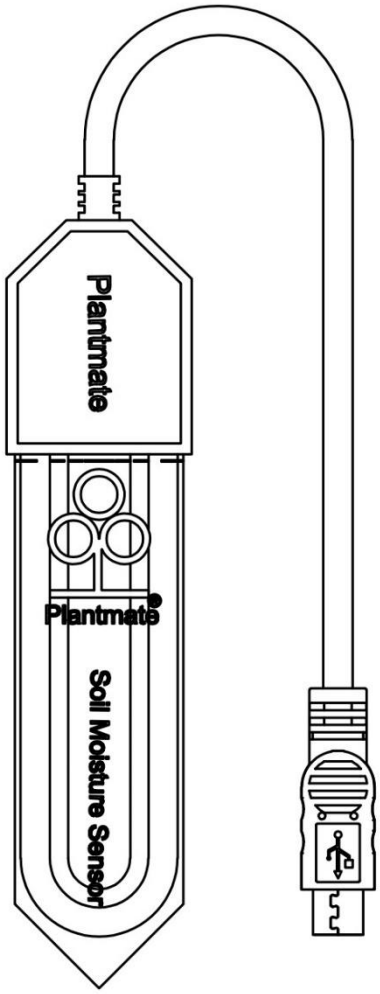
Plantmate® Capacitive Soil Moisture Sensor Module 3.3V



Overview

Whether you're a professional gardener or just a plant lover, it is ideally important to monitor your soil moisture level. There are two types of soil moisture sensors, capacitive and resistive. Although the capacitive sensor and resistive sensor have the same goal in measuring moisture, they differ in their methods. Capacitive sensors are considered to be more accurate and stable. Other than that, the major issue with resistive soil moisture sensors is the corrosion of the sensor probes, not just because it is in contact with the soil but also because there is a DC current flowing which cause electrolysis of the sensors. And that is why the capacitive type of sensor has must longer service life.

Package Content:

- 1 x Plantmate ® Capacitive Soil Moisture Sensor;
- 1 x Miro USB adapter;
- 1 x 4 pin header.

Table of Contents

1. Introduction

Dealing with water or moisture, waterproof is essentially important. The components of the Plantmate ® Capacitive Soil Moisture Sensor are over-molding, it is waterproof;

Plantmate ® Capacitive Soil Moisture Sensor has the sensor probes on both sides of the PCB, it is more sensitive and accurate;

Plantmate ® Capacitive Soil Moisture Sensor has a longer cable to contact with your Arduino board, cable itself is about 85cm with a Micro USB male end. it comes with a Micro USB female adapter to make it even easier to use. Surely that you can cut off the Micro USB male end to contact your Arduino pins directly.

Plantmate ® Capacitive Soil Moisture Sensor operates with 3.3V and it is compatible with low-voltage MCUs. To make it compatible with a Raspberry Pi, an ADC converter is required.

