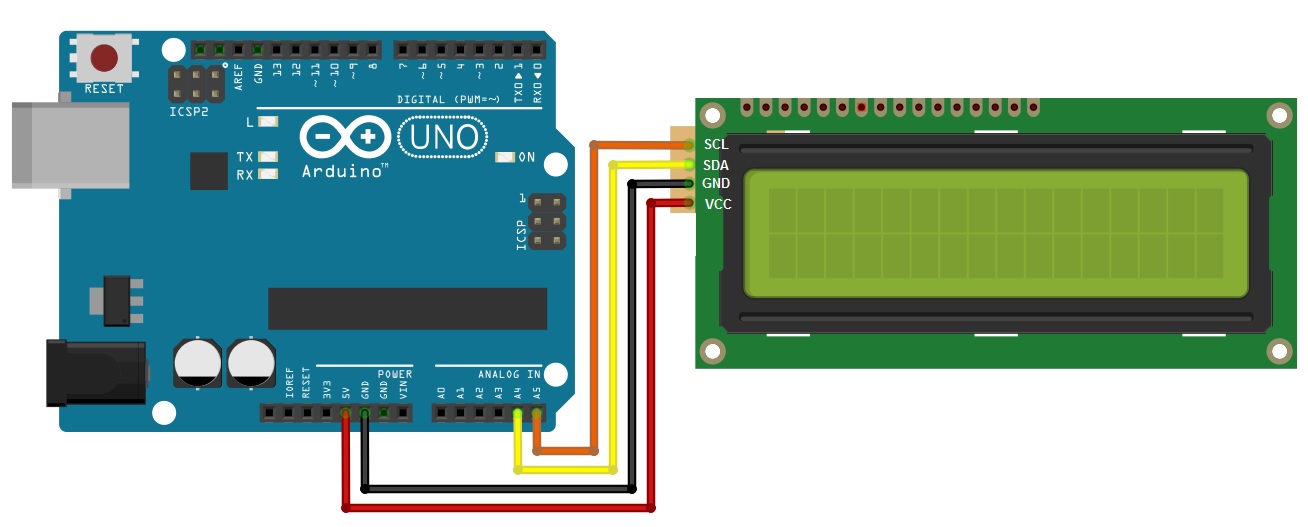
Eco Friend

LCD + I2C

Schematics:



Parts: lcd+i2c

Code:

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x3F, 2,1,0,4,5,6,7,3, POSITIVE);

//LiquidCrystal\_I2C lcd(0x27,20,4); // set the LCD address to 0x27 for a 16 chars and 2 line display

void setup()

{

Wire.begin(8);

lcd.begin(16,2);

lcd.clear();

lcd.print("Home Forestation!");

}

void loop()

{

}

Library:



Remarks: LCD Backlight not working

LDR:

**Definition**

The lux (symbol: lx) is the SI unit of luminance, measuring luminous flux per unit area. It is equal to one lumen per square meter. In photometry, this is used as a measure of the intensity, as perceived by the human eye, of light that hits or passes through a surface.

