

# Soil Moisture measurement using Arduino and Soil Moisture Sensor

Last Updated : 30 Apr, 2024

In This Article, We will learn how to measure the moisture (water content) in soil using the Moisture sensor & Arduino.

Arduino is an open-source electronics platform. It consists ATmega328P 8-bit Microcontroller. It can be able to read inputs from different sensors & we can send instructions to the microcontroller in the Arduino. It provides Arduino IDE to write code & connect the hardware devices like Arduino boards & sensors.

Soil Moisture sensor:

A soil moisture sensor is used to measure the amount of moisture (Water content) present in the soil. Water monitoring is very important for a few crops. A soil moisture sensor is used to automate the process of monitoring moisture levels in the soil.

Working:

It works based on the conductivity of electricity through moisture in the soil. We know that pure water does not conduct electricity, as well as solid salt in the soil, does not conduct electricity, But they together can conduct electricity. Moisture in the soil having salt content helps to conduct electricity. But it is restricted to conduct a limited amount of electricity based on the amount of water content present in water (Resistance). This resistance helps to sense the amount of water content approximately. If the water content is high, the resistance between probes is low. If the water content is less, the resistance between the two probes is high.

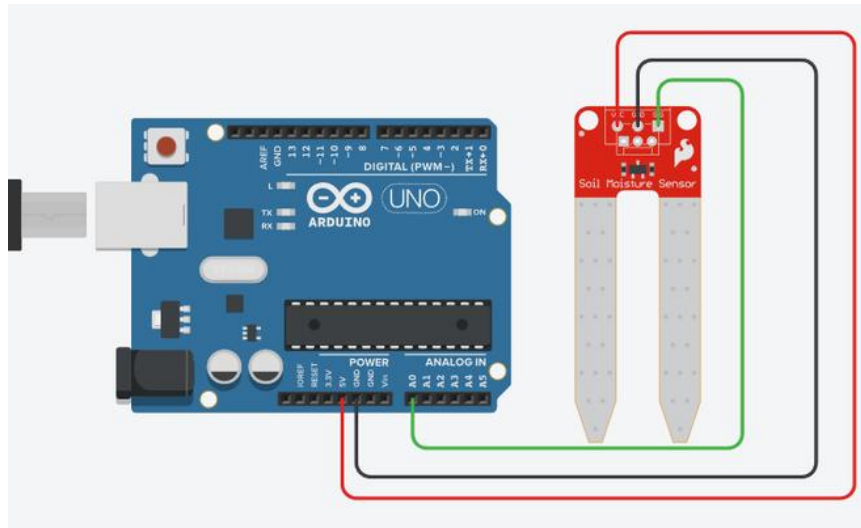
Components Required:

1. Soil moisture sensor
2. Arduino Uno R3
3. 5 x LEDs (Used for Level Indication purposes only)
4. 100 Ohm resistor
5. Jumper wires

Setup:

1. Connect the VCC pin of the sensor to the 5V pin of the Arduino.
2. Connect the GND pin of the sensor to the GND pin of the Arduino.
3. Connect the SIG (signal) pin of the sensor to the A0 (Analog) pin of Arduino.
4. Navigate to Tools in Arduino software and select board and port.
5. Verify and compile the code, then upload the code to the Arduino Uno R3 board.
6. Monitor the output in the Serial monitor (Set the baud rate as 9600). To open Serial monitor Tools>Serial Monitor or (Ctrl+Shift+M).

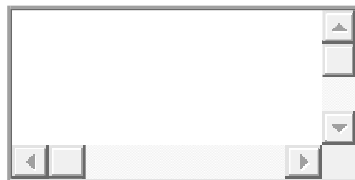
Circuit diagram:



*Soil Moisture Sensor with Arduino*

Arduino code for Analog Reading (output in the serial monitor):

C++



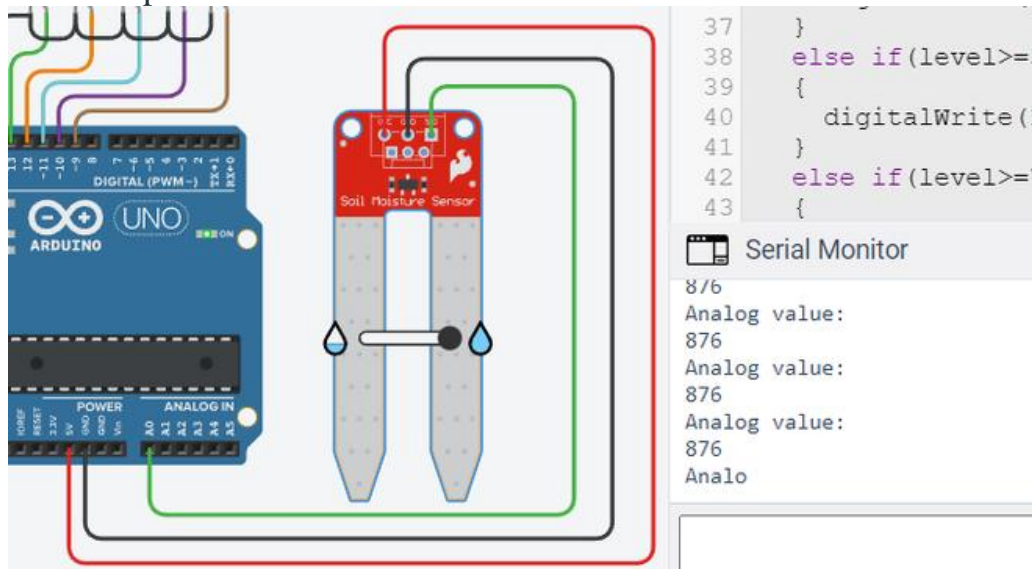
```
void setup() 1  
{ 2  
  // Set the serial monitor baudrate to 9600 3  
  Serial.begin(9600); 4  
} 5  
  
void loop() 6  
{ 7  
  // Variable to store ADC value ( 0 to 1023 ) 8  
  int level; 9  
  // analogRead function returns the integer 10 bit integer (0 to 1023) 10  
  level = analogRead(0); 11  
  
  // Print text in serial monitor 12  
  Serial.println("Analog value:"); 13  
  
  // Print text in serial monitor 14  
  Serial.println("Analog value:"); 15
```

```
// Print analog value in serial monitor
```

```
Serial.println(level);
```

```
}
```

Simulation Output Tinkercad:



Soil Moisture sensor analog reading

*Analog value range:*

*ADC value varies based on the voltage level 0 to 1023.*

*Arduino UNO R3 can measure the voltage level of 0V to 5V.*

*for Analog input 0V output will be 0,*

*for Analog input 5V output will be 1023.*

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.

Note: This range may change based on the hardware manufacturer. Analyze before using the limits.

Demonstrating Moisture level using LEDs

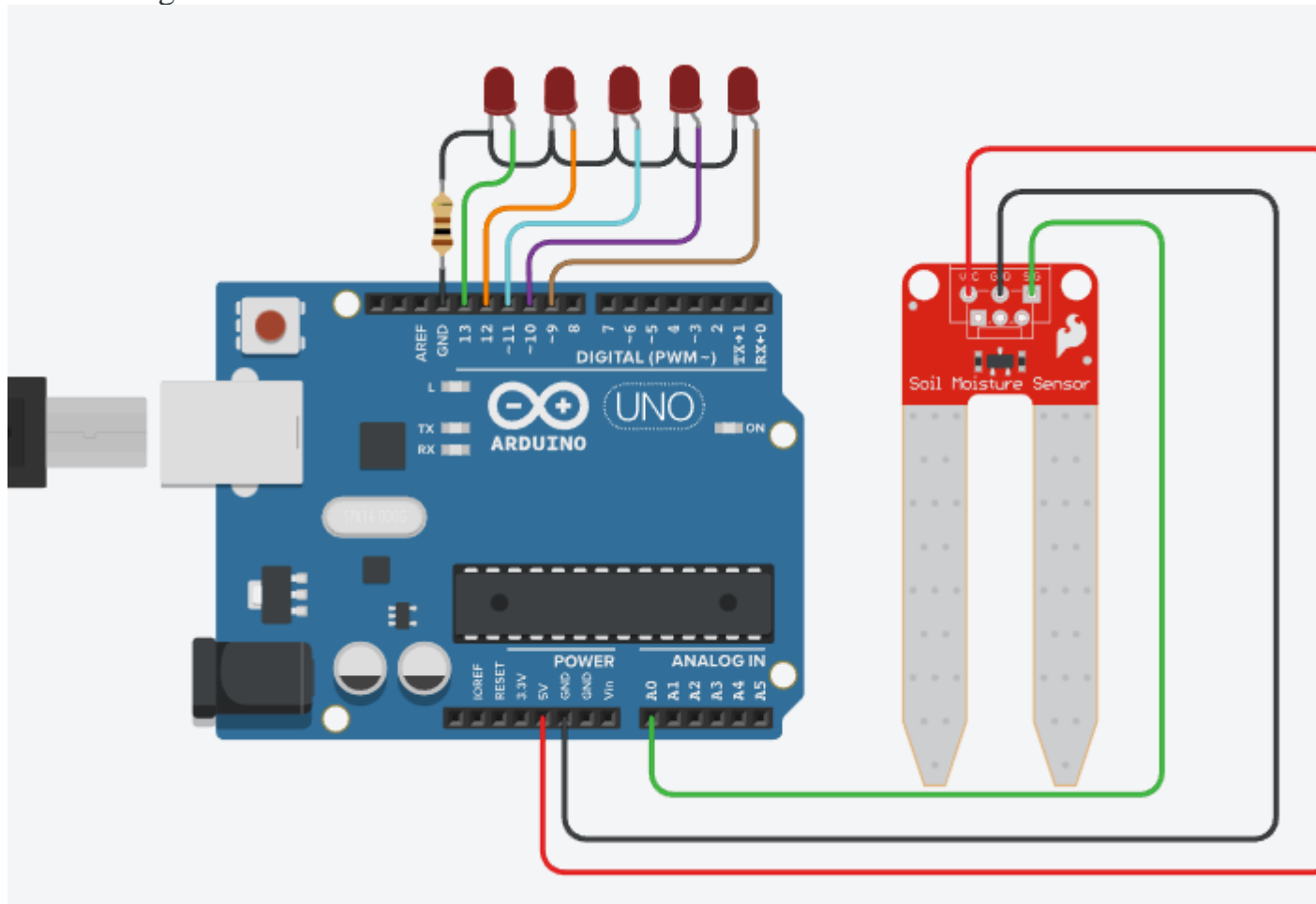
From the Analysis, We got that the output range is 0 to 876. To monitor the moisture level through LED, Use Arduino Digital IO pins to connect the LEDs.

Here we are going to use 5 levels. So we need 5 LEDs.

Setup:

1. Connect +ve terminal of 1st LED Digital 13 PIN ( Level 1).
2. Connect +ve terminal of 2nd LED Digital 12 PIN ( Level 2).
3. Connect +ve terminal of 3rd LED Digital 11 PIN ( Level 3).
4. Connect +ve terminal of 4th LED Digital 10 PIN ( Level 4).
5. Connect +ve terminal of 5th LED Digital 9 PIN ( Level 5).
6. Connect -ve terminal of all the LEDs to the GND pin with a Resistance of 100 Ohm.

Circuit Diagram:



*Soil Moisture Level Indicator Circuit*

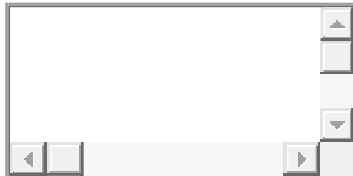
Level Indication:

Level	Analog Value Range
1	0 – 175
2	175 – 350
3	350 – 525

Level	Analog Value Range
4	525 – 700
5	700 – 876

Arduino code for moisture level Indication using LEDs:

C++



```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
void setup()
{
    Serial.begin(
        9600); // Set the serial monitor baudrate to 9600

    pinMode(13, OUTPUT); // Output LED Level 1
    pinMode(12, OUTPUT); // Output LED Level 2
    pinMode(11, OUTPUT); // Output LED Level 3
    pinMode(10, OUTPUT); // Output LED Level 4
    pinMode(9, OUTPUT); // Output LED Level 5
}

void loop()
{
    // Variable to store ADC value ( 0 to 1023 )
    int level;
    // analogRead function returns the integer 10 bit integer (0 to 1023)
    level = analogRead(0);
    // Print text in serial monitor

```

```

Serial.println("Analog value:");
// Print output voltage in serial monitor
Serial.println(level);

// Turn off All LEDs
digitalWrite(13, LOW);
digitalWrite(12, LOW);
digitalWrite(11, LOW);
digitalWrite(10, LOW);
digitalWrite(9, LOW);

// output of sensor varies from 0 to 4.28 Volts,
// It's equivalent ADC value is 0 to 877 ( (4.28/5)*1024 = 877 )

// Splitting 877 into 5 level => 175, 350, 525, 700, 877
// Based on the ADC output, LED indicates the level (1 to 5).

if (level < 175) {
    // LEVEL 1 LED
    digitalWrite(13, HIGH);
}
else if (level < 350) {
    // LEVEL 2 LED
    digitalWrite(12, HIGH);
}
else if (level < 525) {
    // LEVEL 3 LED
    digitalWrite(11, HIGH);
}

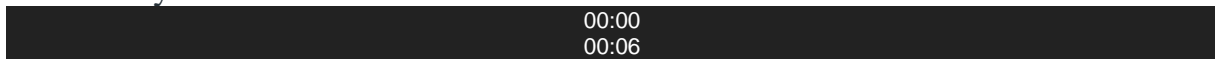
```

```

} 49
else if (level < 700) { 50
    // LEVEL 4 LED 51
    digitalWrite(10, HIGH); 52
} 53
} 54
else if (level < 876) { 55
    // LEVEL 5 LED 56
    digitalWrite(9, HIGH); 57
} 58
}
Output:

```

### Video Player



### Applications:

1. Agricultural and Horticultural research purposes.
2. Climate research
3. Irrigation planning