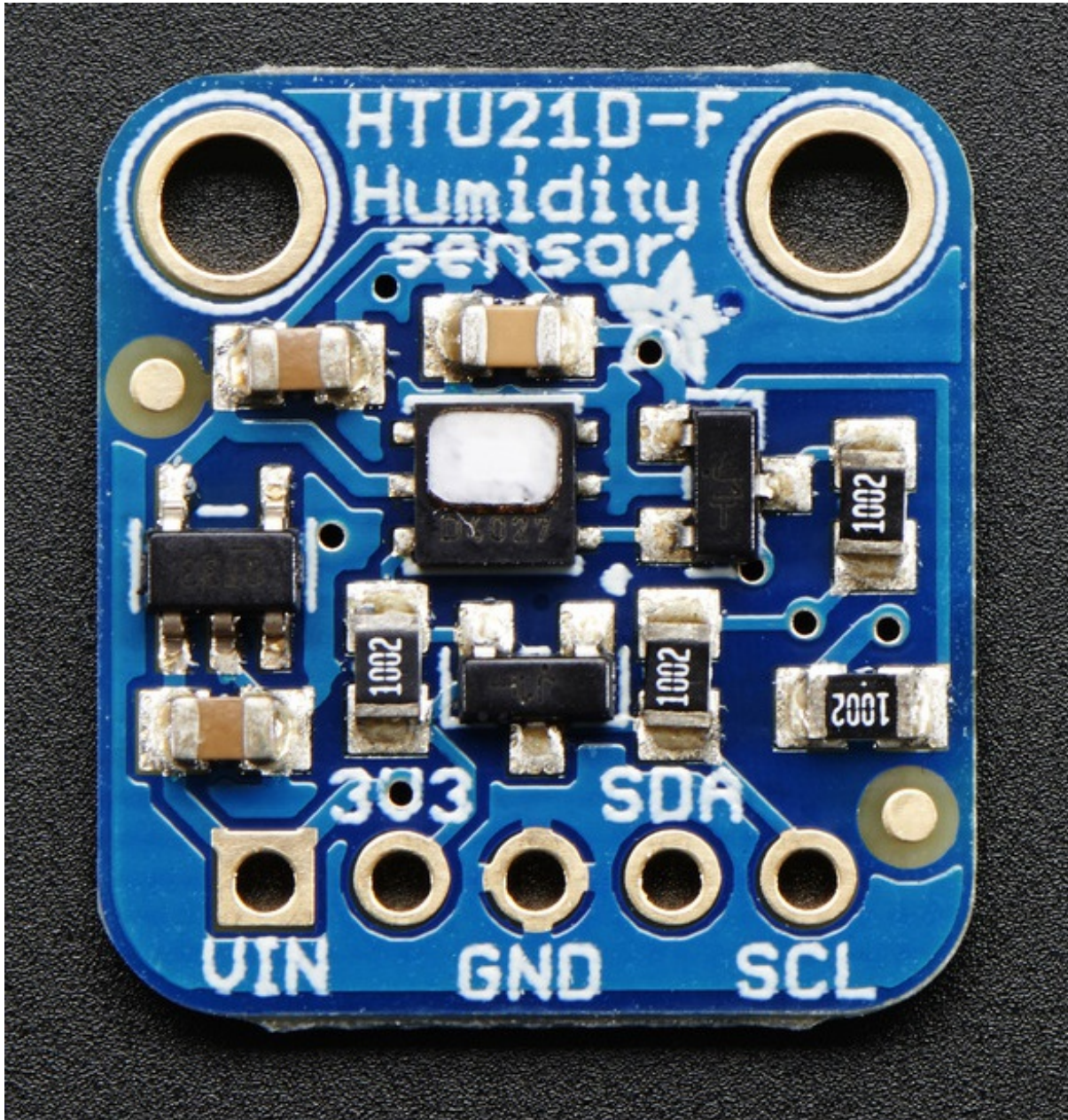
This I2C digital humidity sensor is an accurate and intelligent alternative to the much simpler [Humidity and Temperature Sensor - SHT15 Breakout](http://adafru.it/1638) (<http://adafru.it/1638>) It has a typical accuracy of $\pm 2\%$ with an operating range that's optimized from 5% to 95% RH. Operation outside this range is still possible - just the accuracy might drop a bit. The temperature output has an accuracy of $\pm 1^{\circ}\text{C}$ from $-30\sim 90^{\circ}\text{C}$. If you're looking to measure temperature more accurately, we recommend the [MCP9808 High Accuracy I2C Temperature Sensor Breakout Board](http://adafru.it/1782). (<http://adafru.it/1782>)



