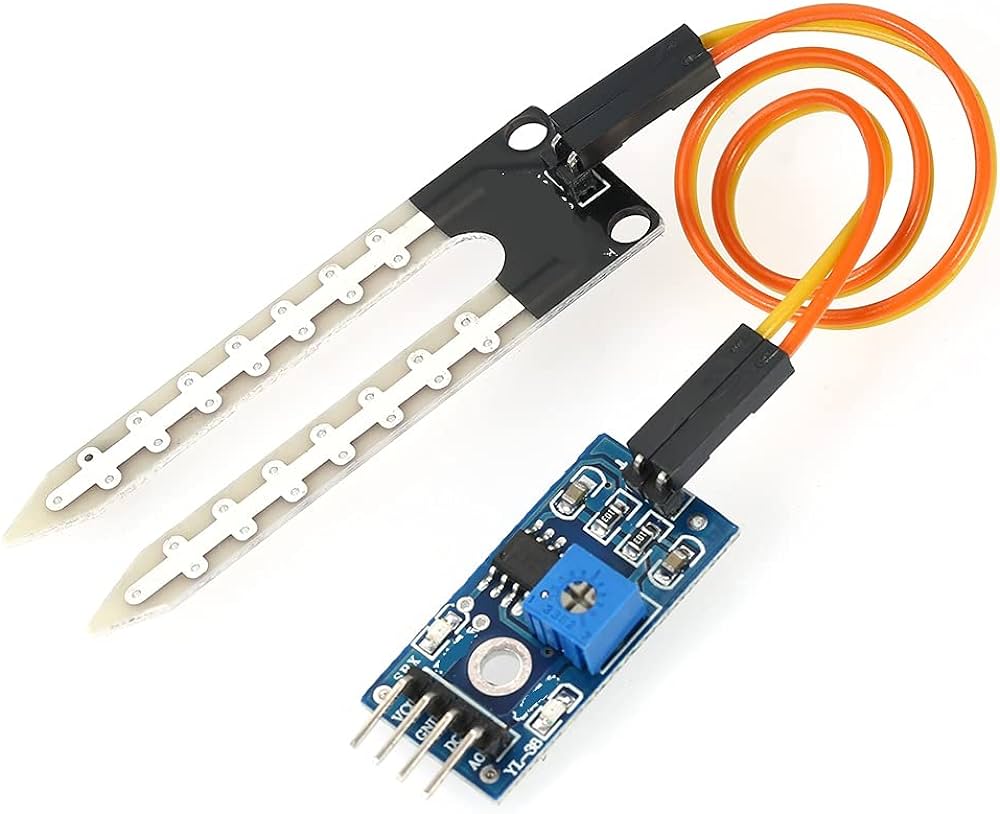
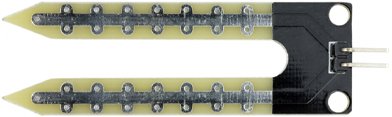
**INTRODUCTION**

A soil moisture sensor serves the purpose of gauging the moisture or water content within the soil. Monitoring water levels is particularly crucial for specific crops. Through the integration of a soil moisture sensor with Arduino, we can create an automated irrigation system capable of watering plants based on their moisture needs. This system helps prevent both overwatering and underwatering.

The soil moisture sensor, when coupled with Arduino support, facilitates the measurement of volumetric water concentration in the soil. This Arduino-compatible moisture sensor is versatile, offering output data in both digital and analog modes. In our project, we will interpret this data, displaying the output status through an LED for digital output. For analog output, we can utilize the serial monitor or employ an LED with Pulse Width Modulation (PWM).

It's essential to note that the soil moisture sensor is also commonly referred to as a soil humidity sensor. This project aims to leverage the capabilities of Arduino to create an efficient and automated irrigation system, optimizing plant care based on real-time soil moisture measurements.

