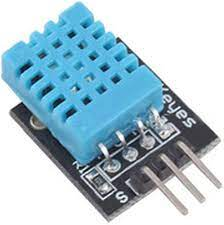
# Interface DHT11 Module With Arduino

Want to keep a log of the climate in your greenhouse, build a humidor control system, or track temperature and humidity data for a weather station project? AOSONG’s DHT11 Temperature and Humidity sensor module could be the perfect fit for you!



This sensor is factory-calibrated and does not require any external components to function. With just a few connections and a bit of Arduino code, you can begin measuring relative humidity and temperature right away.

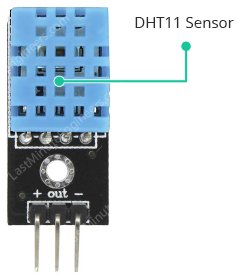
Technical specification:DHT11 Module Hardware Overview

| **Specification** | **Value** |
| --- | --- |
| Sensor Type | DHT11 |
| Measurement Type | Temperature and Humidity |
| Temperature Range | 0°C to 50°C (32°F to 122°F) |
| Humidity Range | 20% to 90% RH |
| Accuracy | ±2°C, ±5% RH |
| Output Type | Digital (Serial) |
| Supply Voltage | 3V - 5.5V |

At the heart of the module is the digital temperature and humidity sensor manufactured by AOSONG – DHT11.

### DHT11 Sensor

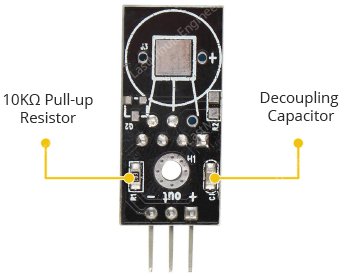
DHT11 can measure temperature from 0°C to 50°C with a ±2.0°C accuracy, and humidity from 20 to 80% with a 5% accuracy.



Note that the DHT11 has a sampling rate of 1Hz, which means it can provide new data once every second.

### Supporting Circuitry

The module includes all of the necessary supporting circuitry, so it can be used straight out of the box.

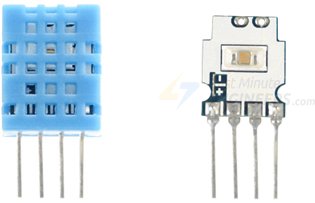


DHT11 sensors typically require an external 10K pull-up resistor on the output pin for proper communication between the sensor and the Arduino. However, because the module already includes a pull-up resistor, you do not need to add one.

The module also includes a decoupling capacitor for filtering power supply noise.

## Inside the DHT11 Sensor

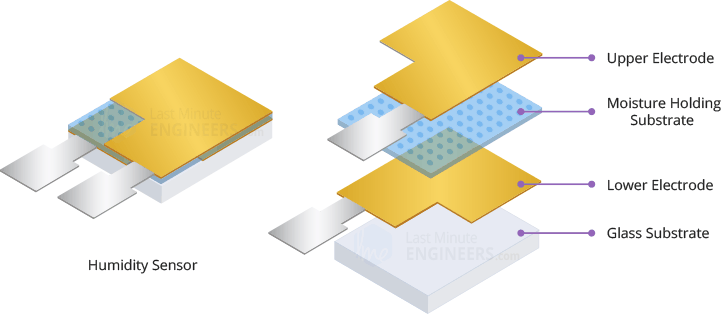
If you remove the sensor’s casing, you will find an NTC thermistor and a humidity sensing component inside.



The humidity sensing component has two electrodes with a moisture-holding substrate (usually a salt or conductive plastic polymer) in between.

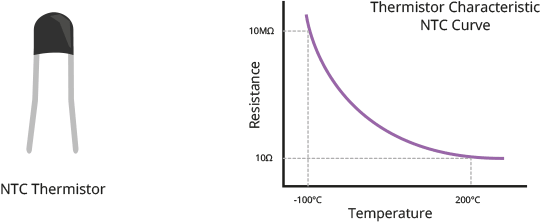
As the humidity rises, the substrate absorbs water vapor, resulting in the release of ions and a decrease in the resistance between the two electrodes.

This change in resistance is proportional to the humidity, which can be measured to estimate relative humidity.

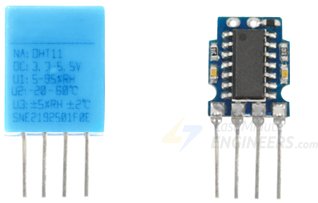


DHt11 also includes a NTC thermistor for measuring temperature. A thermistor is a type of resistor whose resistance varies with temperature.

Technically, all resistors are thermistors in the sense that their resistance changes slightly with temperature, but this change is typically very small and difficult to measure. Thermistors are designed so that their resistance changes dramatically with temperature (by 100 ohms or more per degree). The term “NTC” stands for “Negative Temperature Coefficient,” which means that resistance decreases as temperature rises.

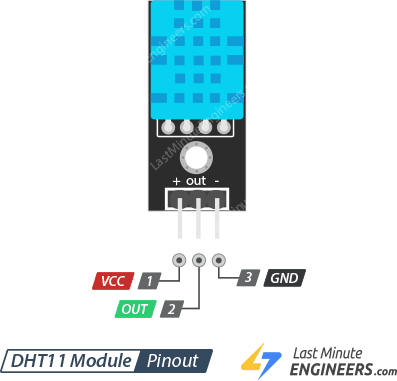


The sensor also includes an 8-bit SOIC-14 packaged IC. This IC measures and processes the analog signal using stored calibration coefficients, converts the analog signal to digital, and outputs a digital signal containing the temperature and humidity.



