


# SHT15 Humidity and Temperature Sensor Hookup Guide

CONTRIBUTORS:  JOEL\_E\_B

♥ FAVORITE 2

## SHT15 Overview

The SHT15 is an easy to use, highly accurate, digital temperature and humidity sensor. All you need is two lines for 2-wire communication, and you'll have relative humidity and temperature readings to help you sense the world around you!



Features you should know about this sensor:

- Operating Voltage: 2.4V min - 5.5V max
- 2 factory calibrated sensors for relative humidity & temperature
- Digital 2-wire interface (Not I<sup>2</sup>C, but similar)
- Measurement range: 0-100% RH
- Absolute RH accuracy: +/- 2% RH (10...90% RH)
- Repeatability RH: +/- 0.1% RH
- Temp. accuracy: +/- 0.3°C @ 25°C
- Precise dewpoint calculation possible
- Fast response time
- Low power consumption (typ. 30 μW)
- Here's the datasheet

This sensor is ideal for environmental sensing and data logging and can be used in applications ranging from a weather station to a humidior control system.

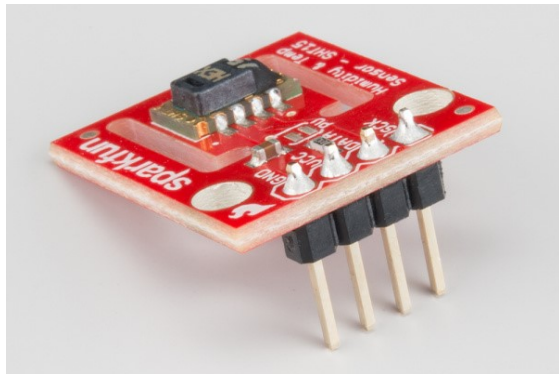
## Suggested Reading

Things you might need to know:

- What are pull-up resistors?
- How to use a Breadboard
- How to Solder
- Serial Terminal Basics

## Hooking It Up

Wiring up the SHT15 is very easy! We recommend soldering four male headers to the breakout board. You can then attach it to an Arduino or Arduino-Compatible Board, such as our RedBoard.

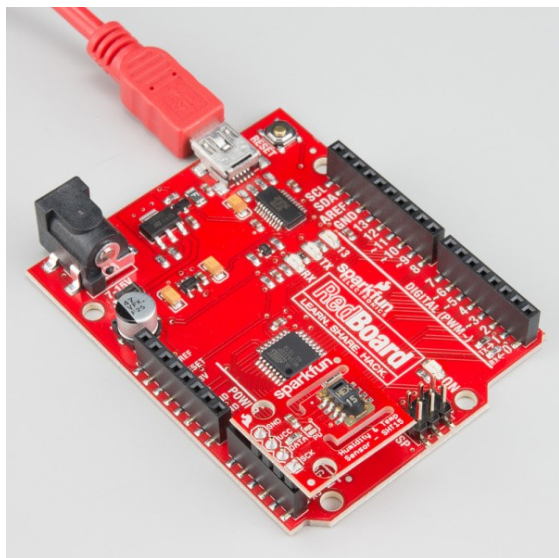


### Connections: Breakout board to Arduino

There are only four pins that need to be hooked up in order to start using this sensor in a project. One for VCC, one for GND, and two data lines.

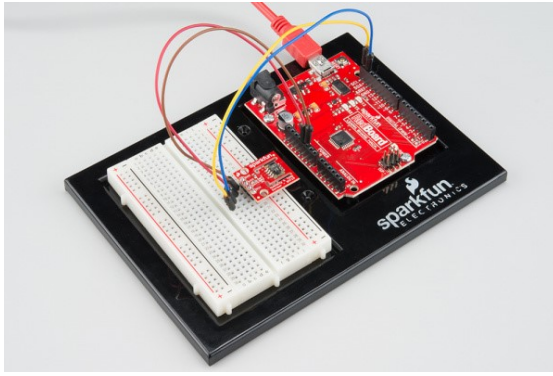
- VCC → 3.3V or 5V
- GND → GND
- Data → Any I/O Pin
- SCK → Any I/O Pin

You can connect this sensor directly to the female headers on your Arduino like so...



*A SHT15 connected to a SparkFun RedBoard, using pins A4 and A5 as communication and pins A2 and A3 as power.*

Or, you can wire it up on a breadboard.



A SHT15 connected to the SDA(A4) and SCL(A5) lines on a SparkFun RedBoard.

## 2-Wire Interface

Please note that the SHT15 has a 2-wire interface that is similar to I<sup>2</sup>C but is **NOT I<sup>2</sup>C**. You may use the SDA and SCL lines to communicate with this sensor so long as they are connected to A4 and A5 on your Arduino or Arduino compatible board. You may use any other I/O pins for the Data and SCK lines as well.

**Heads up!** The datasheet states: “The sensor **cannot** be addressed by I<sup>2</sup>C protocol; however, the sensor can be connected to an I<sup>2</sup>C bus without interference with other devices connected to the bus. The controller must switch between the protocols.” We have experimented with this with mixed results. Make sure you test extensively if adding this to an I<sup>2</sup>C bus containing other sensors. Should bus contention be a problem, simply use two other I/O pins to communicate with the SHT15.