**Light Measurement**

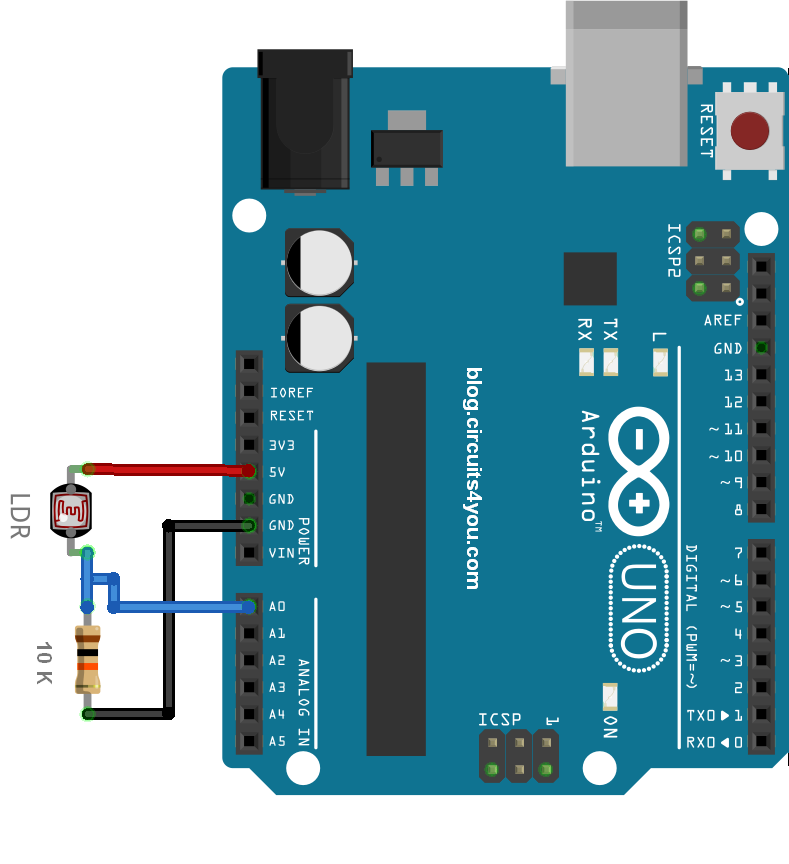
The easiest way to **measure light** with an Arduino is with an LDR. LDR’s (Light dependent resistors) have a low resistance in bright light and a high resistance in the darkness.

Doing that on an Arduino Analog port, would give a reading between 0 and 1024, which of course are really non-descriptive numbers.  What you would want is an output in Lux or Lumen. That is possible but mind you that LDR’s are not really accurate for precise readings. There is a somewhat rough formula that relates the resistance of an LDR to the light in Lux. That is:  
Rldr=500/Lux, or  
Lux=500/Rldr (in kOhm)

**Component Used**

1. Arduino Uno
2. Connecting Wires
3. LDR
4. 10K Ohm resistor

**Light Measurement Circuit**

Arduino Light Measurement Circuit

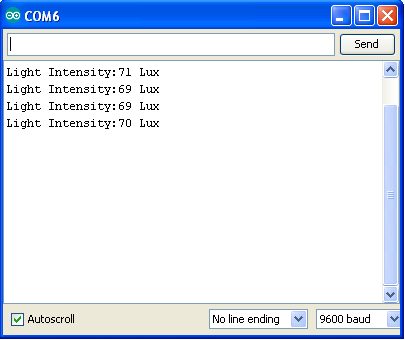
**Arduino Code for Light Intensity Measurement**



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | //Light Intencity Measurement  //www.circuits4you.com    double Light (int RawADC0)  {    double Vout=RawADC0\*0.0048828125;    //int lux=500/(10\*((5-Vout)/Vout));//use this equation if the LDR is in the upper part of the divider    int lux=(2500/Vout-500)/10;    return lux;  }    void setup() {    Serial.begin(9600);  }    void loop() {    Serial.print("Light Intensity:");    Serial.print(int(Light(analogRead(0))));    Serial.println(" Lux");    delay(1000);  } |

**Result of Light measurement**

Open serial monitor and check your creation.



DHT:

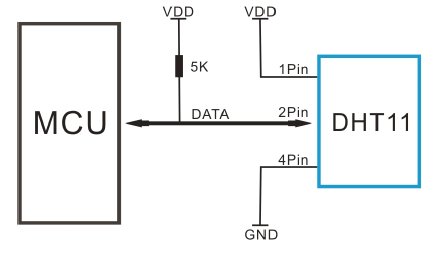
### Description

These DHTXX sensors are very popular among the Arduino Tinkerers. The DHT sensors are inexpensive sensors for measuring temperature and humidity.

These sensors contain a chip that does analog to digital conversion and spits out a digital signal with the temperature and humidity.

These signals are easy to read with any microcontroller (MCU).

**ENROLL IN:**[**Arduino Step-by-step Projects Course - Build 25 Projects**](https://randomnerdtutorials.com/arduino-step-by-step-projects/)



### Specifications DHT11 vs DHT22

There are two versions of the DHT sensor.

