

search blog...

- [Home](#)
- [Forum](#)
- [Wiki](#)
- [Code](#)
  - [RSS](#)
  - [Twitter](#)
  - [Facebook](#)

[Search Wiki](#) [Search Forum](#) [Search Blog](#)

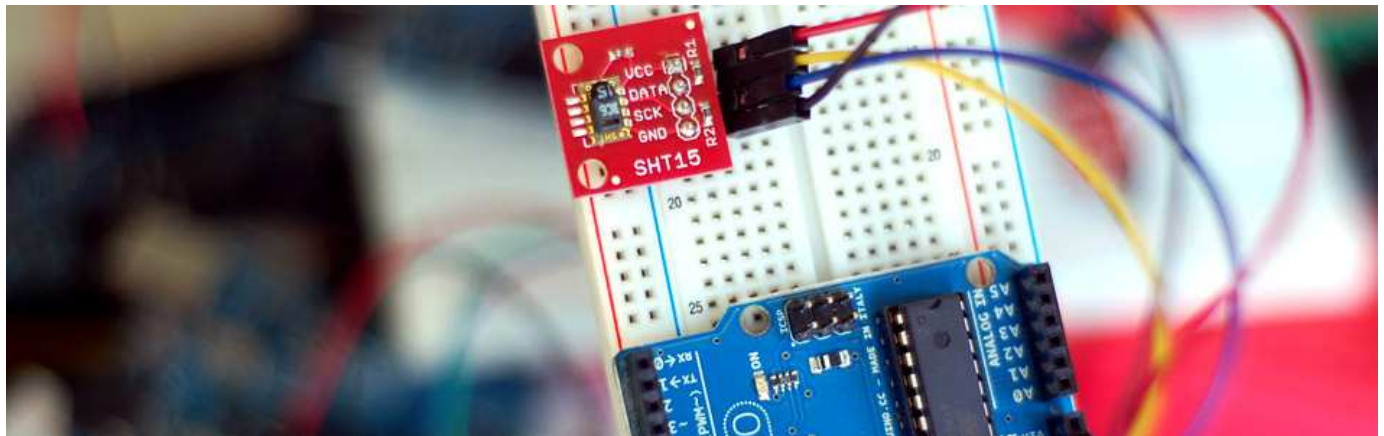
[Login](#) / [Register](#)

- [About](#)

**PLC**

## Sensing Humidity With The SHT15 + Arduino

Tuesday, November 27<sup>th</sup>, 2012



The SHT15 is a digital humidity sensor that outputs a fully calibrated humidity reading. And... because what we are measuring is actually [relative humidity](#), and relative humidity being relative to temperature, the SHT15 has a builtin digital thermometer. This makes things much easier to work with [than sensors without a thermometer onboard](#). You can [pick one up from sparkFun here](#).

### Hooking it up

The SHT15 uses a two-wire connection for communication that is similar to, but not, I2C. So we won't be able to use the Arduino's dedicated lines for this. The downside is it is a bit slower to get readings from, the plus side is that you can connect it to any 2 digital pins you want. We are using pins 2 and 3 on our arduino.

**BEFORE YOU SOLDER IT UP...** Note that this board can not be washed! So if you are using flux, or solder that you normally clean up, don't (They actually recommend to use "no-clean" solder just so you don't have to worry about it). And be extremely careful not to get it wet at all.

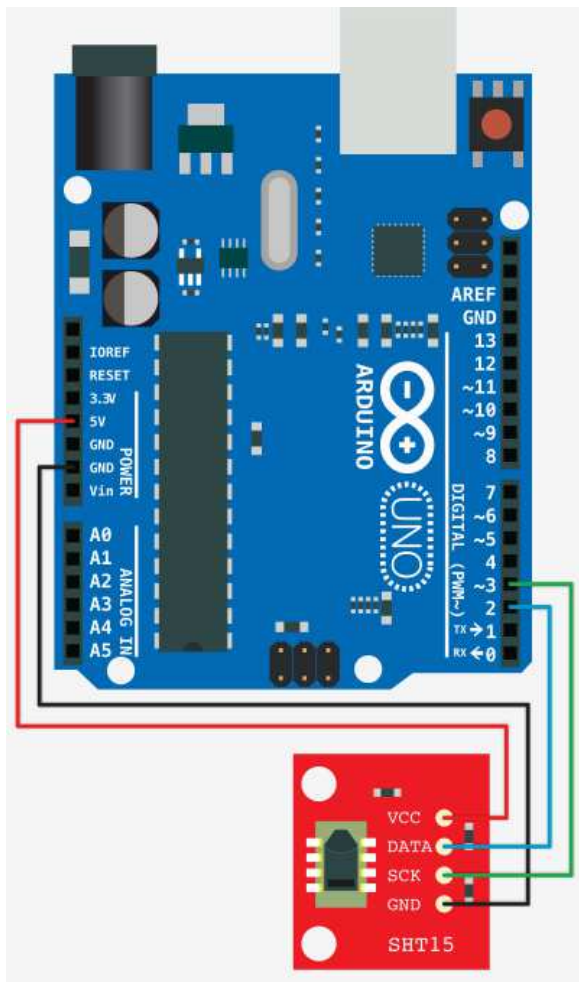
**AFTER YOU SOLDER IT UP...** To get a clean reading, the sensor needs to be stored at >75% humidity for at least 12 hours to allow the polymer to re-hydrate (just what the doc says). If you don't, your SHT15 may read an offset that slowly disappears if exposed to ambient conditions. Alternatively the re-hydration process may be performed at ambient conditions (>40% Humidity) for 5+ days.

I'm not exactly sure how you do that... But someone noted that they put it in a ziplock with a wet towel (not touching) for 12H.

### Code

The code for this is a bit wacky (as with most digital sensors), but it is split up pretty nicely, and is as easy to read as it can be.

**Note that the readings are a bit slow to return a value (100+ ms).**



### [Copy Code](#)

//Based of the wiring code at <http://wiring.org.co/learning/basics/humiditytemperaturesht15.html>

```
int SHT_clockPin = 3; // pin used for clock
int SHT_dataPin = 2; // pin used for data

void setup(){
  Serial.begin(9600); // open serial at 9600 bps
}

void loop(){
  //these can take a bit to get the values (100ms or so)
  float temperature = getTemperature();
  float humidity = getHumidity();

  Serial.print(temperature);
  Serial.print(" | ");
  Serial.println(humidity);
}

float getTemperature(){
  //Return Temperature in Celsius
  SHT_sendCommand(B00000011, SHT_dataPin, SHT_clockPin);
  SHT_waitForResult(SHT_dataPin);

  int val = SHT_getData(SHT_dataPin, SHT_clockPin);
  SHT_skipCrc(SHT_dataPin, SHT_clockPin);
  return (float)val * 0.01 - 40; //convert to celsius
}

float getHumidity(){
  //Return Relative Humidity
  SHT_sendCommand(B00000101, SHT_dataPin, SHT_clockPin);
  SHT_waitForResult(SHT_dataPin);
  int val = SHT_getData(SHT_dataPin, SHT_clockPin);
  SHT_skipCrc(SHT_dataPin, SHT_clockPin);
  return -4.0 + 0.0405 * val + -0.0000028 * val * val;
}
```

```

void SHT_sendCommand(int command, int dataPin, int clockPin){
  // send a command to the SHTx sensor
  // transmission start
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  digitalWrite(dataPin, HIGH);
  digitalWrite(clockPin, HIGH);
  digitalWrite(dataPin, LOW);
  digitalWrite(clockPin, LOW);
  digitalWrite(clockPin, HIGH);
  digitalWrite(dataPin, HIGH);
  digitalWrite(clockPin, LOW);

  // shift out the command (the 3 MSB are address and must be 000, the last 5 bits are the command)
  shiftOut(dataPin, clockPin, MSBFIRST, command);

  // verify we get the right ACK
  digitalWrite(clockPin, HIGH);
  pinMode(dataPin, INPUT);

  if (digitalRead(dataPin)) Serial.println("ACK error 0");
  digitalWrite(clockPin, LOW);
  if (!digitalRead(dataPin)) Serial.println("ACK error 1");
}

void SHT_waitForResult(int dataPin){
  // wait for the SHTx answer
  pinMode(dataPin, INPUT);

  int ack; //acknowledgement

  //need to wait up to 2 seconds for the value
  for (int i = 0; i < 1000; ++i){
    delay(2);
    ack = digitalRead(dataPin);
    if (ack == LOW) break;
  }

  if (ack == HIGH) Serial.println("ACK error 2");
}

int SHT_getData(int dataPin, int clockPin){
  // get data from the SHTx sensor

  // get the MSB (most significant bits)
  pinMode(dataPin, INPUT);
  pinMode(clockPin, OUTPUT);
  byte MSB = shiftIn(dataPin, clockPin, MSBFIRST);

  // send the required ACK
  pinMode(dataPin, OUTPUT);
  digitalWrite(dataPin, HIGH);
  digitalWrite(dataPin, LOW);
  digitalWrite(clockPin, HIGH);
  digitalWrite(clockPin, LOW);

  // get the LSB (less significant bits)
  pinMode(dataPin, INPUT);
  byte LSB = shiftIn(dataPin, clockPin, MSBFIRST);
  return ((MSB << 8) | LSB); //combine bits
}

void SHT_skipCrc(int dataPin, int clockPin){
  // skip CRC data from the SHTx sensor
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  digitalWrite(dataPin, HIGH);
  digitalWrite(clockPin, HIGH);
  digitalWrite(clockPin, LOW);
}

```

Unless otherwise stated, this code is released under the [MIT License](#) – Please use, change and share it.

Tweet

Me gusta 29 [No Comments](#)

## Archives

- [October 2013](#)
- [November 2012](#)
- [August 2012](#)
- [April 2012](#)
- [March 2012](#)
- [February 2012](#)
- [January 2012](#)
- [October 2011](#)
- [September 2011](#)

- [August 2011](#)
- [July 2011](#)
- [June 2011](#)
- [May 2011](#)
- [April 2011](#)
- [March 2011](#)
- [February 2011](#)
- [January 2011](#)
- [December 2010](#)
- [November 2010](#)
- [October 2010](#)
- [September 2010](#)

## Categories

- [funded projects](#)
- [Promotions](#)
- [Review](#)
- [Site](#)
- [Tutorials](#)
- [User Contributed Articles](#)
- [User Project](#)

## about

- Content available under:  
[Attribution-Share Alike 3.0 Unported](#)
- Proudly powered by [WordPress](#), [Free WordPress Themes](#), and [Search Marketing](#)