## Multiple Sensors

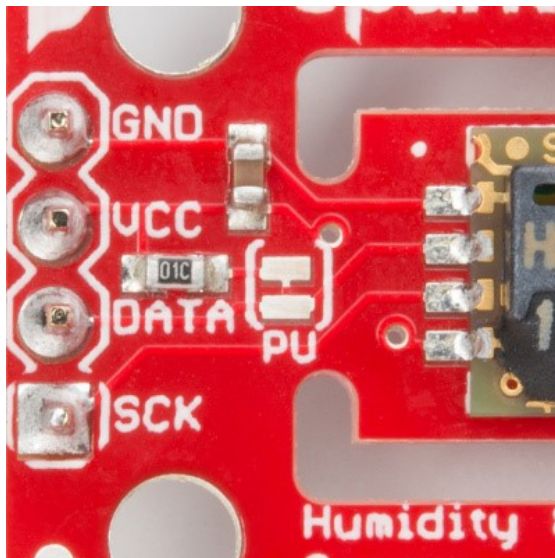
Unfortunately, the SHT15 has its address hard-coded, making it non-addressable. As a result, you may not have more than one SHT15 hooked up to any two pins at a given time. The 2-wire protocol used for this sensor does not allow for multiple of the same sensor to be on the bus, but it does allow multiple I<sup>2</sup>C devices to share the same bus as the SHT15, as mentioned above. In order to use more than one SHT15, you will need to use a dedicated pin for each sensor’s Data line. You may share the Clock (SCK) line between multiple sensors. For example, you could have one sensor’s Data line connected to pin 8 and another connected to pin 7, while both are sharing pin 9 as the SCK line. Declaring this using the library mentioned in the next section would look like this:

```
//Create two instances of the SHT1X sensor
SHT1x sht15(8, 9);//Data, SCK
SHT1x sht15_2(7, 9);//Data, SCK
```

## Pull-up Resistor

As suggested in the datasheet, a 10KΩ pull-up resistor was added to the Data line. Should you find yourself in a situation where you want this sensor and another I<sup>2</sup>C sensor with pull-up resistors to share the I<sup>2</sup>C lines, you may cut the trace in between the solder jumper labeled **PU**. If you ever need

that pull-up again, simply place a blob of solder between the two pads.



## SHT1X Arduino Library and Example

Luckily, there is an Arduino library and some example code to make using the SHT15 as easy as possible. Grab the SHT15 library for Arduino from the SHT15 GitHub Repository , or you can download the files directly from the button below. Alternative library options can be found here.

**SHT15 ARDUINO LIBRARY**

Need help using the Library Manager or want to install the library the old fashioned way? Visit our Arduino Library tutorial, for more information.

Alternatively, you can use the Sensirion Library.

Once the library is installed, open Arduino, and expand the examples menu. You should see the SHT\_1X submenu.



Load this example onto the Arduino. Open the serial terminal at 9600bps. You will see the current humidity and temperature in the room!

Here is the same example sketch if you would rather copy and paste:

```

/*****
*****
SHT15 Example
Joel Bartlett @ SparkFun Electronics
16 Sept 2015
https://github.com/sparkfun/SparkFun_ISL29125_Breakout_Arduino_Library

This example shows how to get the temperature in F or C and humidity
Developed/Tested with:
SparkFun RedBoard
Arduino IDE 1.6.5

Connections:
GND -> A2
Vcc -> A3
DATA -> A4
SCK -> A5

Requires:
SHT1X Arduino Library
https://github.com/sparkfun/SHT15_Breakout/

This code is beerware.
Distributed as-is; no warranty is given.
*****/
#include <SHT1X.h>

//variables for storing values
float tempC = 0;
float tempF = 0;
float humidity = 0;

//Create an instance of the SHT1X sensor
SHT1x sht15(A4, A5); //Data, SCK

//declare output pins for powering the sensor
int power = A3;
int gnd = A2;

void setup()
{
  Serial.begin(9600); // Open serial connection to report values to host

  pinMode(power, OUTPUT);
  pinMode(gnd, OUTPUT);

  digitalWrite(power, HIGH);
  digitalWrite(gnd, LOW);
}
//-----
void loop()
{
  readSensor();
}

```

```
    printOut();
    delay(1000);
}
//-----
-----
void readSensor()
{
    // Read values from the sensor
    tempC = sht15.readTemperatureC();
    tempF = sht15.readTemperatureF();
    humidity = sht15.readHumidity();
}
//-----
-----
void printOut()
{
    Serial.print(" Temp = ");
    Serial.print(tempF);
    Serial.print("F, ");
    Serial.print(tempC);
    Serial.println("C");
    Serial.print(" Humidity = ");
    Serial.print(humidity);
    Serial.println("%");
}
}
```

## Resources and Going Further

You should now have a good idea of how to add humidity and temperature sensing into your next project. Need some inspiration? Check out these other tutorials:

- [Make an automated terrarium that manages heat and humidity levels.](#)
- [Build a humidior control box that maintains a constant humidity in a controlled space.](#)

### Resources:

- [SHT15 Datasheet](#)
- [HTU21D Breakout Board Schematic](#)
- [HTU21D Breakout Board Eagle Files](#)
- [Github repo containing all the latest files and code](#)