Such a lovely chip - so we spun up a breakout board that includes the Filtered version (the white bit of plastic which is a PTFE filter to keep the sensor clean), a 3.3V regulator and I2C level shifting circuitry. This lets you use it safely with any kind of microcontroller with 3.3V-5V power or logic. 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

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 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

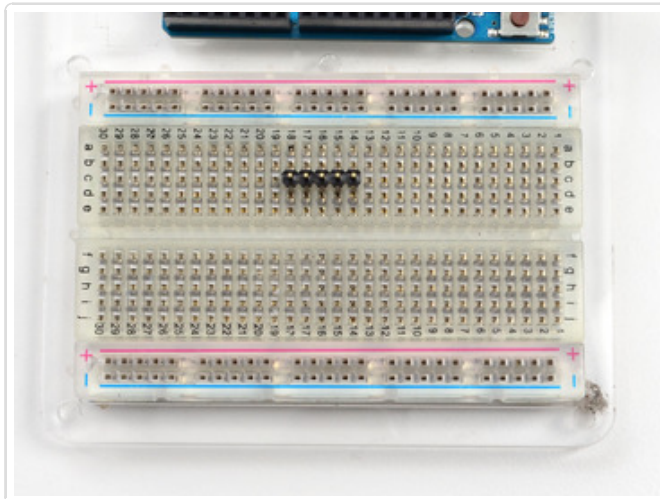
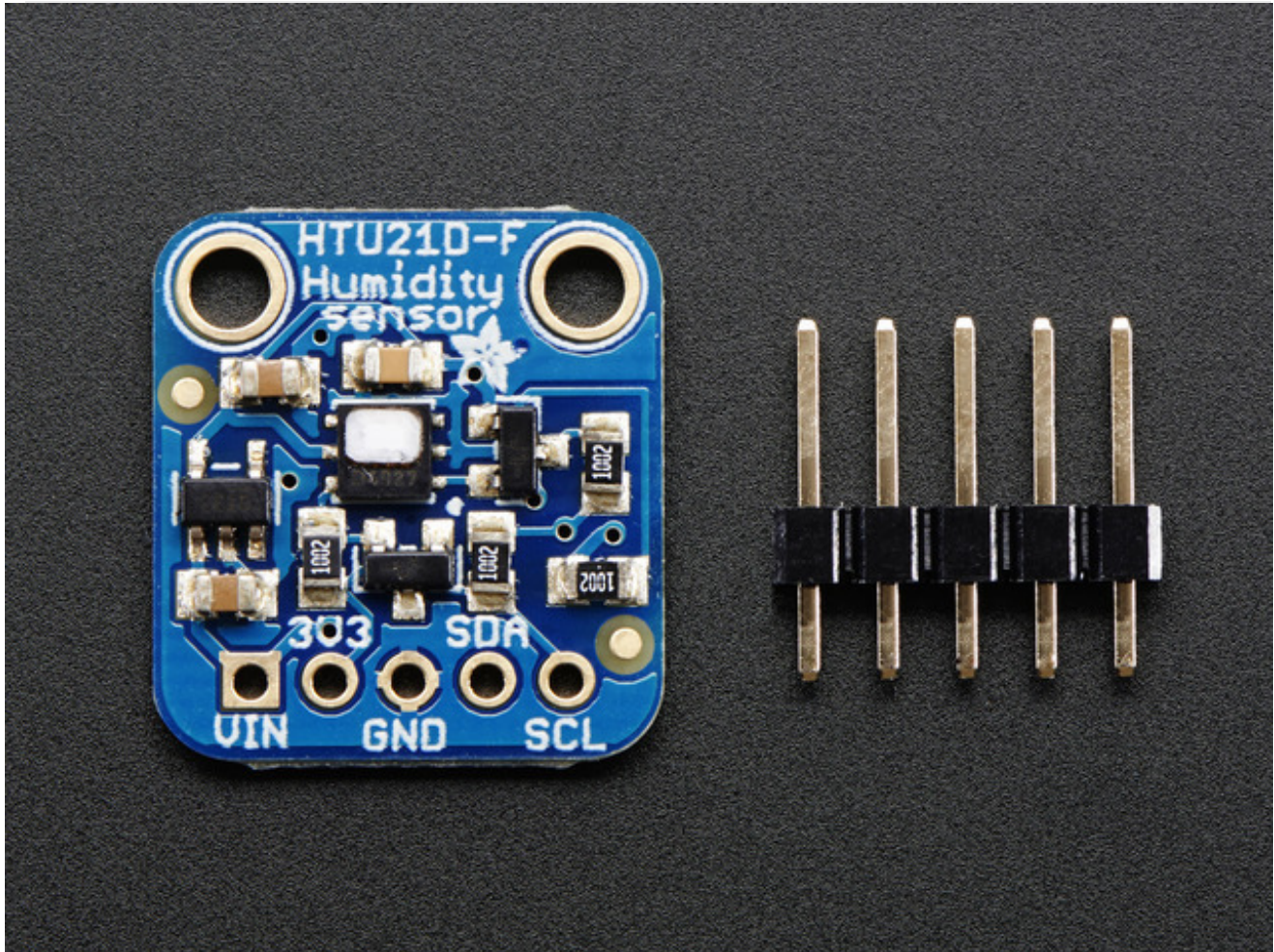
micro like Arduino, use 5V

