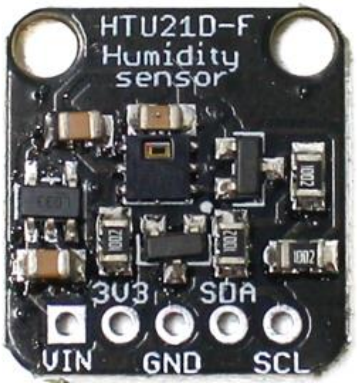
SmartElex HTU21D-F Temperature & Humidity Sensor Breakout Board



This I2C digital humidity sensor is an accurate and intelligent alternative to the much simpler Humidity and Temperature Sensor - SHT15. It has a typical accuracy of ±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 ±1°C from -30~90°C.

We created 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.

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:

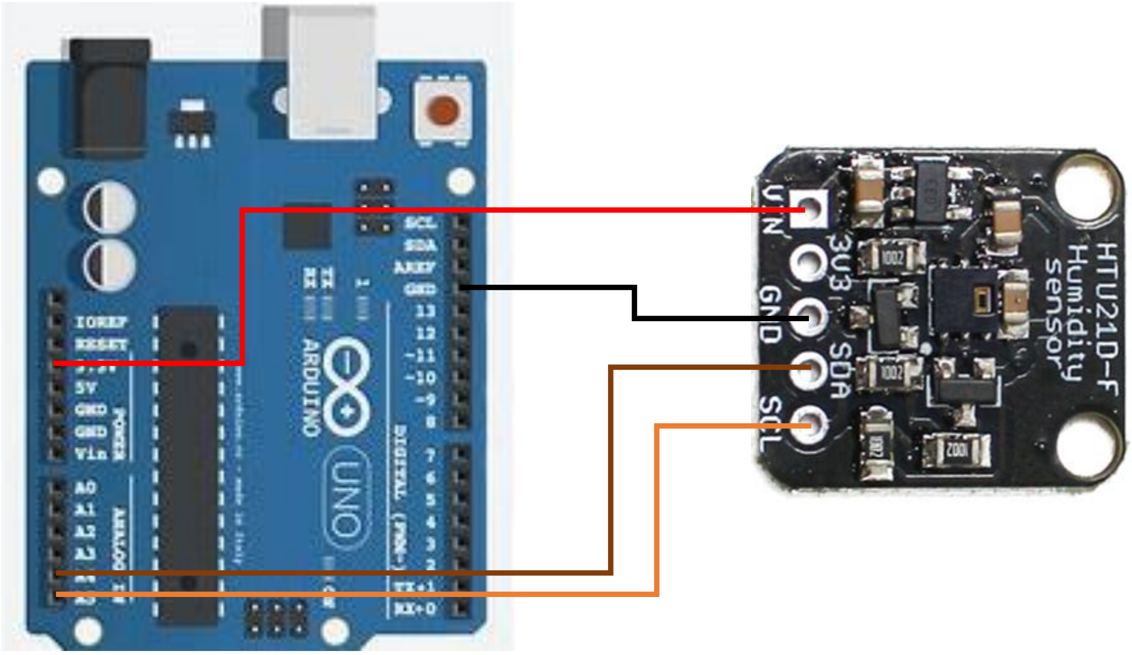
- **Vin** - this is the power pin. Since the chip uses 3 VDC, we have included a voltage regulator on board that will take 3-5VDC and safely convert it down. To power the board, give it the same power as the logic level of your microcontroller - e.g. for a 5V micro like Arduino, use 5V
- **3Vo** - this is the 3.3V output from the voltage regulator, you can grab up to 100mA from this
- **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.

Wiring

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.



.

Arduino	HTU21D-F
SCL(A5)	SCL

SDA(A4)	**SDA**
5v OR 3.3v	**VIN**
GND	**GND**

- Connect **Vin** to the power supply, 3-5V is fine. Use the same voltage that the microcontroller logic is based off. 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!

To begin reading sensor data, you will need to download the **Adafruit HTU21DF** library from the Arduino library manager. Open up the Arduino library manager, Search for the **Adafruit HTU21DF** library and install it.

Load Example

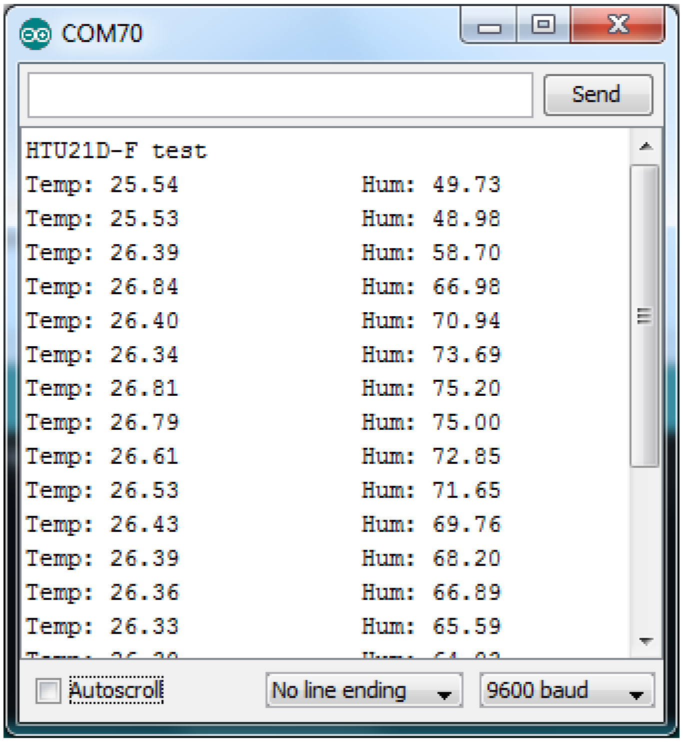
Open up **File->Examples->Adafruit_HTU21DF->HTU21DFtest** and upload to your Arduino wired up to the sensor

Example Code

```
#include <Wire.h>

#include "Adafruit_HTU21DF.h"



// Connect Vin to 3-5VDC

// Connect GND to ground

// Connect SCL to I2C clock pin (A5 on UNO)

// Connect SDA to I2C data pin (A4 on UNO)



Adafruit_HTU21DF htu = Adafruit_HTU21DF();
```

```
void setup() {
  Serial.begin(9600);
  Serial.println("HTU21D-F test");

  if (!htu.begin()) {
    Serial.println("Couldn't find sensor!");
    while (1);
  }
}
void loop() {
    float temp = htu.readTemperature();
    float rel_hum = htu.readHumidity();
    Serial.print("Temp: "); Serial.print(temp); Serial.print(" C");
    Serial.print("\t\t");
    Serial.print("Humidity: "); Serial.print(rel_hum); Serial.println(" \%");
    delay(500);
}
```

Now open up the serial terminal window at 9600 speed to begin the test.



```
HTU21D-F test
Temp: 25.54           Hum: 49.73
Temp: 25.53           Hum: 48.98
Temp: 26.39           Hum: 58.70
Temp: 26.84           Hum: 66.98
Temp: 26.40           Hum: 70.94
Temp: 26.34           Hum: 73.69
Temp: 26.81           Hum: 75.20
Temp: 26.79           Hum: 75.00
Temp: 26.61           Hum: 72.85
Temp: 26.53           Hum: 71.65
Temp: 26.43           Hum: 69.76
Temp: 26.39           Hum: 68.20
Temp: 26.36           Hum: 66.89
Temp: 26.33           Hum: 65.59
```

☐ Autoscroll   No line ending   9600 baud

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
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)