

**Al-Mustaqbal University**  
**College Of Engineering & Technology**  
**Department of Computer Engineering Techniques**

**(Stage: 3)**

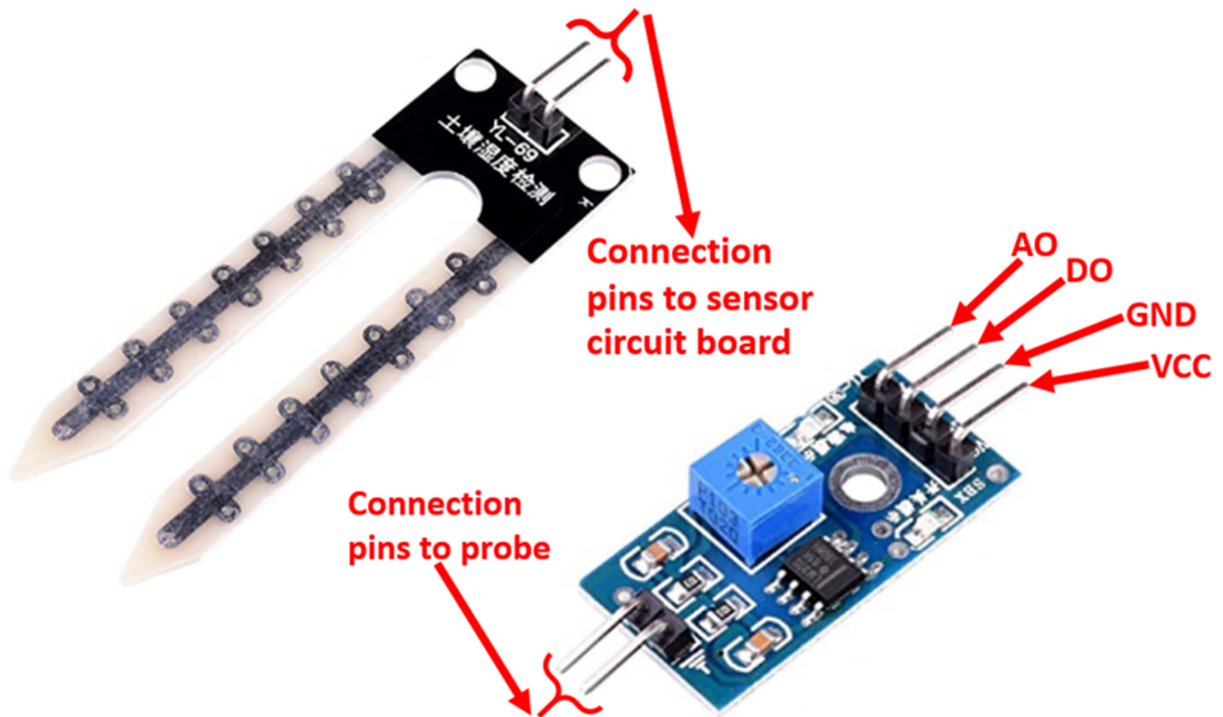
**Digital Control**

**Lecture 13**

**Arduino programming**

**Dr.: Fanar Ali Joda**

**Soil Moisture Sensor**



# Sensor Pins

VCC – Connected to Arduino UNO 5V output.

GND – Connected to Arduino GND.

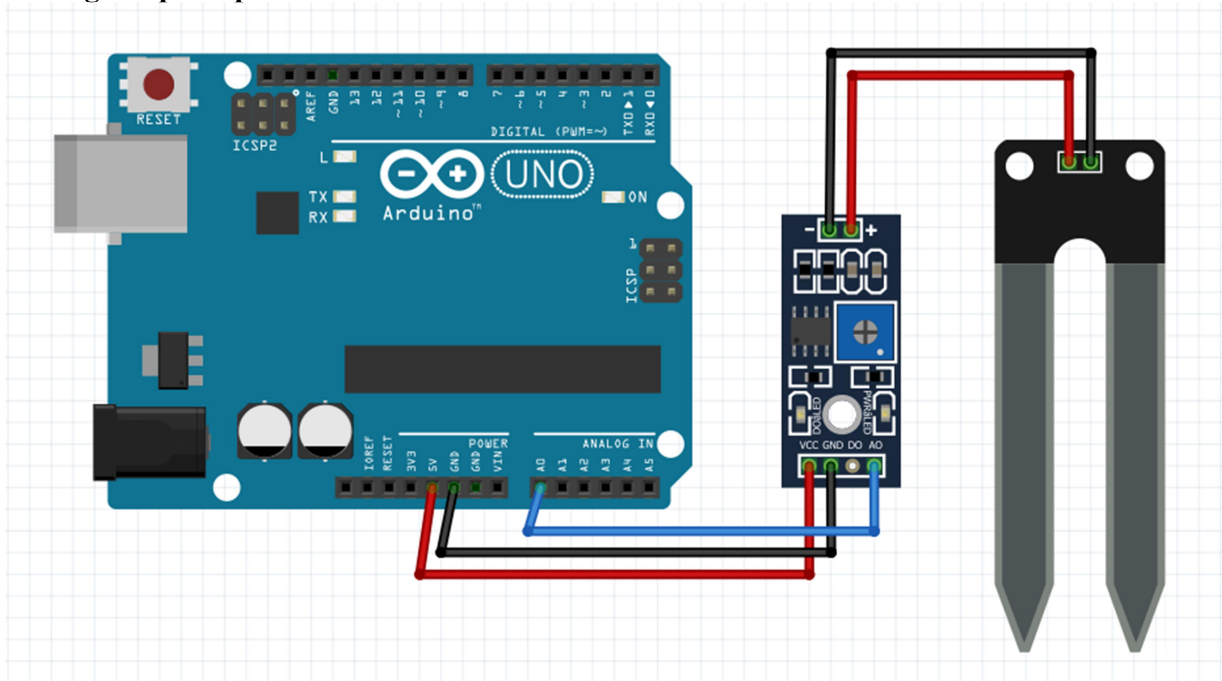
DO – Digital Output

AO – Analog Output

# Connection to Arduino

The sensor can be used through use of the analog output or the digital output. Both options shown below:

## Analog Output Option



## Arduino Code – Analog Output Option

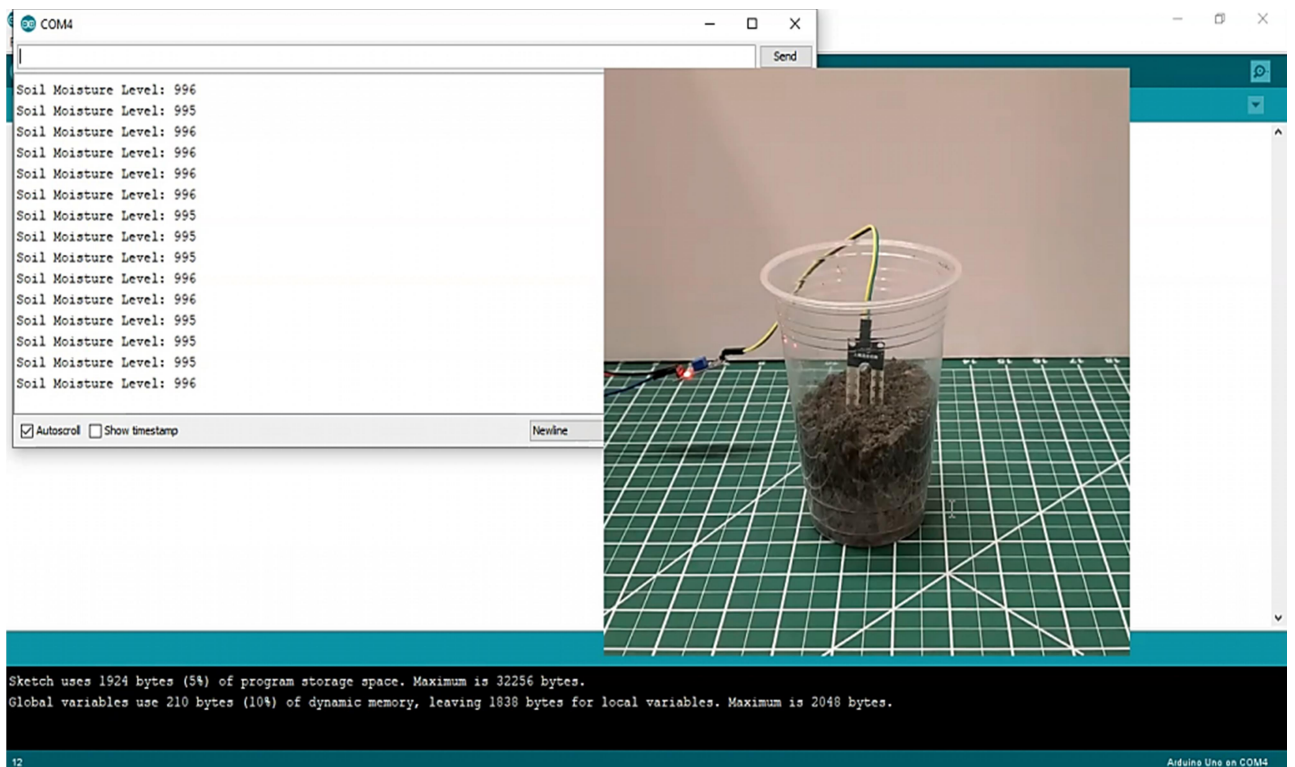
```
//Arduino Code - Soil Moisture
//Define the Analog pin# on the Arduino for the soil
moisture sensor signal
int Moisture_signal = A0; void setup()
{
  Serial.begin(9600); // Start the serial communication
}
void loop()
```

```
{  
  int Moisture = analogRead(Moisture_signal);  
  Serial.print("Soil Moisture Level: ");  
  Serial.println(Moisture);  
  delay(200);  
}
```

**Note // To compute the percentage of moisture value //**

```
moisture_percentage = ( 100 - ( (Moisture /1023.00) * 100 ) );
```