- **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

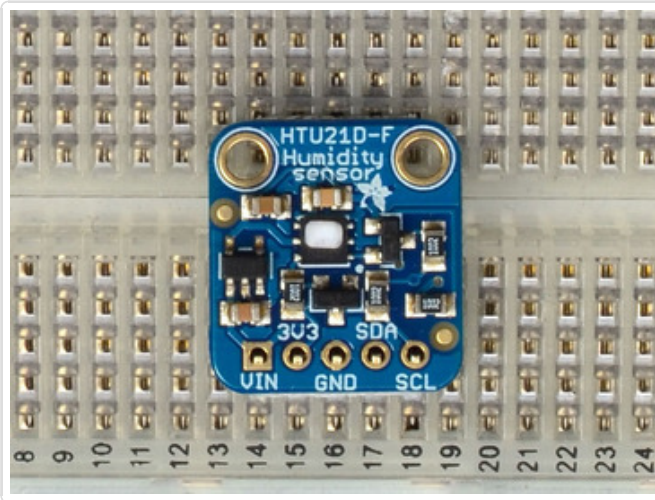
I2C Logic pins:

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

Assembly

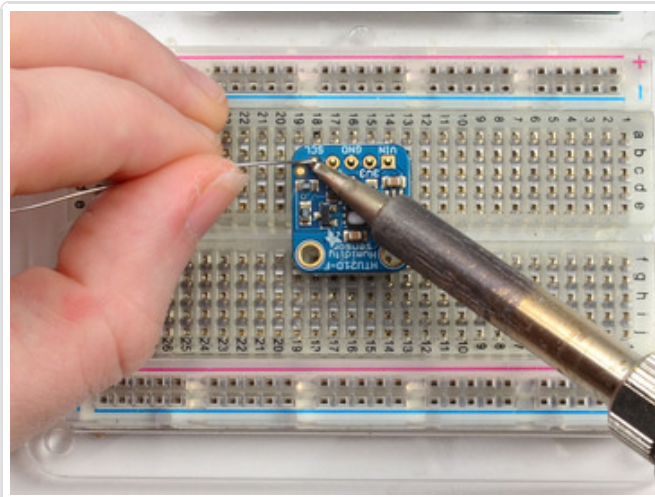


Prepare the header strip:
Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - **long pins down**



Add the breakout board:

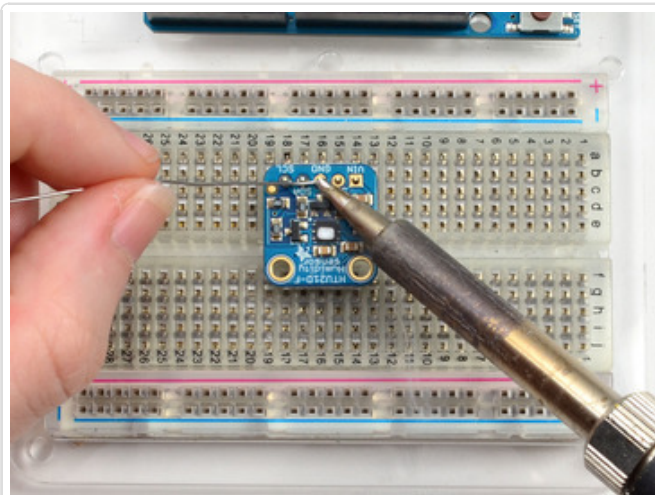
Place the breakout board over the pins so that the short pins poke through the breakout pads

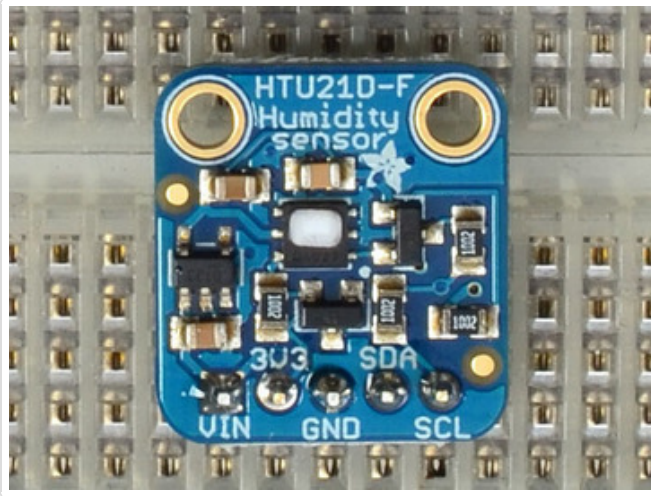


And Solder!

Be sure to solder all pins for reliable electrical contact.

(For tips on soldering, be sure to check out our [Guide to Excellent Soldering](http://adafruit.it/aTk) (<http://adafruit.it/aTk>)).