**DHT11**

* Range: 20-90%
* Absolute accuracy: ±5%
* Repeatability: ±1%
* Long term stability: ±1% per year
* Price: $1 to $5

**DHT22**

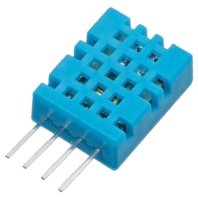
* Range: 0-100%
* Absolute accuracy: ±2%
* Repeatability: ±1%
* Long term stability: ±0.5% per year
* Price: $4 to $10

As you can see from the specs above, the DHT22 is a bit more accurate.

### Where to buy?

You can check [Maker Advisor Tools](https://makeradvisor.com/tools/)‘ page and find the best price for this modules:

* [Click here to find DHT11 best price](https://makeradvisor.com/tools/dht11-temperature-humidity-sensor/)
* [Click here to find DHT22 best price](https://makeradvisor.com/tools/dht22-temperature-humidity-sensor/)



### Arduino with DHT11 Temperature and Humidity Sensor

You need the following components to make this circuit:

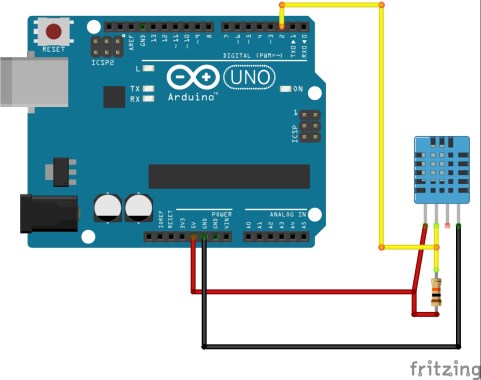
**DOWNLOAD FREE GUIDE:**[Ultimate Guide for Arduino Modules and Sensors](https://randomnerdtutorials.com/complete-guide-for-dht11dht22-humidity-and-temperature-sensor-with-arduino/javascript%3Avoid(0))

* [Arduino UNO](https://makeradvisor.com/tools/compatible-arduino-uno-r3-board/) – read [Best Arduino Starter Kits](https://makeradvisor.com/best-arduino-starter-kits/)
* [DHT11 temperature and humidity sensor](https://makeradvisor.com/tools/dht11-temperature-humidity-sensor/)
* [Breadboard](https://makeradvisor.com/tools/mb-102-solderless-breadboard-830-points/)
* [10K Resistor](https://makeradvisor.com/tools/resistors-kits/)
* [Jumper wires](https://makeradvisor.com/tools/jumper-wires-kit-120-pieces/)

You can use the preceding links or go directly to [MakerAdvisor.com/tools](https://makeradvisor.com/tools/?utm_source=rnt&utm_medium=post&utm_campaign=post) to find all the parts for your projects at the best price!

[https://i2.wp.com/randomnerdtutorials.com/wp-content/uploads/2017/10/header-200.png?ssl=1](https://makeradvisor.com/tools/?utm_source=rnt&utm_medium=post&utm_campaign=post)

Here’s how to connect the DHT11 to an Arduino:

[](https://i1.wp.com/randomnerdtutorials.com/wp-content/uploads/2015/05/humidity_schematics.jpg?ssl=1)

Pins:

* VCC (3V to 5V)
* Data OUT
* Don’t connect
* GND

### Source code

Here’s the code you need for this project:

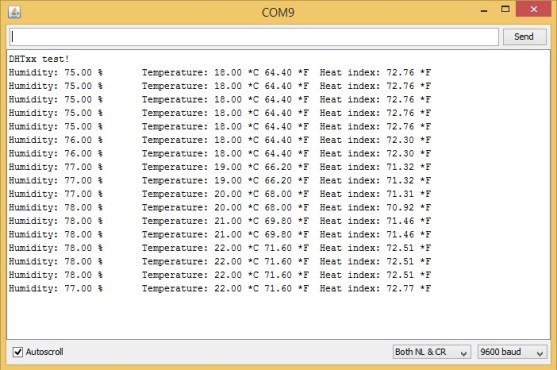
1. Download the [**DHT11 library here**](https://github.com/adafruit/DHT-sensor-library/archive/master.zip)
2. Unzip the DHT library
3. Rename the extracted folder and remove the “-“. Otherwise your Arduino IDE won’t recognize your library
4. Install the DHT11 in your Arduino IDE
5. Restart your Arduino IDE
6. Go to Files / Examples / DHT\_SENSOR\_LIB / DHT Tester
7. Upload the code