2. Features

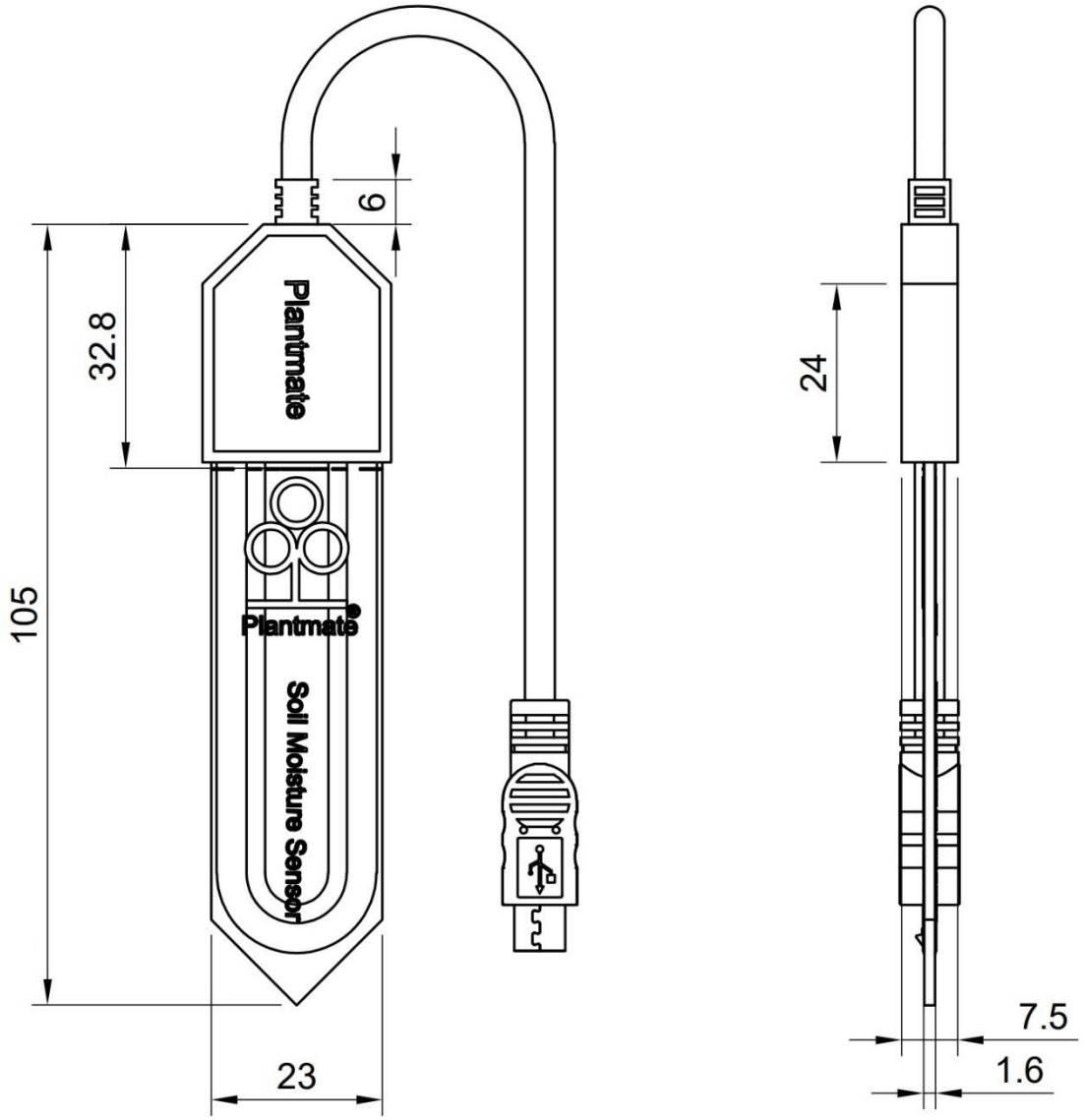
2.1 Specification:

Interface	Micro USB
Operating Voltage	3.3 V DC
Output Voltage	3.0 V DC
Cable length	85 cm
Compatible	Arduino
Weight	32g
Dimensions	105 mm x 25.4 mm (L xW)
UPC/EAN	766133579898
SKU	PMS1121200

2.2 Wire Assignments

Wire	Description
Red	3.3V DC-input (“VCC” on Micro USB)
Gray & White	0-3 V Analog signal output (” D+ & D-” on Micro USB)
Black	GND (“GND” on Micro USB)