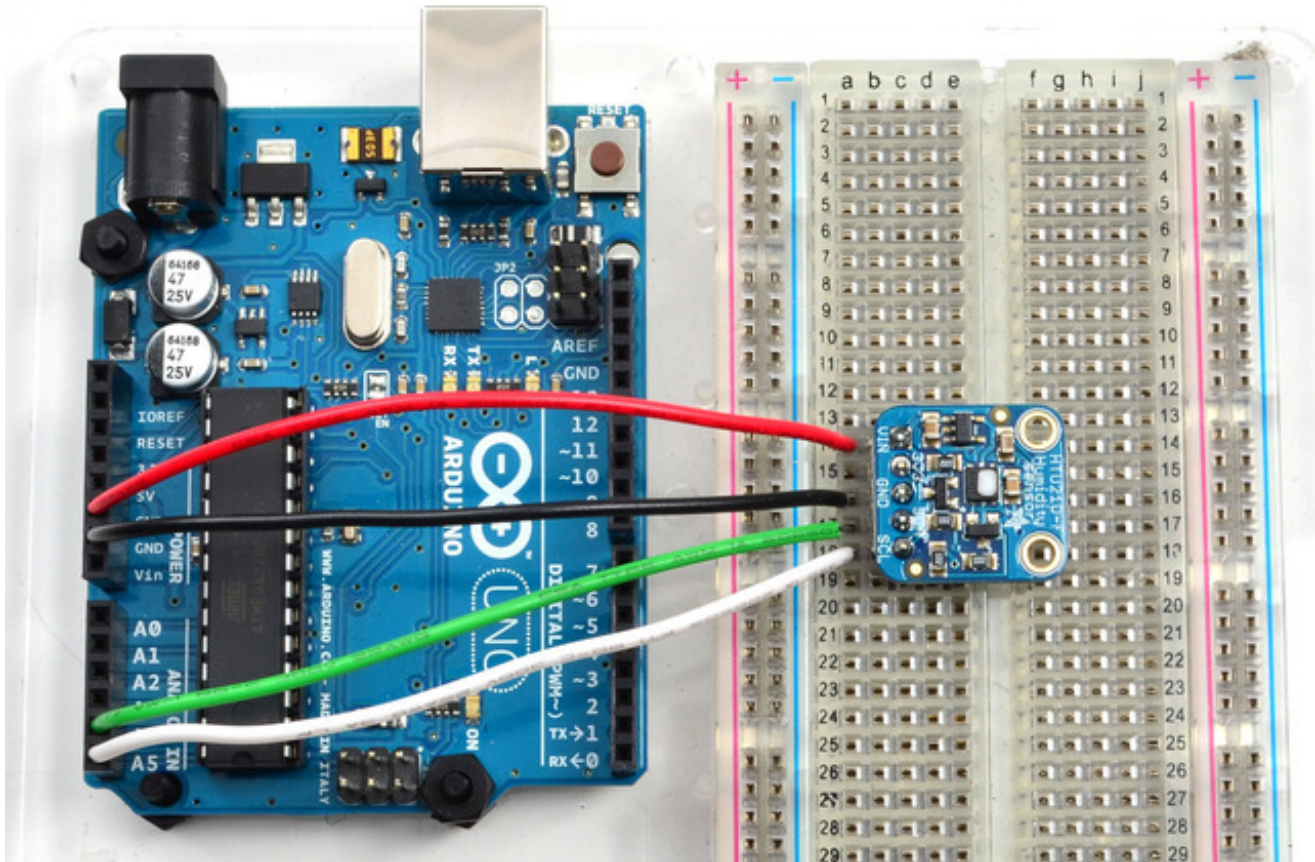




You're done! Check your solder joints visually and continue onto the next steps

Wiring & Test

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 HTU21D-F has a default I2C address of **0x40** and cannot be changed!

Download Adafruit_HTU21DF

To begin reading sensor data, you will need to [download Adafruit_HTU21DF_Library from our github repository \(http://adafru.it/dw5\)](http://adafru.it/dw5). You can do that by visiting the github repo and manually downloading or, easier, just click this button to download the zip

Download Adafruit HTU21D-F Library

<http://adafru.it/dw6>

Rename the uncompressed folder **Adafruit_HTU21DF** and check that the **Adafruit_HTU21DF** folder contains **Adafruit_HTU21DF.cpp** and **Adafruit_HTU21DF.h**

Place the **Adafruit_HTU21DF** library folder your **arduinorsketchfolder/libraries/** folder.