// Example testing sketch for various DHT humidity/temperature sensors  
// Written by ladyada, public domain  
  
#include "DHT.h"  
  
#define DHTPIN 2     // what pin we're connected to  
  
// Uncomment whatever type you're using!  
#define DHTTYPE DHT11   // DHT 11   
//#define DHTTYPE DHT22   // DHT 22  (AM2302)  
//#define DHTTYPE DHT21   // DHT 21 (AM2301)  
  
// Initialize DHT sensor for normal 16mhz Arduino  
DHT dht(DHTPIN, DHTTYPE);  
  
void setup() {  
  Serial.begin(9600);   
  Serial.println("DHTxx test!");  
   
  dht.begin();  
}  
  
void loop() {  
  // Wait a few seconds between measurements.  
  delay(2000);  
  
  // 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 temperature as Celsius  
  float t = dht.readTemperature();  
  // Read temperature as Fahrenheit  
  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("Failed to read from DHT sensor!");  
    return;  
  }  
  
  // Compute heat index  
  // Must send in temp in Fahrenheit!  
  float hi = dht.computeHeatIndex(f, h);  
  
  Serial.print("Humidity: ");   
  Serial.print(h);  
  Serial.print(" %\t");  
  Serial.print("Temperature: ");   
  Serial.print(t);  
  Serial.print(" \*C ");  
  Serial.print(f);  
  Serial.print(" \*F\t");  
  Serial.print("Heat index: ");  
  Serial.print(hi);  
  Serial.println(" \*F");  
}

[view raw](https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/raw/master/Projects/DHTtester.ino) [Projects/DHTtester.ino](https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/DHTtester.ino)

### Demonstration

In this project the Arduino is measuring the temperature and humidity. Those two measures are being displayed in the serial monitor. Here’s what you should see in your Arduino IDE serial monitor.



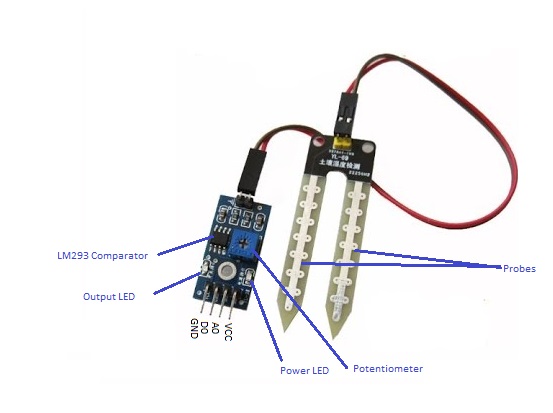
Soil Moistrue:

#### ****Pin Out – Soil Moisture Sensor****

The soil Moisture sensor FC-28 has four pins

* VCC: For power
* A0: Analog output
* D0: Digital output
* GND: Ground

The Module also contains a potentiometer which will set the threshold value and then this threshold value will be compared by the LM393 comparator. The output LED will light up and down according to this threshold value.

[](http://www.circuitstoday.com/arduino-soil-moisture-sensor/pin-out)