### Result for Analog Output Option

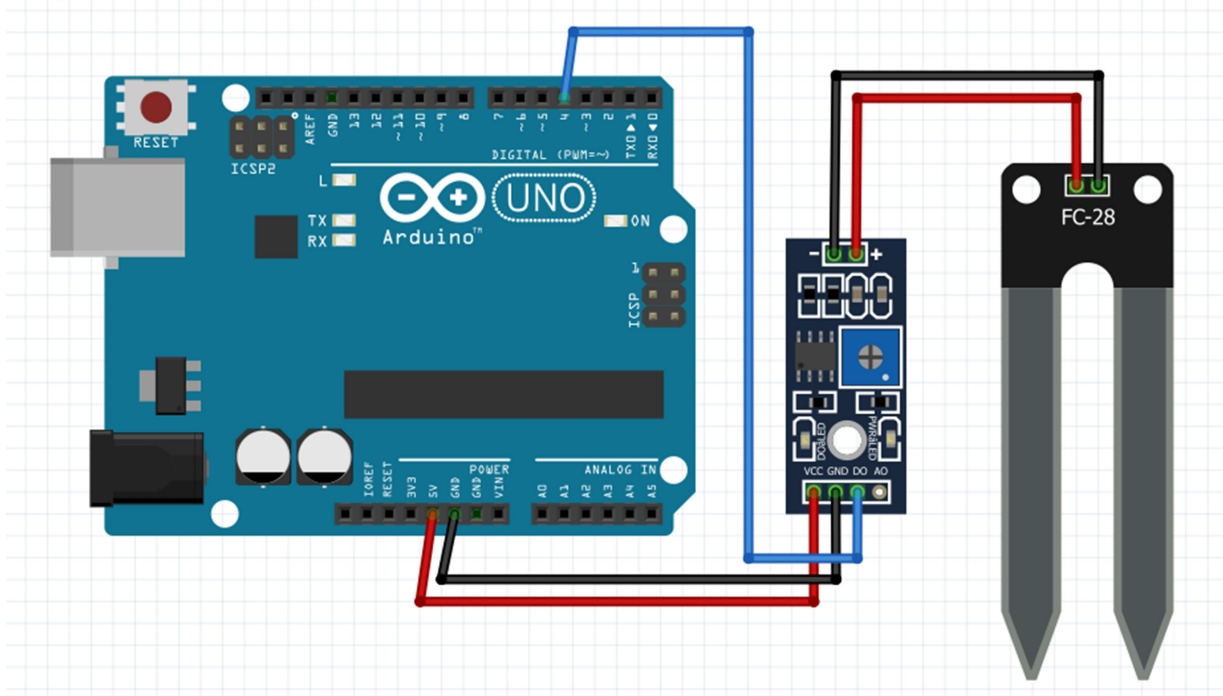


The image shows a screenshot of an Arduino IDE serial monitor window titled 'COM4'. The monitor displays a series of 15 lines of text: 'Soil Moisture Level: 996', 'Soil Moisture Level: 995', 'Soil Moisture Level: 996', 'Soil Moisture Level: 996', 'Soil Moisture Level: 996', 'Soil Moisture Level: 995', 'Soil Moisture Level: 995', 'Soil Moisture Level: 995', 'Soil Moisture Level: 996', 'Soil Moisture Level: 996', 'Soil Moisture Level: 995', 'Soil Moisture Level: 995', 'Soil Moisture Level: 996', 'Soil Moisture Level: 995', and 'Soil Moisture Level: 995'. Below the text, there are checkboxes for 'Autoscroll' (checked) and 'Show timestamp' (unchecked), and a 'Newline' button. To the right of the serial monitor is a photograph of the hardware setup. It shows a clear plastic cup filled with dark soil, placed on a green cutting mat with a white grid. A black sensor module is inserted into the soil. Wires connect the sensor to an Arduino Uno board, which is partially visible on the left. The background is a plain, light-colored wall.

Sketch uses 1924 bytes (5%) of program storage space. Maximum is 32256 bytes.  
Global variables use 210 bytes (10%) of dynamic memory, leaving 1838 bytes for local variables. Maximum is 2048 bytes.

