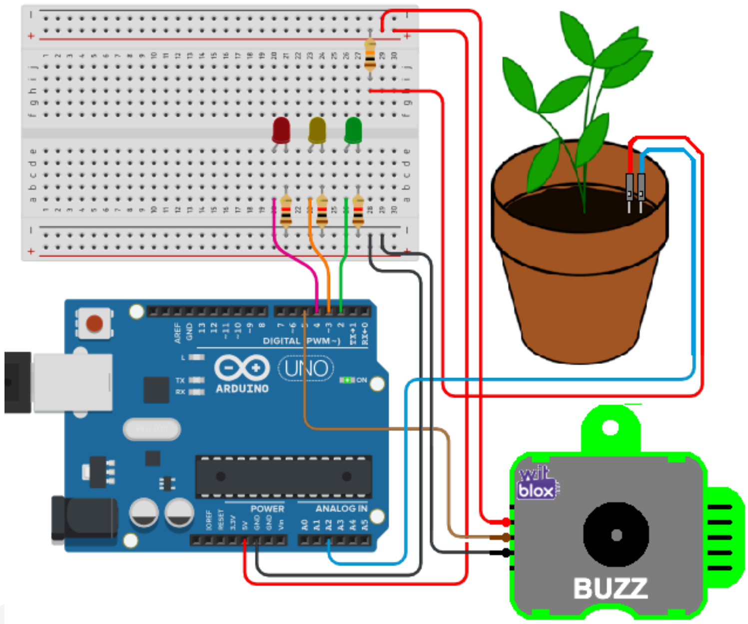
Activity: Make a Continuous Soil Moisture Monitoring System

Objective:

In agriculture and gardening, maintaining optimal soil moisture levels is crucial for the healthy growth of plants. Soil moisture monitoring systems provide an effective solution to ensure that your plants receive the right amount of water. In this project, we will guide you through the process of creating a soil moisture monitoring system using Arduino, colored LEDs, and a buzzer. This system will help you keep your plants thriving by alerting you when it's time to water them.



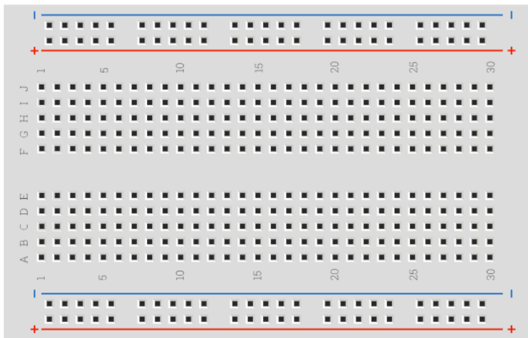
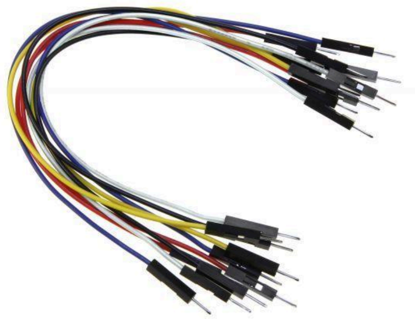
The soil moisture monitoring system works by measuring the moisture level in the soil using a soil moisture sensor. The sensor has two probes that are inserted into the soil. As the soil's moisture content changes, the resistance between the probes also changes. This resistance is converted into a voltage by the sensor, which can be read by the Arduino.

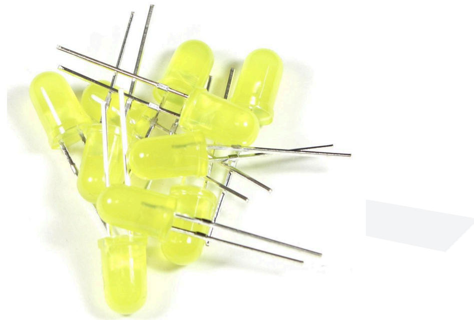
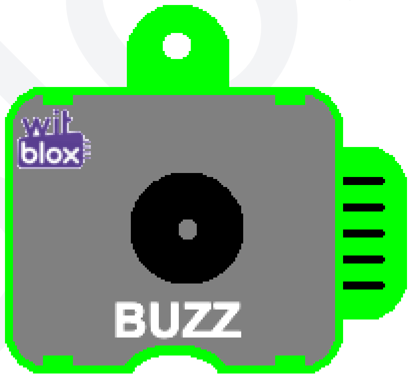
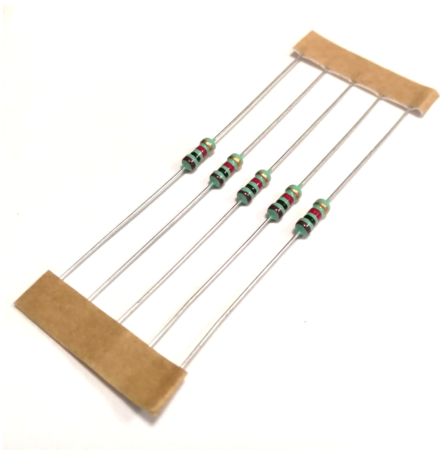
The Arduino processes the moisture level data and triggers actions based on predefined thresholds. In this project, we'll use colored LEDs and a buzzer to indicate the soil moisture condition:

1. **Green LED:** Lights up when the soil moisture is optimal, indicating that the soil has sufficient moisture.
2. **Yellow LED:** Lights up as a warning when the soil is getting drier but is not yet critically dry.
3. **Red LED:** Lights up and the buzzer sounds when the soil moisture level drops below the critical threshold, indicating that it's time to water the plant.

So, in today's activity we will learn how to make a Soil Moisture Monitoring system that uses colored LEDs for moisture level indication, Buzzer for Dry Alarm and an Arduino to process all the inputs and outputs in the system. This system will help you maintain healthy plants by ensuring they receive the right amount of water, ultimately contributing to a flourishing garden or indoor plant collection.

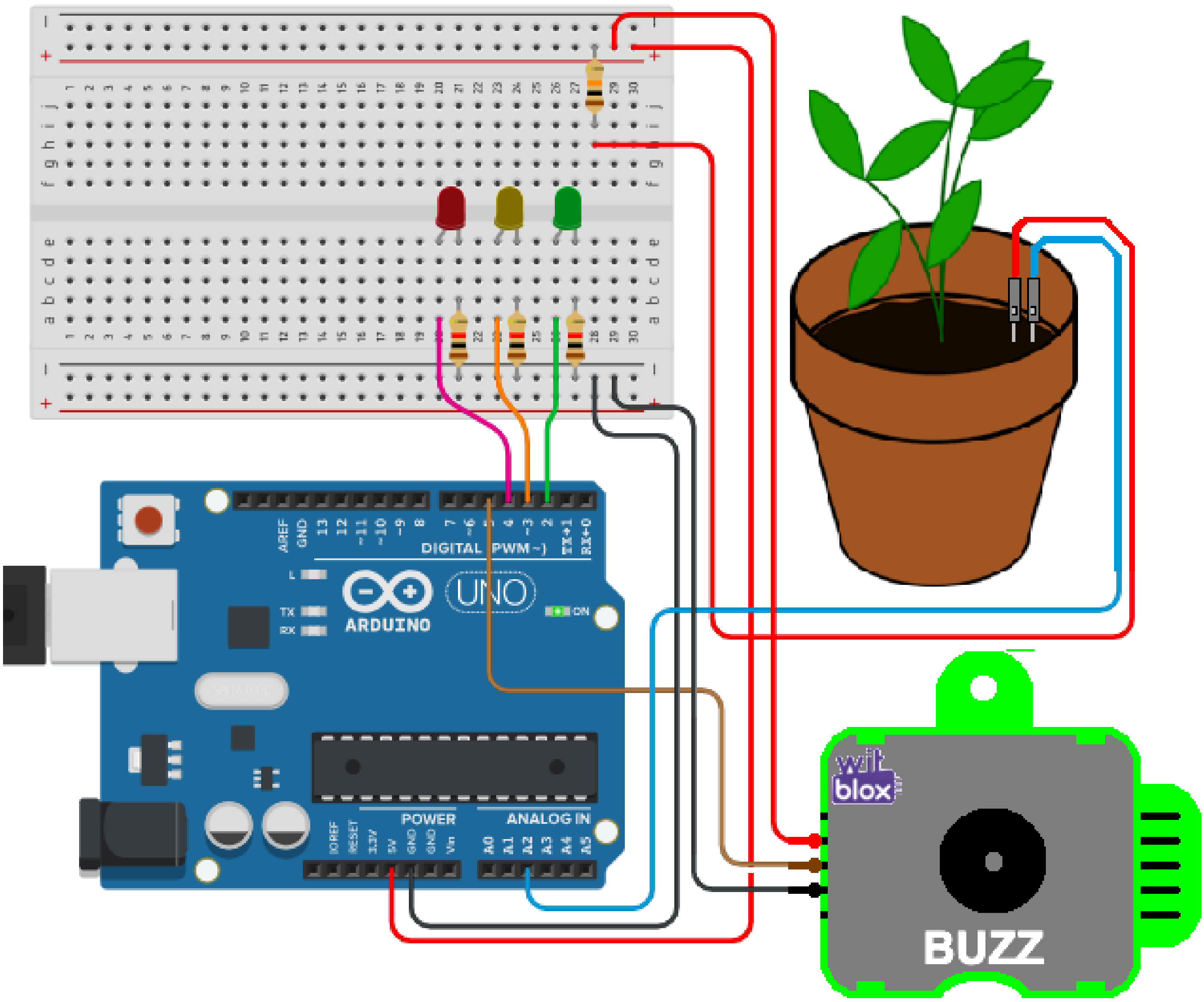
Materials Required:

S.no.	Item	Qty	Image
1	Arduino	1	
2	Breadboard	1	
3	M-M Connection Wires	10	
4	5mm Red LED	1	

5	5mm Green LED	1	
6	5mm Yellow LED	1	
7	Buzz Blox	1	
8	1k Resistor (Brown Black Red Golden)	3	
9	Arduino USB cable	1	

Connection Diagrams:

The connection diagram shown below is built around the Arduino UNO.



Explanation:

Soil moisture is simply the amount of water contained in the form of tiny microscopic droplets in a given volume of soil. The special technique here is how this moisture content of the soil can be read as an analog value by the Arduino.

It is just 2 wires, whose metal tips are used as Moisture sensitive inputs. Since water (moisture) conducts electricity, we can assume that more moisture will cause more conductivity and less moisture will cause less conductivity. Thus, we get an electrical change w.r.t. a physical change.

The Moisture sensitive terminals are such that one of the wires (**Blue**) is connected directly to Arduino Analog pin 2 (A2). The other wire (**Red**) has one of its terminals connected to one side of the 10k resistor while the other side of the tok resistor is connected to the +ve of the breadboard. The other end of both these wires are inserted into the soil where moisture needs to be calculated.

The Green LED is connected such that its +ve terminal is connected to the Arduino's digital Pin 2 (D2) and the -ve terminal is connected to one side of the 1k resistor. The other side of the 1k resistor is connected to the -ve of the Breadboard. This Arduino - Green LED connection is shown by the **Green line** in the diagram above.

The Yellow LED is connected such that its +ve terminal is connected to the Arduino's digital Pin 3 (D3) and the -ve terminal is connected to one side of the 1k resistor. The other side of the 1k resistor is connected to the -ve of the Breadboard. This Arduino - Yellow LED connection is shown by the **Orange line** in the diagram above.

The Red LED is connected such that its +ve terminal is connected to the Arduino's digital Pin 4 (D4) and the -ve terminal is connected to one side of the 1k resistor. The other side of the 1k resistor is connected to the -ve of the Breadboard. This Arduino - Red LED connection is shown by the **Pink line** in the diagram above.

The Buzz blox is connected to the Arduino such that its +ve power pin (2nd pin from the top) is connected to the +ve of the breadboard and its -ve power pin (4th pin from the top) is connected to the -ve of the breadboard.

The Data input pin of the Buzz blox (3rd pin from the top) is connected to the Arduino's digital pin 5 (D5). This Arduino - Buzz blox connection is shown by the **Brown line** in the diagram above.

Lastly, the Arduino's 5v pin is connected to the +ve of the breadboard and the Gnd pin is connected to the -ve of the breadboard. Additionally, the USB cable connects the Arduino board to the Computer. This USB cable serves the purpose of providing power, program uploading as well as bi-directional serial data communication.