Pin Out – Diagram

### **Analog Mode – Interfacing Soil Moisture Sensor and Arduino**

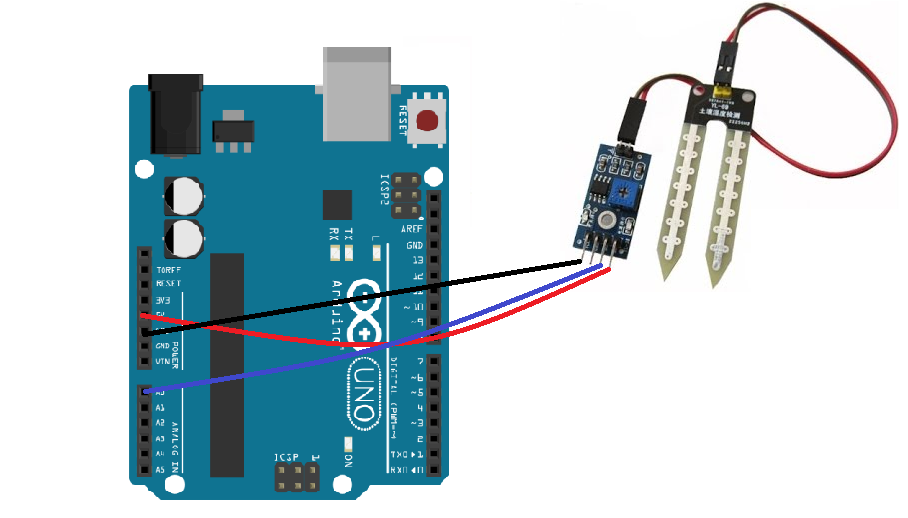
To connect the sensor in the analog mode, we will need to use the analog output of the sensor. When taking the analog output from the soil moisture sensor FC-28, the sensor gives us the value from 0-1023. The moisture is measured in percentage, so we will map these values from 0 -100 and then we will show these values on the serial monitor.

You can further set different ranges of the moisture values and turn on or off the water pump according to it.

#### ****Circuit Diagram****

The connections for connecting the soil moisture sensor FC-28 to the Arduino are as follows.

* VCC of FC-28 to 5V of Arduino
* GND of FC-28 to GND of Arduino
* A0 of FC-28 to A0 of Arduino

[](http://www.circuitstoday.com/arduino-soil-moisture-sensor/interface_soil_sensor_arduino)

Circuit Diagram – Analog Mode

#### ****Analog Code****

int sensor\_pin = A0;

int output\_value ;

void setup() {

   Serial.begin(9600);

   Serial.println("Reading From the Sensor ...");

   delay(2000);

   }

void loop() {

   output\_value= analogRead(sensor\_pin);

   output\_value = map(output\_value,550,0,0,100);

   Serial.print("Mositure : ");

   Serial.print(output\_value);

   Serial.println("%");

   delay(1000);

   }

#### ****Code Explanation****

First of all, we have defined two variables; one for the soil moisture sensor pin and the other for storing the output of the sensor.

int sensor\_pin = A0; // Soil Sensor input at Analog PIN A0

int output\_value ;

In the setup function, the “Serial.begin(9600)” command will help in communication between the Arduino and serial monitor. Then, we will print the “Reading From the Sensor …” on the serial monitor.

void setup() {

   Serial.begin(9600);

   Serial.println("Reading From the Sensor ...");

   delay(2000);

   }

In the loop function, we will read from the sensor analog pin and will store the values in the “output\_ value” variable. Then, we will map the output values to 0-100, because the moisture is measured in percentage. When we took the readings from the dry soil, then the sensor value was 550 and in the wet soil, the sensor value was 10. So, we mapped these values to get the moisture. After that, we printed these values on the serial monitor.

void loop() {

   output\_value= analogRead(sensor\_pin);

   output\_value = map(output\_value,550,10,0,100);

   Serial.print("Mositure : ");

   Serial.print(output\_value);

   Serial.println("%");

   delay(1000);