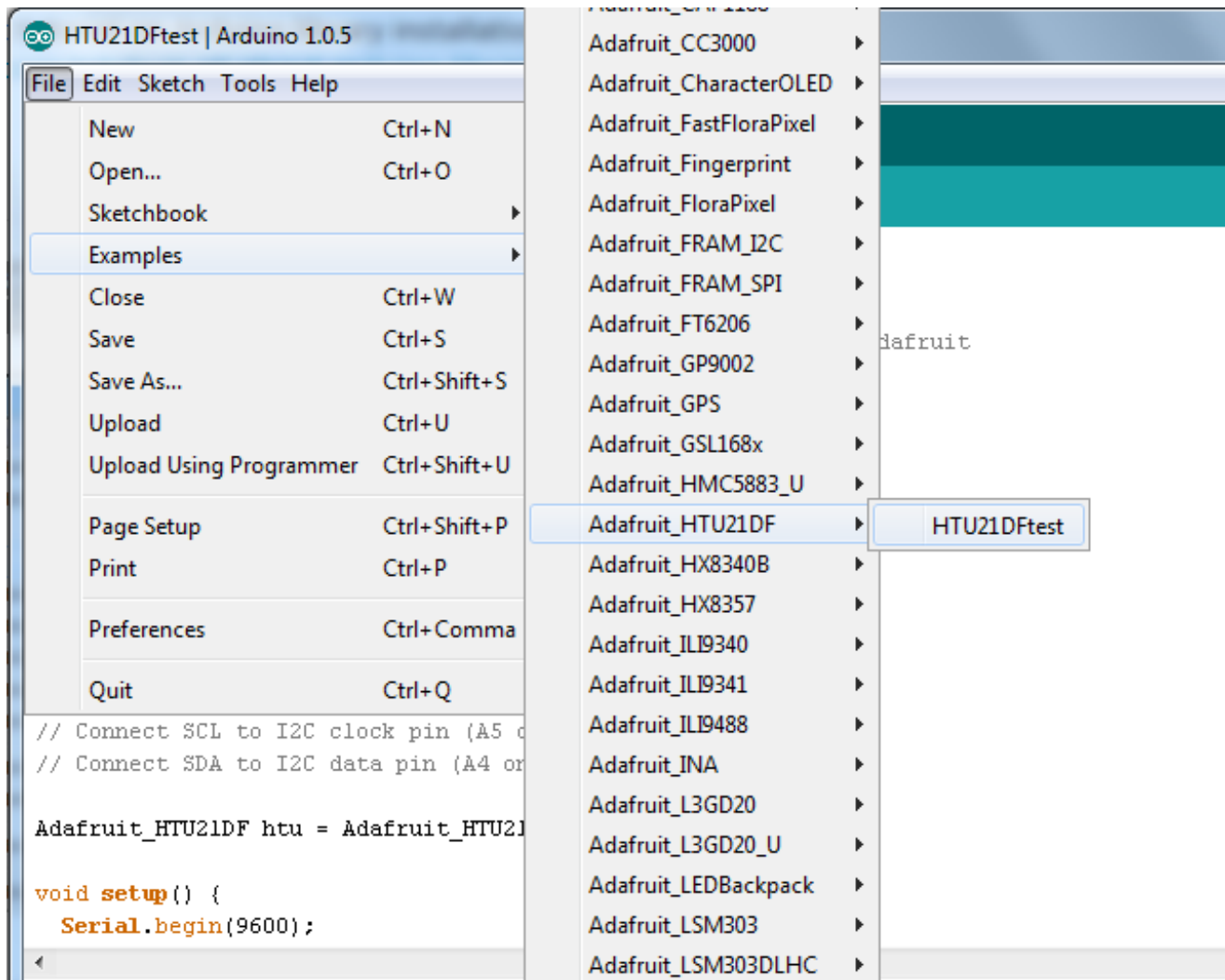
You may need to create the **libraries** subfolder if its your first library. Restart the IDE.

We also have a great tutorial on Arduino library installation at:

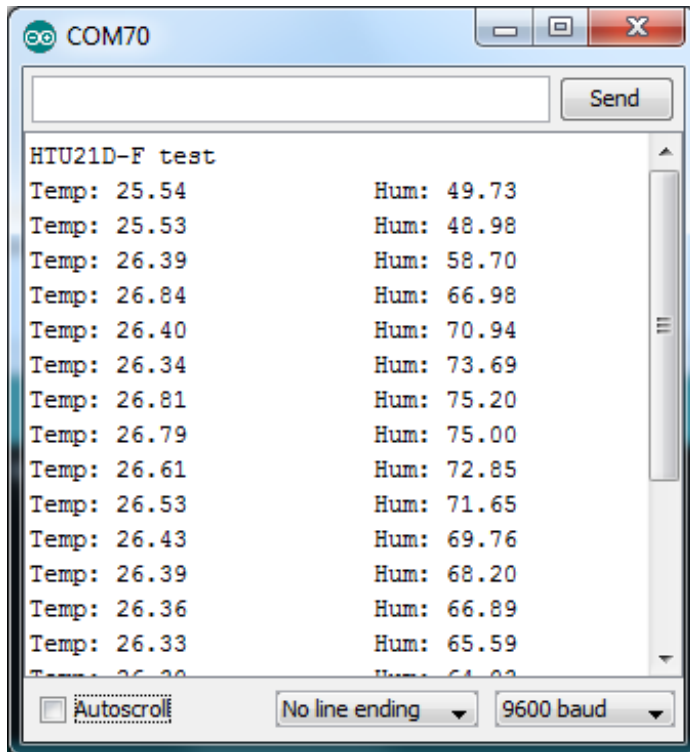
<http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use> (<http://adafru.it/aYM>)

Load Demo

Open up **File->Examples->Adafruit_HTU21DF->HTU21DFtest** 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_HTU21DF** object with:

```
Adafruit_HTU21DF htu = Adafruit_HTU21DF()
```

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

Then initialize the sensor with:

```
htu.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

```
htu.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!