2.3 Dimensions



Front View **Side View**

Dimension Unit: mm

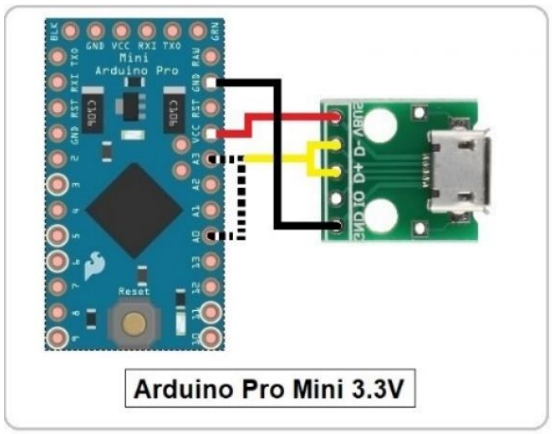
3 Example

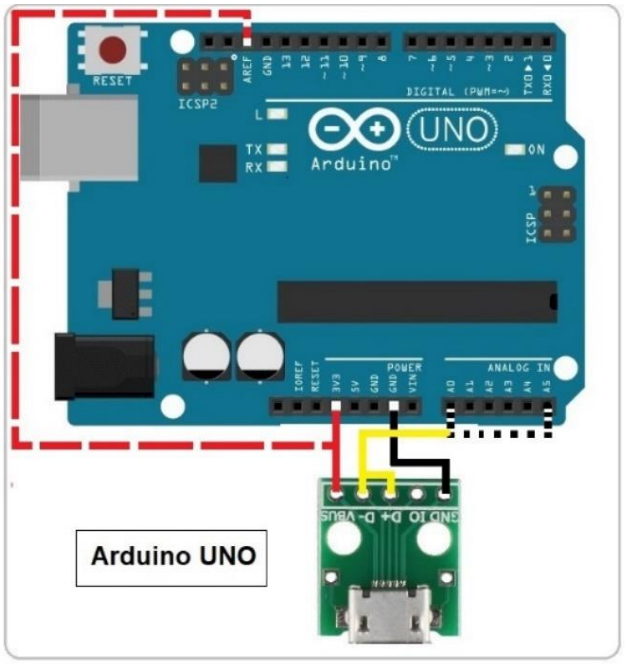
3.1 Requirements:

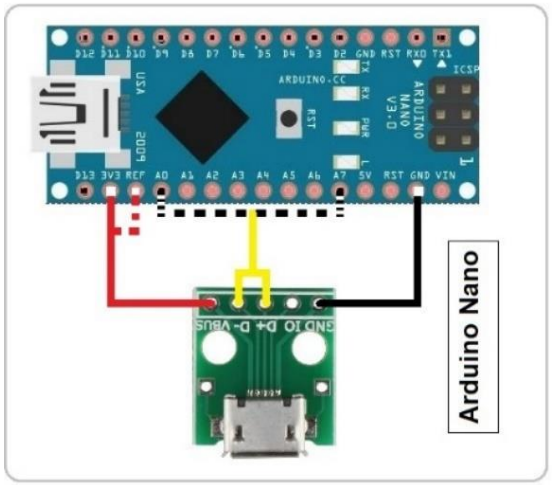
- **Hardware**
  1. Arduino UNO Board
  2. Plantmate Capacitive Soil Moisture Sensor
  3. Breadboard
  4. Male to male jumper wires
  5. Soldering Tools (soldering pins and Micro USB adapter if need.)

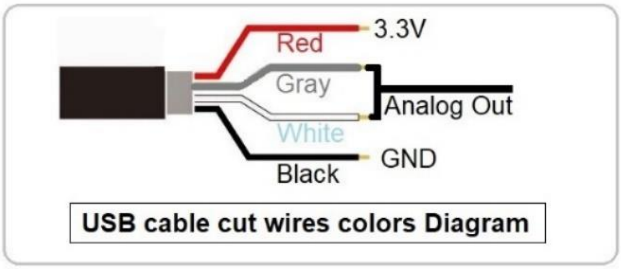
- **Software**
  Arduino IDE V1.6.5 or later.
  Download link: https://www.arduino.cc/en/software