## DHT11 Module Pinout

The DHT11 module is relatively simple to connect. There are only three pins:



+ (VCC) :pin provides power to the sensor. Despite the fact that the supply voltage of the module ranges from 3.3V to 5.5V, a 5V supply is recommended. With a 5V power supply, the sensor can be placed up to 20 meters away. With 3.3V supply voltage, the sensor can be placed just 1 meter away; otherwise, the line voltage drop will cause measurement errors.

Out pin :is used for communication between the sensor and the microcontroller.

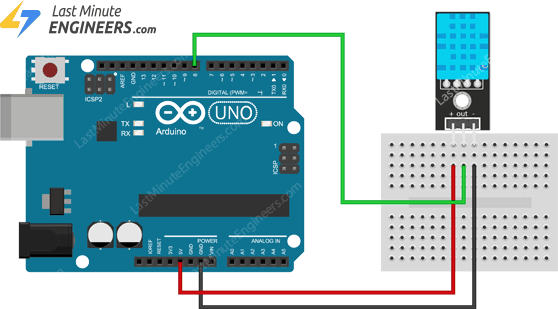
– (GND): is the ground pin.

## Wiring DHT11 Module to Arduino

Now it’s time to connect the DHT11 module to the Arduino!

Connections are relatively simple. Begin by connecting the + (VCC) pin to the Arduino’s 5V output and the – (GND) pin to ground. Finally, connect the Out pin to digital pin #8.

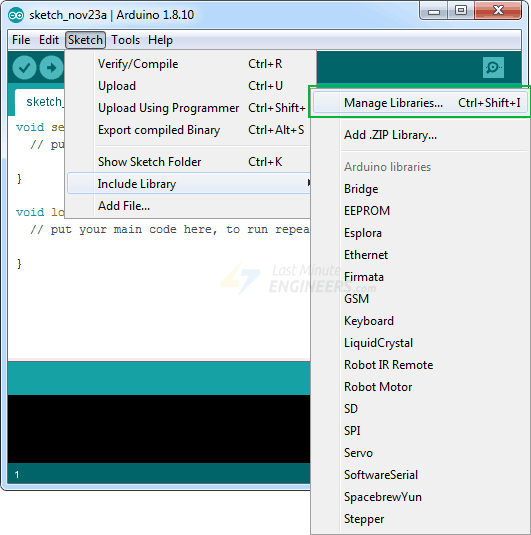
The diagram below shows how to connect everything.



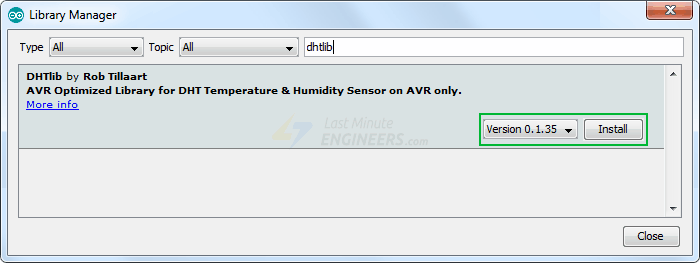
## Installing DHT library

The DHT sensors has their own proprietary single-wire data transfer protocol. This protocol requires precise timing. We don’t have to worry too much about this, though, because we’ll be using the [DHTlib library](https://github.com/RobTillaart/Arduino/tree/master/libraries/DHTlib), which handles almost everything.

To install the library, navigate to Sketch > Include Library > Manage Libraries… Wait for the Library Manager to download the libraries index and update the list of installed libraries.



Filter your search by entering ‘dhtlib’.There should only be a single entry. Click on that and then choose Install.



## Arduino Example 1 – Displaying Readings on Serial Monitor

After installing the library, copy and paste this sketch into the Arduino IDE.

The following test sketch will print the temperature and relative humidity values to the serial monitor. Try out the sketch, and then we’ll go over it in more detail.

#include <dht.h> // Include library

#define outPin 8 // Defines pin number to which the sensor is connected

dht DHT; // Creates a DHT object

void setup() {

Serial.begin(9600);

}

void loop() {

int readData = DHT.read11(outPin);

float t = DHT.temperature; // Read temperature

float h = DHT.humidity; // Read humidity

Serial.print("Temperature = ");

Serial.print(t);

Serial.print("°C | ");

Serial.print((t\*9.0)/5.0+32.0); // Convert celsius to fahrenheit

Serial.println("°F ");

Serial.print("Humidity = ");

Serial.print(h);

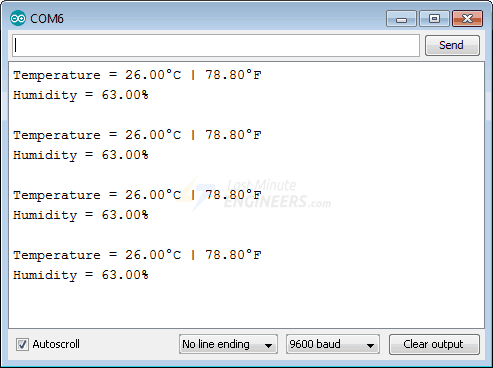
Serial.println("% ");

Serial.println("");

delay(2000); // wait two seconds

}

After uploading the sketch, you should see the following output on the serial monitor.



Where we installed s library for DHT11 sensor ,so here we create one object for that library sd DHT .after that we use the function read to read the value of the sensor and followed by that we store the readed values( value of temperature and value of humidity)in the variable t and h.After that we print those values using serial monitor.

### 