12 Arduino Uno on COM4

## Digital Output Option



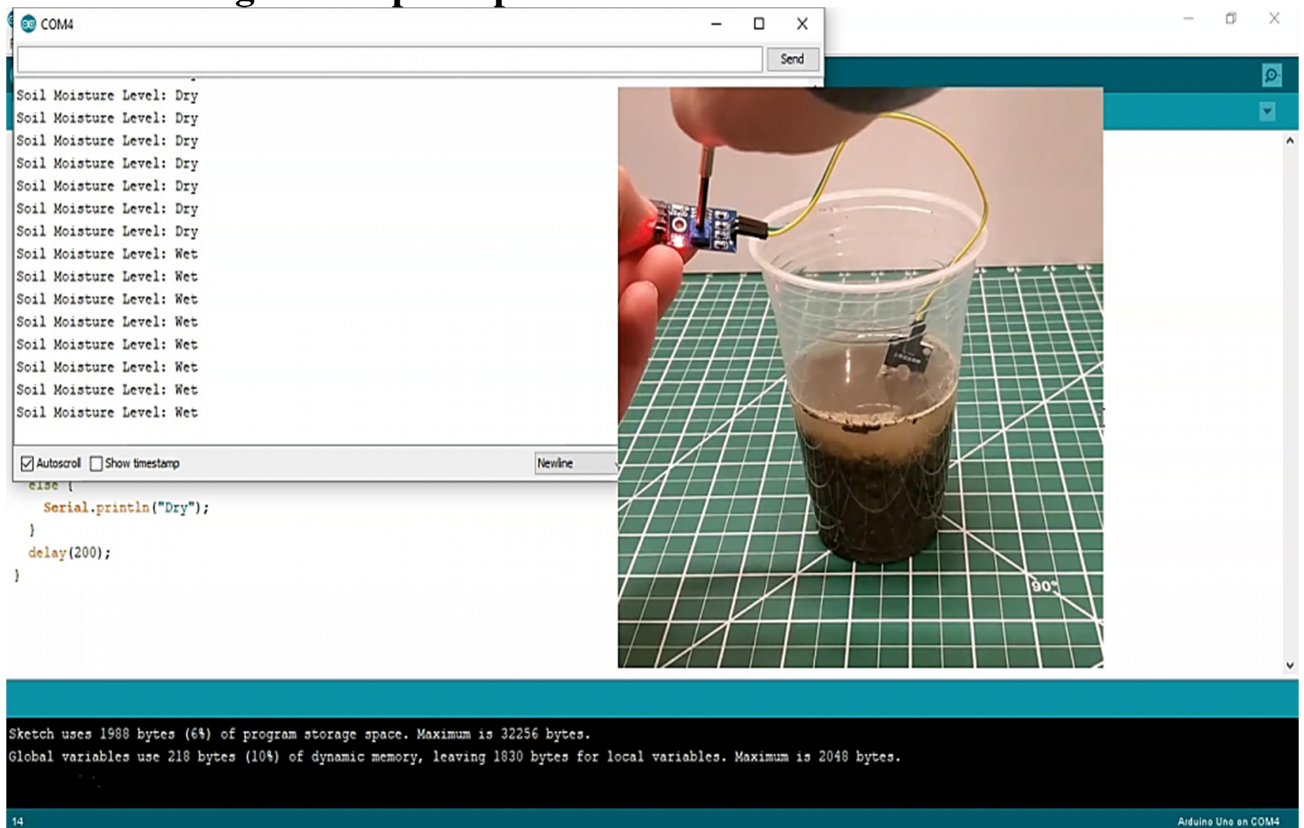
//Arduino Code - Soil Moisture

```
//Define the Digital Input on the Arduino for the sensor signal
int Moisture_dig_signal = 4;
int Sensor_State = 1;
void setup()
{
  pinMode(Moisture_dig_signal, INPUT); //Step pin as input
  Serial.begin(9600); // Start the serial communication
}

void loop()
{
  Serial.print("Soil Moisture Level: ");
  Sensor_State = digitalRead(Moisture_dig_signal);
  if (Sensor_State == 1)
  {
    Serial.println("Wet");
  }
}
```

```
}  
else  
{  
    Serial.println("Dry");  
}  
delay(200);  
}
```

## Result for Digital Output Option



The screenshot shows the Arduino IDE interface. The serial monitor window, titled 'COM4', displays the following output:

```
Soil Moisture Level: Dry  
Soil Moisture Level: Dry  
Soil Moisture Level: Dry  
Soil Moisture Level: Dry  
Soil Moisture Level: Dry  
Soil Moisture Level: Dry  
Soil Moisture Level: Dry  
Soil Moisture Level: Dry  
Soil Moisture Level: Wet  
Soil Moisture Level: Wet  
Soil Moisture Level: Wet  
Soil Moisture Level: Wet  
Soil Moisture Level: Wet  
Soil Moisture Level: Wet  
Soil Moisture Level: Wet  
Soil Moisture Level: Wet
```

Below the serial monitor, the code editor shows the following code snippet:

```
else {  
    Serial.println("Dry");  
}  
delay(200);  
}
```

The photograph shows a hand holding a small blue soil moisture sensor module with a red LED. The sensor is inserted into a clear plastic cup filled with dark soil. The cup is placed on a green grid mat. A yellow wire is connected to the sensor.

Sketch uses 1988 bytes (6%) of program storage space. Maximum is 32256 bytes.  
Global variables use 218 bytes (10%) of dynamic memory, leaving 1830 bytes for local variables. Maximum is 2048 bytes.

14 Arduino Uno en COM4