3.2 Connection Diagram









Option wire

Alternative pin

3.3 Calibration

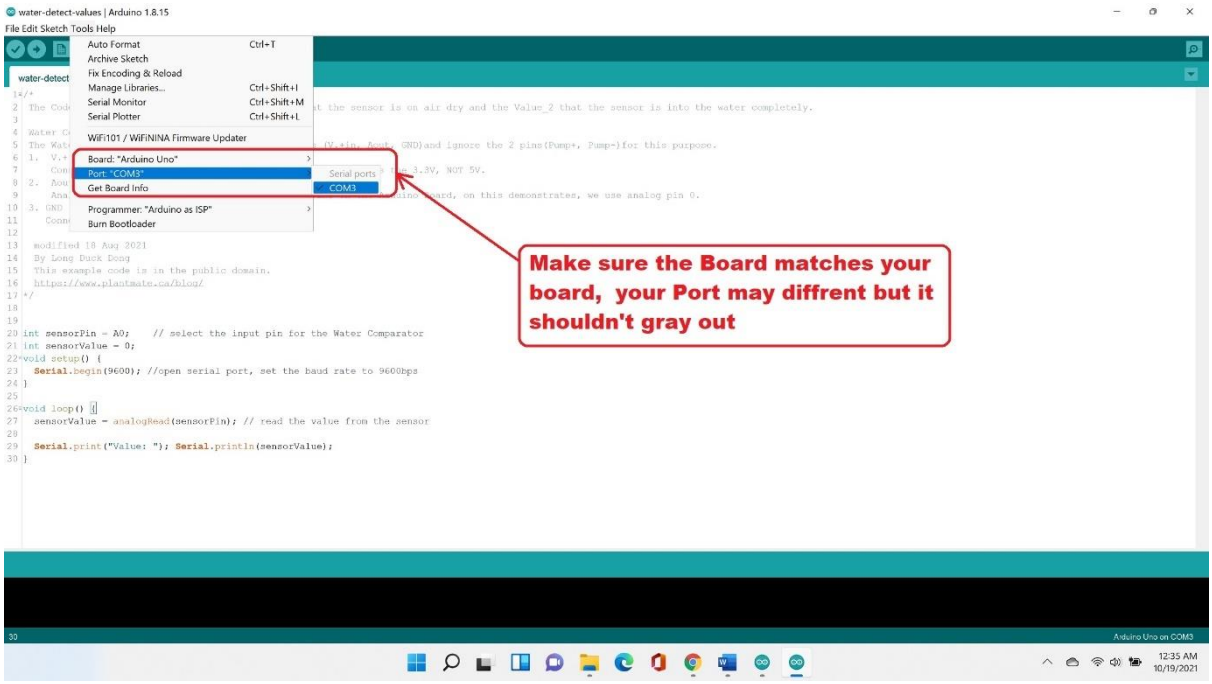
Each moisture sensor does not provide accurate data as expected, but we can get the closest accurate value by running these codes below.

3.4 Calibration Code

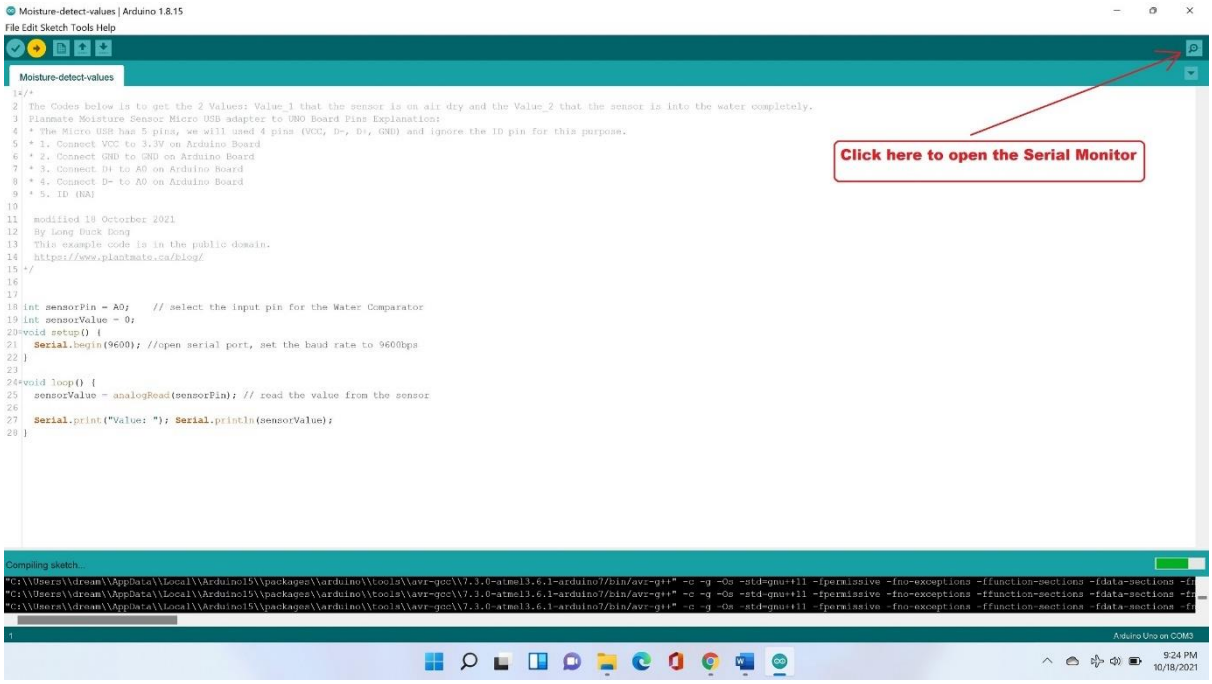
```
/*
 The Codes below is to get the 2 Values: Value_dry that the sensor is on air dry and the Value_wet that the sensor is into the water completely.

 Water Comparator pins explanation:
 The Water Comparator has 5 pins, we will use 3 pins (V.+in, Aout, GND) and ignore the 2 pins (Pump+, Pump -) for this purpose.
 1.  V.+ in
    Connect 3.3V from Arduino Board, please pay attention it is the 3.3V, NOT 5V.
 2.  Aout
    Analog Input can be connected with any analog pins on the Arduino Board, on this demonstrates, we use analog pin 0.
 3. GND
   Connect GND to GND on Arduino.

  modified 18 Aug 2021
  By Long Duck Dong
  This example code is in the public domain.
  https://www.plantmate.ca/blog/
*/


int sensorPin = A0;    // select the input pin for the Water Comparator
int sensorValue = 0;
void setup() {
  Serial.begin(9600); //open serial port, set the baud rate to 9600bps
}

void loop() {
  sensorValue = analogRead(sensorPin); // read the value from the sensor

  Serial.print("Value: "); Serial.println(sensorValue);
}
```