   }

### **Digital Mode – Interfacing Arduino and Soil Moisture Sensor**

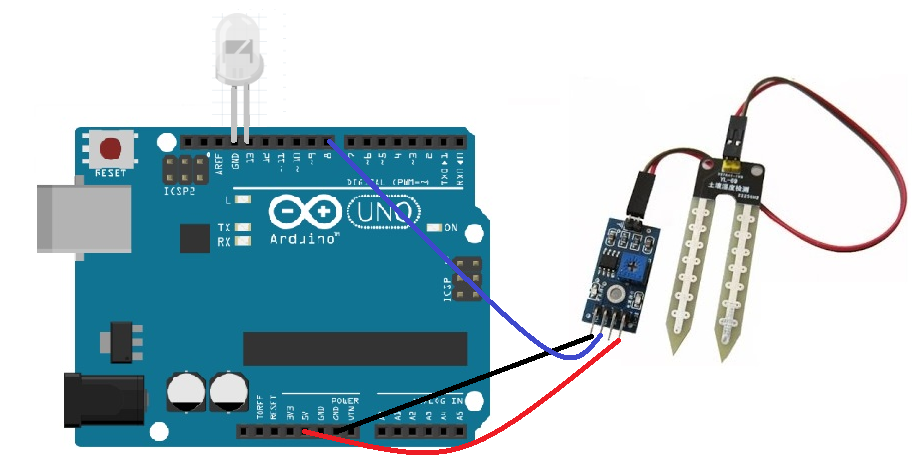
To connect the soil moisture sensor FC-28 in the digital mode, we will connect the digital output of the sensor to the digital pin of the Arduino. The Sensor module contains a potentiometer with it, which is used to set the threshold value. This threshold value is then compared with the sensor output value using the LM393 comparator which is placed on the sensor module.

The LM393 comparator will compare the sensor output value and the threshold value and then gives us the output through the digital pin. When the sensor value will be greater than the threshold value, then the digital pin will give us 5V and the LED on the sensor will light up and when the sensor value will be less than this threshold value, then the digital pin will give us 0V and the light will go down.

#### ****Circuit Diagram****

The connections for connecting the soil moisture sensor FC-28 to the Arduino in digital mode are as follows.

* VCC of FC-28 to 5V of Arduino
* GND of FC-28 to GND of Arduino
* D0 of FC-28 to pin 12 of Arduino
* LED positive to pin 13 of Arduino
* LED negative to GND of Arduino

[](http://www.circuitstoday.com/arduino-soil-moisture-sensor/interface_arduino_soil_moisture_sensor)

Circuit Diagram – Digital Mode

#### ****Digital Code****

int led\_pin =13;

int sensor\_pin =8;

void setup() {

  pinMode(led\_pin, OUTPUT);

  pinMode(sensor\_pin, INPUT);

}

void loop() {

  if(digitalRead(sensor\_pin) == HIGH){

    digitalWrite(led\_pin, HIGH);

  } else {

    digitalWrite(led\_pin, LOW);

    delay(1000);

  }

}

#### ****Code Explanation****

First of all, we have initialized two variable for connecting the LED pin and the Sensor digital pin.

int led\_pin =13;

int sensor\_pin =8;

In the setup function, we have declared the LED pin as the output pin because; we will power the LED through that pin. Then, we declared the sensor pin as input pin because the Arduino will take the values from the sensor through that pin.

void setup() {

  pinMode(led\_pin, OUTPUT);

  pinMode(sensor\_pin, INPUT);

}

In the loop function, we have read from the sensor pin. If the output value of the sensor will be higher than the threshold value, then the digital pin will be high and the LED will light up. If the sensor value will be lower than the threshold value, then the LED will go down.

void loop() {

  if(digitalRead(sensor\_pin) == HIGH){

    digitalWrite(led\_pin, HIGH);

  } else {

    digitalWrite(led\_pin, LOW);

    delay(1000);

  } }

So that finishes our tutorial on interfacing Arduino and Soil moisture sensor.If you got any doubts, please ask in the comments section.

#### Photographs

[](http://www.circuitstoday.com/arduino-soil-moisture-sensor/arduino_moisture_sensor)

[](http://www.circuitstoday.com/arduino-soil-moisture-sensor/arduino_soil)

[](http://www.circuitstoday.com/arduino-soil-moisture-sensor/arduino_soil_moisture)

[](http://www.circuitstoday.com/arduino-soil-moisture-sensor/arduino_soil_sensor)

#### Video

### Applications of Soil Moisture Sensor

A Soil Moisture Sensor has many applications, especially in agriculture. Irrigation is a key factor in farming. Detecting the amount of moisture in the soil and managing irrigation systems (turn on the system when the moisture level falls below a certain predefined value) helps to avoid a lot of wastage of water and human resources. These kinds of sensors make automation of farming easier. This is also used in controlled environments where experiments are conducted.

Bluetoooth: