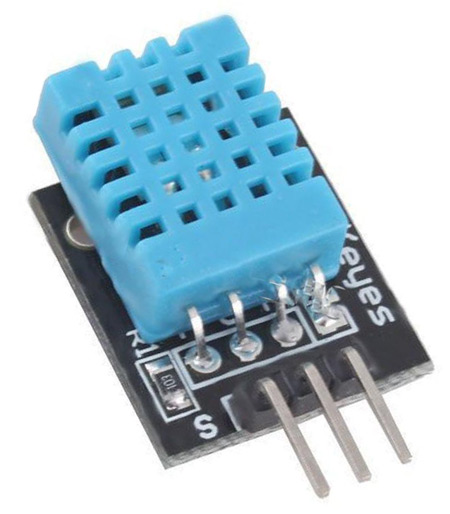
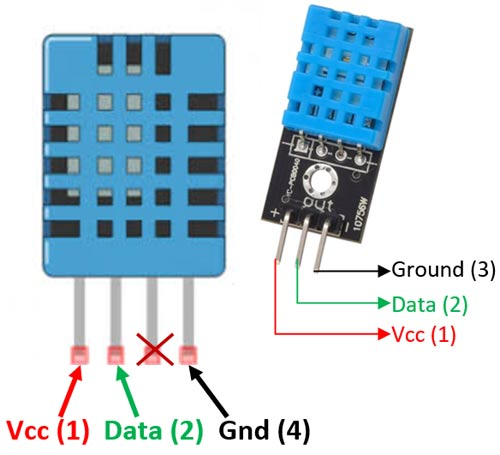
# DHT11–Temperature and Humidity Sensor

**DHT11–Temperature and Humidity Sensor DHT11 Sensor Pinout**

### Pin Identification and Configuration:

|  |  |  |
| --- | --- | --- |
| **No:** | **Pin Name** | **Description** |
| **For DHT11 Sensor** |  |  |
| 1 | Vcc | Power supply 3.5V to 5.5V |
| 2 | Data | Outputs both Temperature and Humidity through serial Data |
| 3 | NC | No Connection and hence not used |
| 4 | Ground | Connected to the ground of the circuit |
| **For DHT11 Sensor module** |  |  |
| 1 | Vcc | Power supply 3.5V to 5.5V |
| 2 | Data | Outputs both Temperature and Humidity through serial Data |
| 3 | Ground | Connected to the ground of the circuit |

You can buy [DHT11 sensor module](https://quartzcomponents.com/products/dht11-temperature-humidity-sensor-module) from here.

### DHT11 Specifications:

* Operating Voltage: 3.5V to 5.5V
* Operating current: 0.3mA (measuring) 60uA (standby)
* Output: Serial data
* Temperature Range: 0°C to 50°C
* Humidity Range: 20% to 90%
* Resolution: Temperature and Humidity both are 16-bit
* Accuracy: ±1°C and ±1%

### DHT11 Equivalent Temperature Sensors:

DHT22, AM2302, SHT71

### Other Temperature Sensors:

Thermocouple, TMP100, LM75, DS18820, SHT15, [LM35DZ](https://components101.com/lm35-temperature-sensor), TPA81, D6T

### **DHT11 Sensor**

The DHT11 is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed). It's fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, the sensor readings can be up to 2 seconds old.

[Compared to the DHT22](http://www.adafruit.com/product/385), this sensor is less precise, less accurate, and works in a smaller range of temperature/humidity, but it is smaller and less expensive.

*Technical Details*

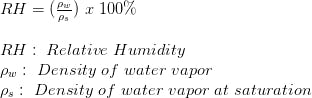
* Low cost
* 3 to 5V power and I/O
* 2.5mA max current use during conversion (while requesting data)
* Good for 20-80% humidity readings with 5% accuracy
* Good for 0-50°C temperature readings ±2°C accuracy
* No more than 1 Hz sampling rate (once every second)
* Body size 15.5mm x 12mm x 5.5mm
* 4 pins with 0.1" spacing

### **What is Relative Humidity?**

The DHT11 measures *relative humidity*. The relative humidity is the amount of water vapor in air vs. the saturation point of water vapor in the air. At the saturation point, water vapor starts to condense and accumulate on surfaces forming dew.

The saturation point changes with air temperature. Cold air can hold less water vapor before it becomes saturated, and hot air can hold more water vapor before it becomes saturated.

The formula to calculate relative humidity is:



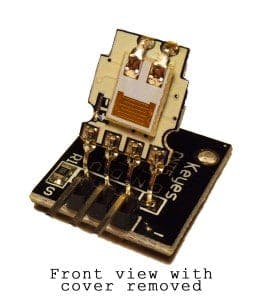
The relative humidity is expressed as a percentage. At 100% RH, condensation occurs, and at 0% RH, the air is completely dry.

### **How the DHT11 Measures Humidity and Temperature**

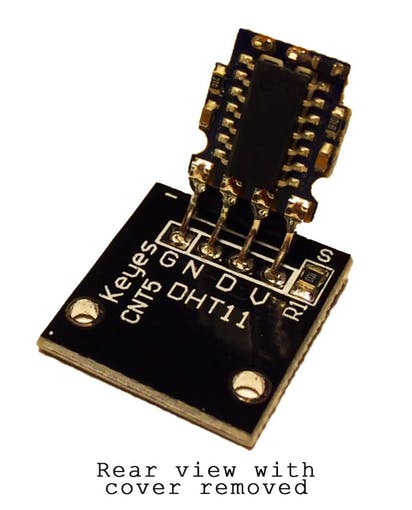
The DHT11 detects water vapor by measuring the electrical resistance between two electrodes. The humidity sensing component is a moisture holding substrate with electrodes applied to the surface. When water vapor is absorbed by the substrate, ions are released by the substrate which increases the conductivity between the electrodes. The change in resistance between the two electrodes is proportional to the relative humidity. Higher relative humidity decreases the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes.

The DHT11 measures temperature with a surface mounted [NTC temperature sensor](http://www.amazon.com/gp/product/B00GD471PO/ref=as_li_qf_sp_asin_il_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=B00GD471PO&linkCode=as2&tag=circbasi-20&linkId=UJTHZ5Z3JDMKGOCK) (thermistor) built into the unit. To learn more about how thermistors work and how to use them on the Arduino, check out this [Arduino Thermistor Temperature Sensor Tutorial.](https://www.circuitbasics.com/arduino-thermistor-temperature-sensor-tutorial)

With the plastic housing removed, you can see the electrodes applied to the substrate:

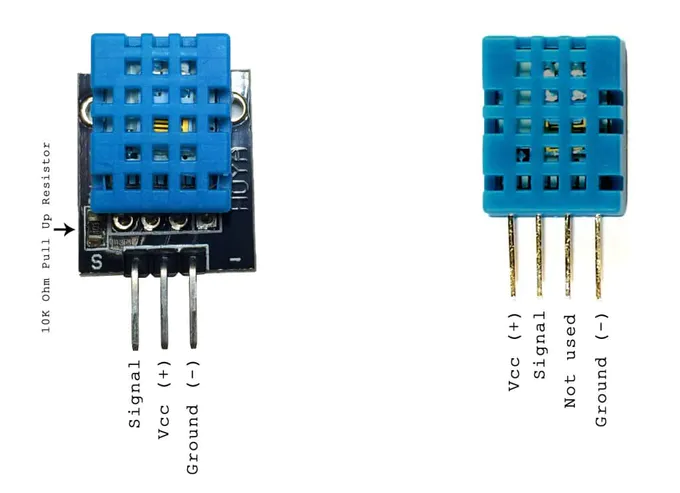


An IC mounted on the back of the unit converts the resistance measurement to relative humidity. It also stores the calibration coefficients, and controls the data signal transmission between the DHT11 and the Arduino:



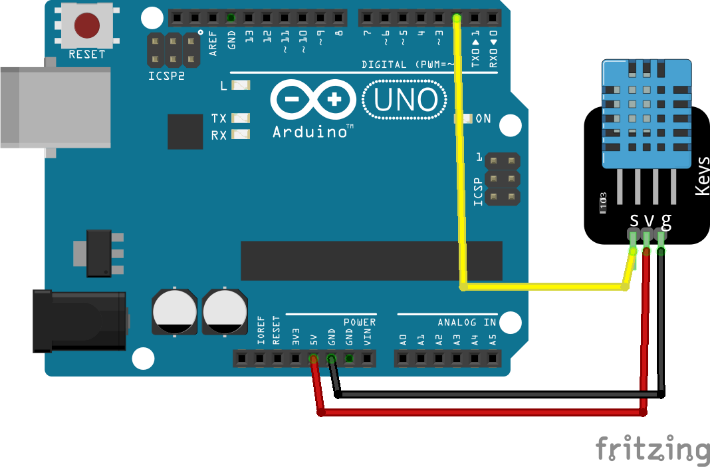
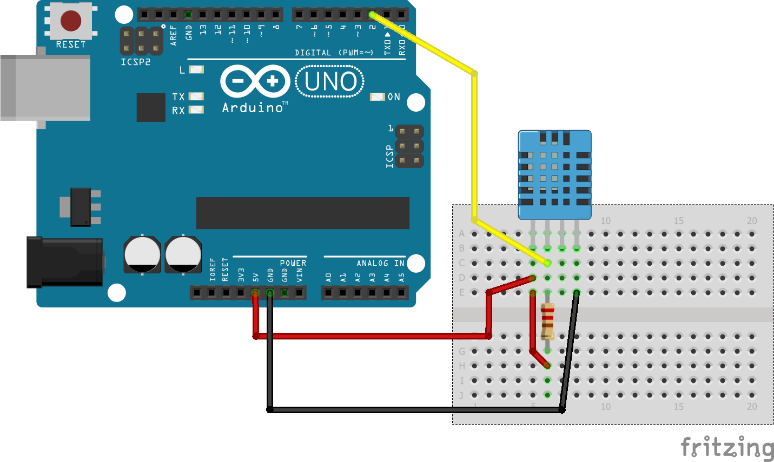
The DHT11 uses just one signal wire to transmit data to the Arduino. Power comes from separate 5V and ground wires. A 10K Ohm pull-up resistor is needed between the signal line and 5V line to make sure the signal level stays high by default (see the datasheet for more info).

There are two different versions of the DHT11 you might come across. One type has four pins, and the other type has three pins and is mounted to a small PCB. The PCB mounted version is nice because it includes a surface mounted 10K Ohm pull up resistor for the signal line. Here are the pinouts for both versions:



### **How to Set Up the DHT11 on an Arduino**

Wiring the DHT11 to the Arduino is really easy, but the connections are different depending on which type you have.

* VCC - red wire Connect to 3.3 - 5V power. Sometime 3.3V power isn't enough in which case try 5V power.
* Data out - white or yellow wire
* Not connected
* Ground - black wire

Simply ignore pin 3, it's not used. You will want to place a 10 K ohm resistor between VCC and the data pin, to act as a medium-strength pull up on the data line. The Arduino has built-in pull-ups you can turn on but they're very weak, about 20-50K

### **Programming Arduino**

You should have the [Arduino IDE](https://www.arduino.cc/en/Main/Software)software running at this time. Next, it’s necessary to install the DHT Sensor library, which can be done through the Arduino Library Manager:

Sketch→Include Library→Manage Libraries…

Enter “dht” in the search field and look through the list for the “DHT sensor library by Adafruit.” Click the “Install” button, or “Update” from an earlier version.

IMPORTANT: As of version 1.3.0 of the DHT library you will also need to install the Adafruit\_Sensor library, which is also available in the Arduino Library Manager.

Now load up the Examples→DHT→DHTtester sketch.

**Code Explanation**

First we have to include the "DHT.h" Library.

#include "DHT.h"

Then define the digital pin in which the DHT11 is connected to.

#define DHTPIN 2 // Digital pin connected to the DHT sensor

Now define the type of DHT Sensor. Since we're using a DHT11 sensor, we can write like this.

#define DHTTYPE DHT11 // DHT 11

If you have a DHT22, write this.

#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321

Then define the DHT parameter to Initialize DHT sensor.

DHT dht(DHTPIN, DHTTYPE);

Note that older versions of this library took an optional third parameter to tweak the timings for faster processors. This parameter is no longer needed as the current DHT reading algorithm adjusts itself to work on faster procs.

Inside the void setup function, Initialize the Serial Communication and the DHT Sensor.

void setup() {  
 Serial.begin(9600);  
 Serial.println(F("DHTxx test!"));  
  
 dht.begin();  
}

Now, inside the void loop function, let's measure the readings.

Reading temperature or humidity takes about 250 milliseconds! Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor).

float h = dht.readHumidity(); // read humidity

Read temperature as Celsius (the default)

float t = dht.readTemperature(); // read temperature

Read temperature as Fahrenheit (isFahrenheit = true)

float f = dht.readTemperature(true);

Check if any reads failed and exit early (to try again).

if (isnan(h) || isnan(t) || isnan(f)) {  
 Serial.println(F("Failed to read from DHT sensor!"));  
 return;  
}

Compute heat index in Fahrenheit (the default)

float hif = dht.computeHeatIndex(f, h);

Compute heat index in Celsius (isFahreheit = false)

float hic = dht.computeHeatIndex(t, h, false);

Print the measured readings on the Serial Monitor.

Serial.print(F(" Humidity: "));  
Serial.print(h);  
Serial.print(F("% Temperature: "));  
Serial.print(t);  
Serial.print(F("°C "));  
Serial.print(f);  
Serial.print(F("°F Heat index: "));  
Serial.print(hic);  
Serial.print(F("°C "));  
Serial.print(hif);  
Serial.println(F("°F"));

That's all. You can use this code to measure the temperature and humidity values using the DHT11 or DHT22 sensors in any of your projects.

### **Upload Program to the Arduino**