1. On your Arduino IDE, click on 'File' > 'New' to open a new sketch window;
2. Copy the code above then paste it to the sketch window that you just open;
3. Connect the moisture senor Micro USB cable to the Micro USB adapter;
4. Connect the Arduino UNO board to your computer via USB cable;

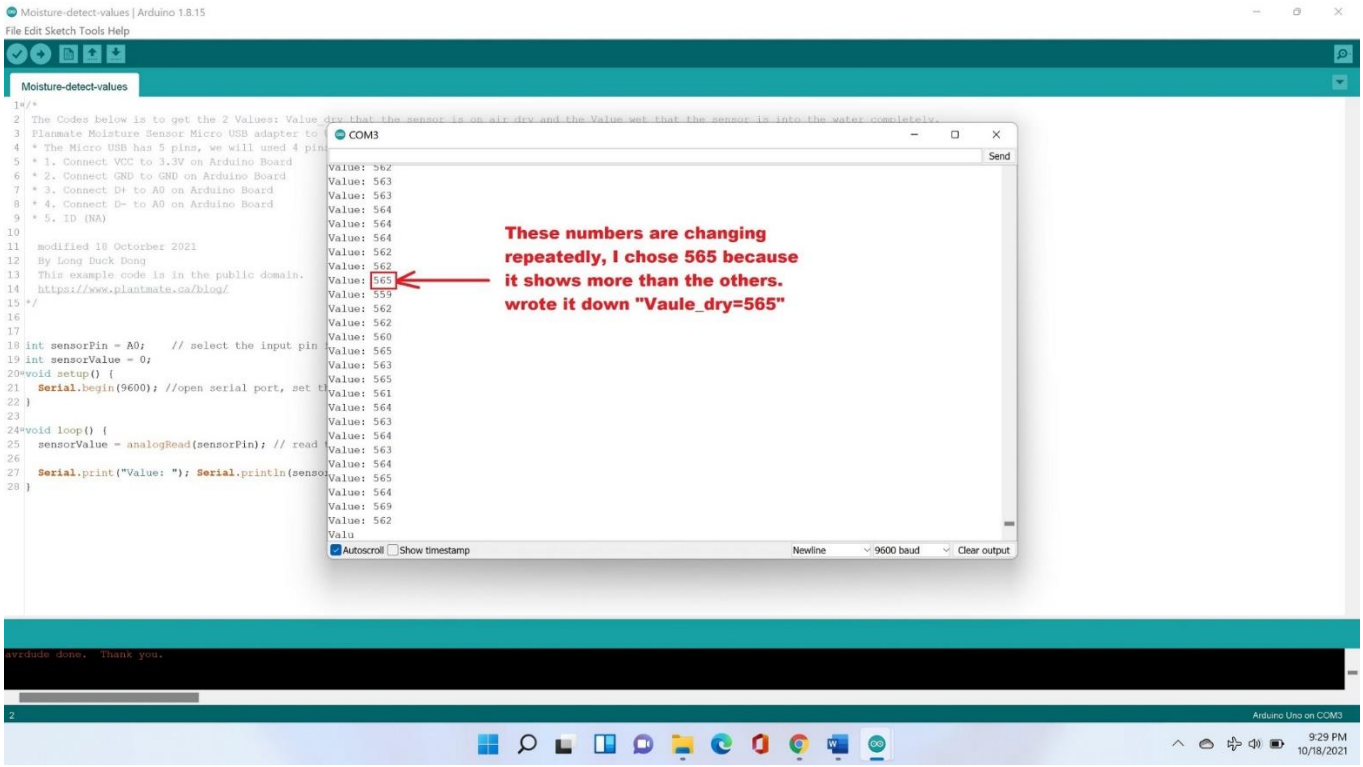
5. Upload the code to your Arduino UNO board, but before that, please click ‘Tools’ on your Arduino IDE to double-check if your board information and the Port is connected.



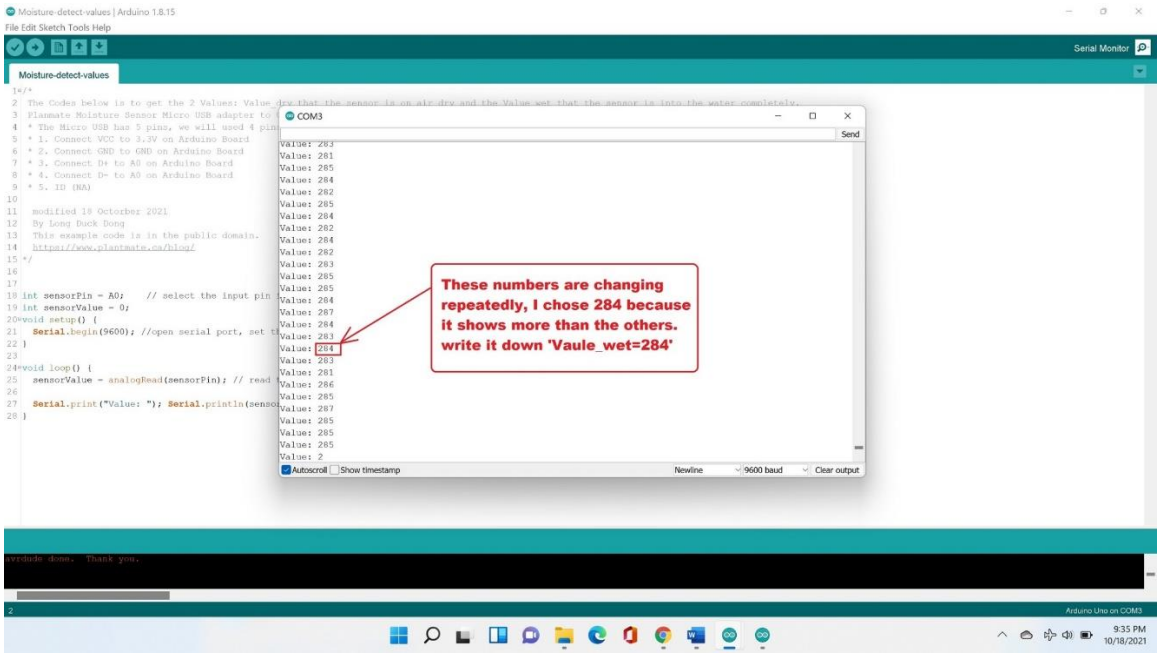
6. After upload the code, leave the moisture sensor on dry, and click the’ Serial Monitor’ on our Arduino IDE.



7. Now you will see the “**Value**s” on your pop-up Serial Monitor window similar as below:



8. Write the **Vaule_dry** on a piece of paper. Now we put the moisture sensor into the water completely, and click the 'Serial Monitor' on your Arduino IDE again and Write the **Vaule_wet** on the paper that you wrote the Vaule_dry.



3.5 Test Code:

Open a new sketch, copy the code below to your new sketch.

Code

```
/*
 * Planmate Moisture Sensor Micro USB adapter to UNO Board PIN connection:
 * 1. VCC to 3.3V
 * 2. GND to GND
 * 3. D+ to A0
 * 4. D- to A0
 * 5. ID (NA)
 */
const int Value_dry = 580;   //replace this value with Value_dry from water-detect-values.ino
const int Value_wet = 280;   //replace this value with Value_wet from water-detect-values.ino
int SensorValue = 0;
int MoisturePercent=0;
void setup() {
  Serial.begin(9600); // use 9600 for 16MHz Arduino
  while (!Serial){;}
  }
void loop() {
SensorValue = analogRead(A0);
Serial.print("Sensor Value: "); Serial.println(SensorValue);
MoisturePercent = map(SensorValue, Value_dry, Value_wet, 0, 100); // Translate Moisture Sensor to a scale of
0% to 100%
                                          // More info:
https://www.arduino.cc/reference/en/language/functions/math/map/
  if(MoisturePercent > 100) //correct the percentage to 100% if read over 100.
    {
      Serial.print("Moisture Percent: "); Serial.println("100 %");
    }
   else if(MoisturePercent <0) //correct the percentage to 0% if read less than 0.
      {
  Serial.print("Moisture Percent: "); Serial.println("0 %");
      }
   else
      {
  Serial.print("Moisture Percent: "); Serial.print(MoisturePercent);
  Serial.println("%");
  delay(250);
      }
  }
```

Replace the **Value_dry** and **Value_wet** with the new values that you wrote down from the prior step.


```
/*
 * Planmate Moisture Sensor Micro USB adapter to UNO Board PIN connection:
 * 1. VCC to 3.3V
 * 2. GND to GND
 * 3. D+ to A0
 * 4. D- to A0
 * 5. ID (NA)
 */
const int Value_dry = 580;   //replace this value with Value_dry from water-detect-values.ino
const int Value_wet = 280;   //replace this value with Value_wet from water-detect-values.ino
int SensorValue = 0;
int MoisturePercent=0;
void setup() {
  Serial.begin(9600); // use 9600 for 16MHz Arduino
  while (!Serial){;}
  }
void loop() {
SensorValue = analogRead(A0);
Serial.print("Sensor Value: "); Serial.println(SensorValue);
MoisturePercent = map(SensorValue, Value_dry, Value_wet, 0, 100); // Translate Moisture Sensor to a sca
                                                                  // More info: https://www.arduino.cc/re
  if(MoisturePercent > 100) //correct the percentage to 100% if read over 100.
    {
      Serial.print("Moisture Percent: "); Serial.println("100 %");
    }
   else if(MoisturePercent <0) //correct the percentage to 0% if read less than 0.
      {
  Serial.print("Moisture Percent: "); Serial.println("0 %");
      }
```

Save the code and upload it to the Arduino UNO.

If you did all the steps correctly, you should see moisture value in percentage (%) on your Serial Monitor.

4. FAQ

You may notice that the Plantmate Capacitive Soil Moisture Sensor is a 3.3V rated sensor and I am using an Arduino UNO which is a 5V rate on analog pins, which means that it can read 0-5V with 1023 steps. That is, 5000 mV/1024 = 4.88 mV, meaning that it can only measure a signal change in the ADC if the voltage increases or decreases by about 5 mV.

The default analog reference of 5 V (on 5V Arduino boards) or 3.3V (on 3.3V Arduino boards)

We can use small voltage output sensors (less than 5 V) on projects. However, this makes the 5 V Arduino boards reading less accurate.

There are many ways to increase the accuracy for a 3.3 V sensor with a 5V operating board,
related articles:

Can I use a 3.3 V rated sensor with a 5 V operating board?

https://support.arduino.cc/hc/en-us/articles/4405245694738-Can-I-use-a-3-3-V-rated-sensor-with-a-5-V-operating-board-

The easy way for this demo project to increase accuracy is using a jumper wire to connect the 3.3V with the AREF pin on your Arduino UNO board.