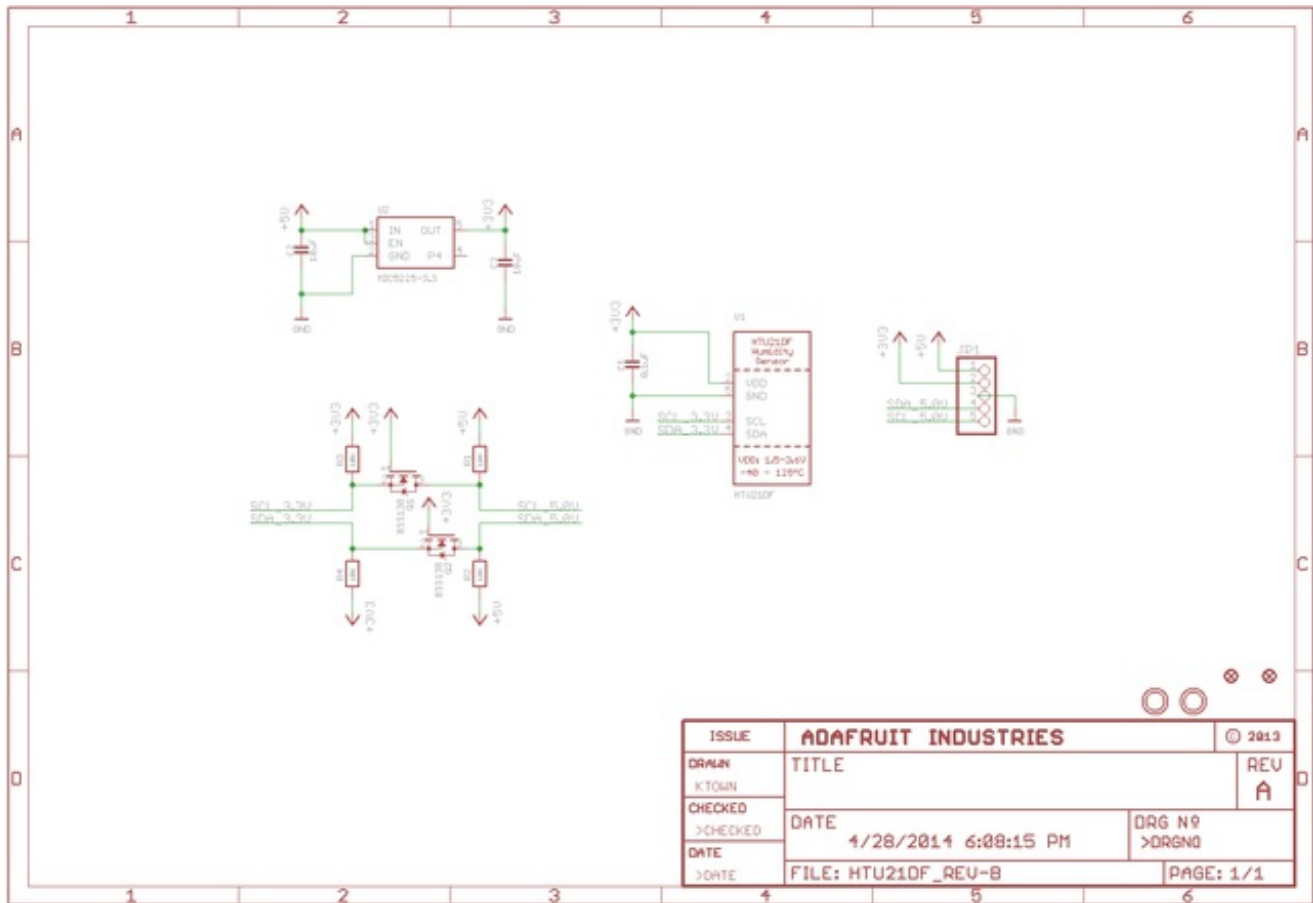
Reading the humidity is equally simple. Call

```
htu.readHumidity()
```

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

Datasheet for the HTU21D-F (<http://adafru.it/dKR>) (the -F part is for the PTFE Filter, which is the white insert on top of the sensor)

Schematics



PCB Print

Dimensions in Inches!