**Specifications :-**

Operating Voltage : 3.3V – 5V

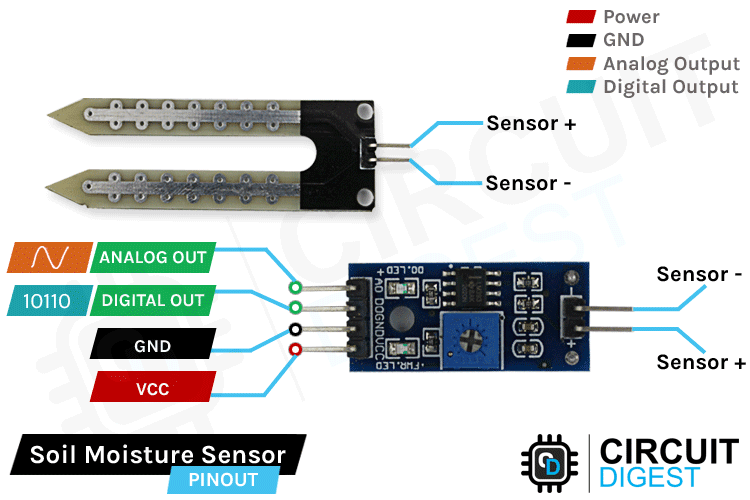
Operating Current : 15 mA

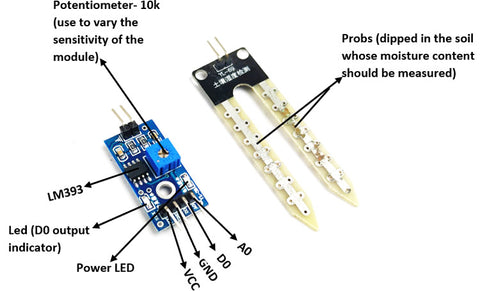
Working Temperature : 10°C - 70°C

Output Digital : 0 – 1023 / 0 – 5V

Output Analog : 0 – 5V

**Soil Moisture Sensor Pinout**





**A0 (Analog Output) -** generates analog output voltage proportional to the soil moisture level, so a higher level results in a higher voltage and a lower-level results in a lower voltage.

**D0 (Digital Output)** - indicates whether the soil moisture level is within the limit. D0 becomes LOW when the moisture level exceeds the threshold value (as set by the potentiometer), and HIGH otherwise.

**VCC** – supplies power to the sensor. It is recommended that the sensor be powered from 3.3V to 5V. Please keep in mind that the analog output will vary depending on the voltage supplied to the sensor.

**GND** – is the ground pin.

**Pinouts of Soil Moisture Sensor**

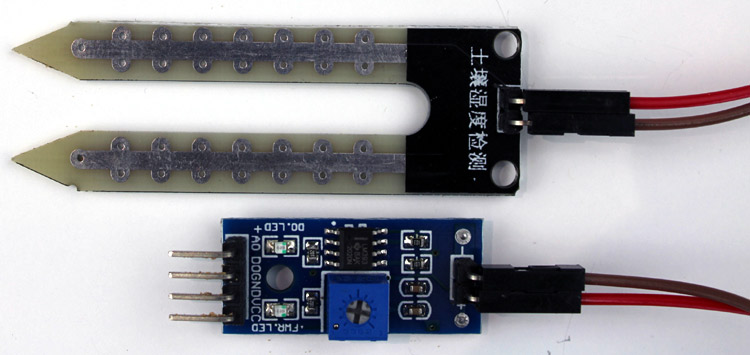
|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | VCC | +5v Power Supply |
| 2 | GND | Ground (-) Power Supply |
| 3 | D0 | Digital Output (0 or 1) |
| 4 | A0 | Analog Output ( 0 – 1023) |

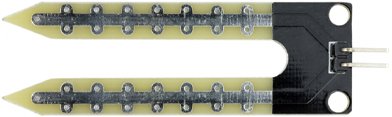
**Hardware Overview**

Soil Moisture sensor consists of two parts:

**The Probe**

The soil moisture sensor comprises a fork-shaped probe featuring two exposed conductors. This probe is inserted into the soil or the location where moisture content needs to be assessed. Operating as a variable resistor, the sensor's resistance undergoes changes based on the moisture levels in the surrounding soil.





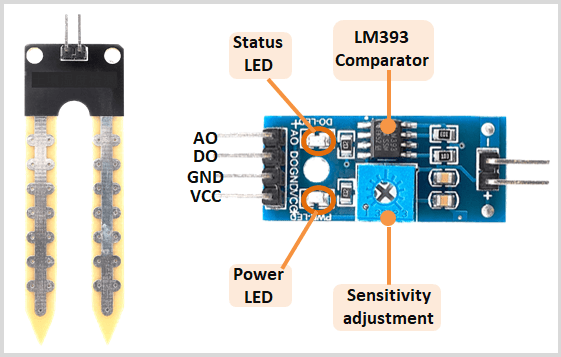
**The Module**

The soil moisture sensor module is there to convert the incoming analog signal to digital signal; this is designed in such a way that the sensor can be used without microcontroller support. The module consists of two signal input pins where the probe gets connected. It also has four other pins, two of which are VCC and GND. The other two are Digital Output and Analog Output pins.

The module generates an output voltage based on the resistance of the probe, which is available at an Analog Output (AO) pin.

The same signal is fed to an LM393 High Precision Comparator, which digitizes it and makes it available at a Digital Output (DO) pin.

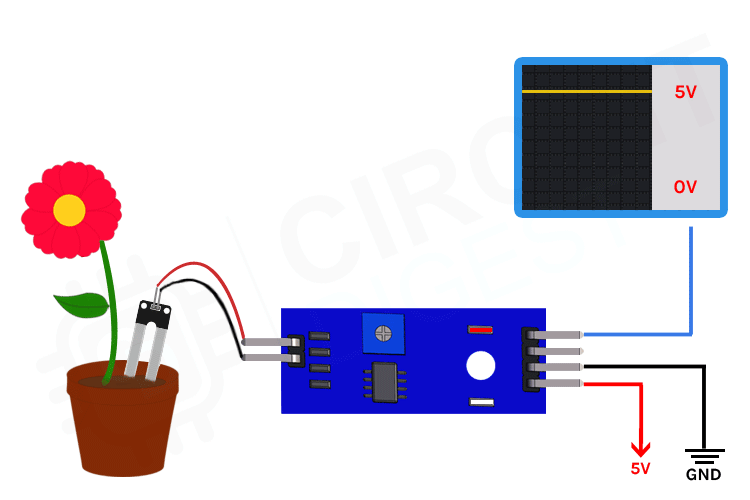
This module also consists of a High Precision Comparator, LM393, that is used to digitize the analog signal coming out of the sensor probe. The module has a built-in potentiometer that is used for sensitivity adjustment of the digital output. The main objective of the potentiometer is to set a threshold, so that when the moisture level exceeds the threshold value, the module will output LOW otherwise HIGH. This feature of the module can come in very handy because when a certain threshold is reached, you can trigger a relay that can start pumping water.



**Working of the sensor**

The fork-shaped probe of the soil moisture sensor functions as a variable resistor, akin to a potentiometer, with its resistance adjusting based on the soil's moisture content. This resistance is inversely related to soil moisture; higher moisture leads to better conductivity and lower resistance, while lower moisture results in poorer conductivity and higher resistance. The sensor produces an output voltage proportional to the resistance, enabling the determination of soil moisture levels.

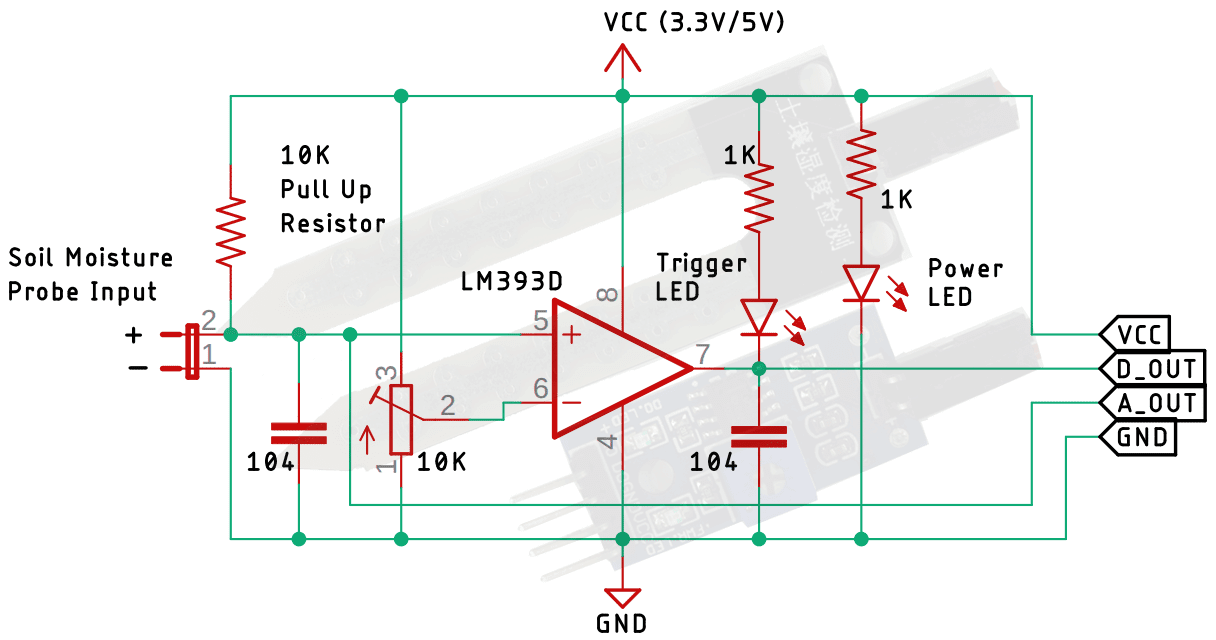
The working principle is straightforward placing the conductive probe into the soil establishes two exposed plates acting as a variable resistor. The resistance of the probe is inversely proportional to soil moisture; greater moisture enhances conductivity, reducing resistance, while lower moisture increases resistance. The Arduino moisture sensor produces output voltage for moisture level determination.



**In the demonstration GIF, the analog output changes as water are added to the soil, with voltage dropping from 5V to 0V. The signal LED on the board indicates this change. The digital pin, not shown in the GIF, transitions from low (0V) to high (5V) when water is added to the soil. The onboard comparator op-amp facilitates this transition, and the sensitivity of the digital pin can be adjusted using the module's potentiometer.**

**Circuit diagram of soil moisture sensor**

The**schematic diagram for the soil moisture sensor module** is shown below. The schematic itself is very simple and needs a handful of generic components to build. If you don't have a prebuilt module on hand but still want to test your project, the schematic below will come in handy.



In the schematic, we have a **LM393 op-amp**which is a **low-power low offset voltage** op-amp that can be powered from a 3.3V or 5V supply. Please note that the analog output voltage of the device will depend on the input voltage. The main job for this op-amp is to convert the incoming analog signal from the sensor probe to digital signal. There is also this 10K potentiometer that is used to set a reference voltage for the op-Amp, the input voltage of the sensor goes below the threshold voltage set by the potentiometer, the output of the op-map goes low. Other than that, we have two LEDs. The first one is a power LED and the other one is the trigger LED. The power LED turns on when power is applied to the board and the trigger LED turns on when a certain set threshold is reached. This is how this basic circuit works.

**Measuring Soil Moisture using Analog Output (A0)**

A diagram of a soil moisture sensor

Description automatically generated

|  |  |
| --- | --- |
| **Arduino Uno/Nano (Pin Number)** | **Soil Moisture Sensor (Pin Number)** |
| **A0** | **A0** |
| **D7** | **VCC** |
| **GND** | **GND** |
| **-** | **D0** |

**Note: - Keep in mind that sensor may be more or less sensitive depending on the type of soil we use. Also, dissolved in water from fertilizers and other sources can affect the sensor output.**

**Finding the threshold value: To estimate the soil moisture level, record the value of your sensor output when the soil is as dry as possible when it is completely saturated.**

**Code:**

#define sensorPower 7 // Sensor pins

#define sensorPin A0

void setup() {

pinMode(sensorPower, OUTPUT);

digitalWrite(sensorPower, LOW); // Initially keep the sensor OFF

Serial.begin(9600);

}

void loop() {

//get the reading from the function below and print it

Serial.print("Analog output: ");

Serial.println(readSensor());

delay(1000);

}

// This function returns the analog soil moisture measurement

int readSensor() {

digitalWrite(sensorPower, HIGH); // Turn the sensor ON

delay(10); // Allow power to settle

int val = analogRead(sensorPin);// Read the analog value form sensor

digitalWrite(sensorPower, LOW); // Turn the sensor OFF

return val; // Return analog moisture value

}

When you run the sketch, you should see readings like the ones below:

* When the soil is dry (around 850)
* When the soil is completely saturated (around 400)

This test may require some trial and error. Once you have the readings, you can use them as a threshold to trigger an action.

**Code:**

The level of soil moisture using the following threshold values:

* < 500 is too wet
* 500 – 750 is the target range
* >750 is dry enough to be watered

/\* Change these values based on your calibration values \*/

#define soilWet 500 // Define max value we consider soil 'wet'

#define soilDry 750 // Define min value we consider soil 'dry'

// Sensor pins

#define sensorPower 7

#define sensorPin A0

void setup() {

pinMode(sensorPower, OUTPUT);

digitalWrite(sensorPower, LOW); // Initially keep the sensor OFF

Serial.begin(9600);

}

void loop() { //get the reading from the function below and print it

int moisture = readSensor();

Serial.print("Analog Output: ");

Serial.println(moisture);

// Determine status of our soil

if (moisture < soilWet) {

Serial.println("Status: Soil is too wet");

} else if (moisture >= soilWet && moisture < soilDry) {

Serial.println("Status: Soil moisture is perfect");

} else {

Serial.println("Status: Soil is too dry - time to water!");

}

delay(1000); // Take a reading every second for testing

// Normally you should take reading perhaps once or twice a day

Serial.println();

}

//This function returns the analog soil moisture measurement

int readSensor() {

digitalWrite(sensorPower, HIGH); // Turn the sensor ON

delay(10); // Allow power to settle

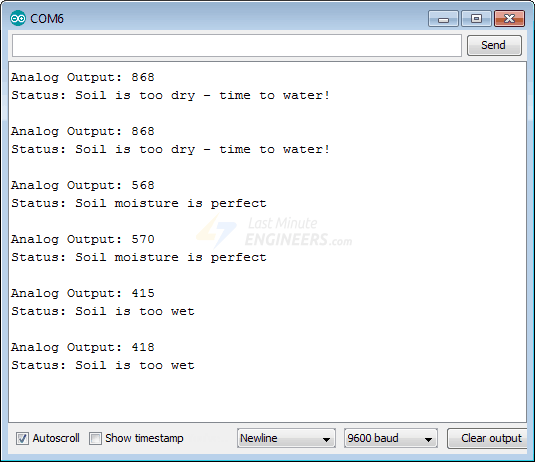
int val = analogRead(sensorPin); // Read the analog value form sensor

digitalWrite(sensorPower, LOW); // Turn the sensor OFF

return val; // Return analog moisture value

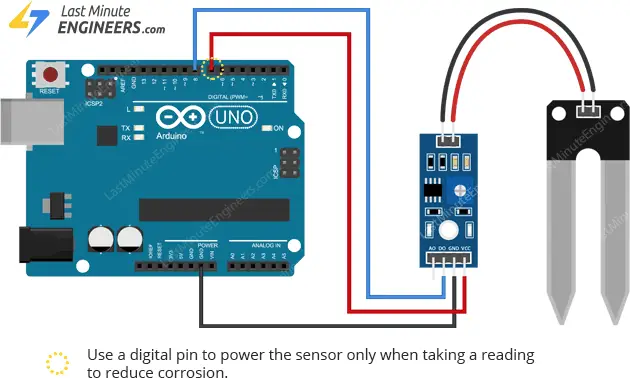
}

**OUTPUT:**





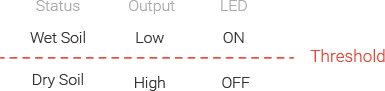
**Measuring soil Moisture using Digital output (D0)**

****



**Note: - Keep in mind that sensor may be more or less sensitive depending on the type of soil we use. Also, dissolved in water from fertilizers and other sources can affect the sensor output.**

**Setting the threshold:** The module has a built-in potentiometer for setting the moisture level threshold above which the module outputs LOW and the status LED lights up.



Now, to set the threshold, stick the probe into the soil when your plant needs watering and turn the pot clockwise until the Status LED is on. Then, turn the pot back counterclockwise just until the LED goes off.

**Code:**

// Sensor pins

#define sensorPower 7

#define sensorPin 8

void setup() {

pinMode(sensorPower, OUTPUT);

// Initially keep the sensor OFF

digitalWrite(sensorPower, LOW);

Serial.begin(9600);

}

void loop() {

//get the reading from the function below and print it

int val = readSensor();

Serial.print("Digital Output: ");

Serial.println(val);

// Determine status of our soil moisture situation

if (val) {

Serial.println("Status: Soil is too dry - time to water!");

} else {

Serial.println("Status: Soil moisture is perfect");

}

delay(1000); // Take a reading every second for testing

// Normally you shoul take reading perhaps every 12 hours

Serial.println();

}

//This function returns the analog soil moisture measurement

int readSensor() {

digitalWrite(sensorPower, HIGH); // Turn the sensor ON

delay(10); // Allow power to settle

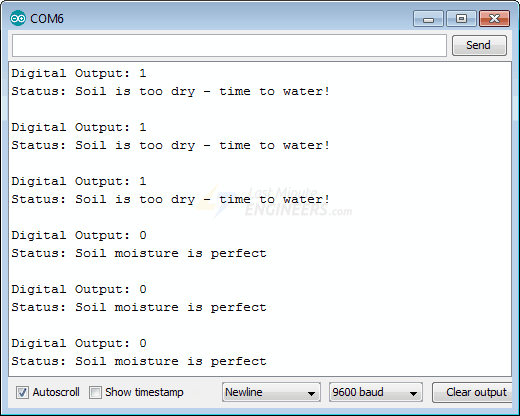
int val = digitalRead(sensorPin); // Read the analog value form sensor

digitalWrite(sensorPower, LOW); // Turn the sensor OFF

return val; // Return analog moisture value

}

**OUTPUT:**





**Analog to Digital conversion:**

Analog Read output = (Input Voltage / Maximum Voltage) \* 1024

Note: Arduino Uno supports 10-bit ADC, which means the resolution of the output is 2^10 = 1024.

We can notice that the sensor reading output goes up to a maximum of Analog value: 876.

Here we are applying 5 Volts to the VCC pin of the sensor, for 5 Volts, we are getting a maximum of 4.28 Volts Approximately in the SIG pin (output pin).

Its equivalent ADC value is 876.

Calculation: (4.28 / 5 ) \* 1024 = 876

So, the output varies from 0 to 876.