Arduino Code:

Here is the complete Arduino code for making a continuous Soil Moisture Monitoring System by taking Soil Moisture Content as an analog Input and controlling 3 LEDs and a Buzzer with an Arduino.

```
#define Green 2   // Green  for pin 2
#define Yellow 3  // Yellow for pin 3
#define Red 4     // Red    for pin 4
#define Buzzer 5  // Buzzer for pin 5
#define Soil A2   // Soil   for pin A2

int soil_value;   // for analog reading
int moisture;     // for moisture %

void setup()
{
  pinMode(Green, OUTPUT);  // Green  as Output
  pinMode(Yellow, OUTPUT); // Yellow as Output
  pinMode(Red, OUTPUT);    // Red    as Output
  pinMode(Buzzer, OUTPUT); // Buzzer as Output
  pinMode(Soil, INPUT);    // Soil   as Output
}


void loop()
{
  soil_value = analogRead(Soil);

  moisture = map(soil_value, 0, 1023, 0, 100);

if((moisture > 0) && (moisture < 25))        // 0% - 25%
    {
      digitalWrite(Green, LOW);   // Green LED Off
      digitalWrite(Yellow, LOW);  // Yellow LED Off
      digitalWrite(Red, LOW);     // Red LED Off
      digitalWrite(Buzzer, HIGH); // Buzzer On
    }

  if((moisture > 25) && (moisture < 50))     // 25% - 50%
    {
      digitalWrite(Green, HIGH);  // Green LED On
      digitalWrite(Yellow, LOW);  // Yellow LED Off
      digitalWrite(Red, LOW);     // Red LED Off
      digitalWrite(Buzzer, LOW);  // Buzzer Off
    }
```

```
  if((moisture > 50) && (moisture < 75))     // 50% - 75%
    {
      digitalWrite(Green, HIGH);  // Green LED On
      digitalWrite(Yellow, HIGH); // Yellow LED On
      digitalWrite(Red, LOW);     // Red LED Off
      digitalWrite(Buzzer, LOW);  // Buzzer Off
    }

  if((moisture > 75) && (moisture < 100))    // 75% - 100%
    {
      digitalWrite(Green, HIGH);  // Green LED On
      digitalWrite(Yellow, HIGH); // Yellow LED On
      digitalWrite(Red, HIGH);    // Red LED On
      digitalWrite(Buzzer, LOW); // Buzzer Off
    }
}
```

Explanation:

```
#define Green 2   // Green  for pin 2
#define Yellow 3  // Yellow for pin 3
#define Red 4     // Red    for pin 4
#define Buzzer 5  // Buzzer for pin 5
#define Soil A2   // Soil   for pin A2

int soil_value;   // for analog reading
int moisture;     // for moisture %
```

Here, we are declaring that from now on:

1. Arduino Digital Pin 2 (D2) will be referred to as “Green” as it is used for connecting the Green LED.

2. Arduino Digital Pin 3 (D3) will be referred to as “Yellow” as it is used for connecting the Yellow LED.

3. Arduino Digital Pin 4 (D4) will be referred to as “Red” as it is used for connecting the Red LED.

4. Arduino Digital Pin 5 (D5) will be referred to as “Buzzer” as it is used for connecting the Buzz blox.

5. Arduino Analog Pin 2 (A2) will be referred to as “Soil” as it is used for connecting the Soil Moisture sensing probe wires.

6. Next, we are taking two integer type variables, one for storing the analog value of the soil moisture probe and the other for storing the converted value in percentage moisture content in the soil.

```
void setup()
{
  pinMode(Green, OUTPUT);  // Green  as Output
  pinMode(Yellow, OUTPUT); // Yellow as Output
  pinMode(Red, OUTPUT);    // Red    as Output
  pinMode(Buzzer, OUTPUT); // Buzzer as Output
  pinMode(Soil, INPUT);    // Soil   as Output
}
```

Here, inside the setup function, we are declaring that:

1. “Green” pin will be used as an Output pin.

2. “Yellow” pin will be used as an Output pin.

3. “Red” pin will be used as an Output pin.

4. “Buzzer” pin will be used as an Output pin.

5. “Soil” pin will be used as an Input pin.

```
void loop()
{
  soil_value = analogRead(Soil);

  moisture = map(soil_value, 0, 1023, 0, 100);
```

Here, inside the loop function, we are first taking the analog reading from the inserted soil moisture probes. These values will range from 0 - 1023 as this is the Arduino's range of values for the analogRead() function.

But we want the readings in the form of percentage soil moisture. So we use the "map" function for mapping the values in the range of 0 - 1023 to values in the range of 0 - 100. Thus we are converting raw analog readings to soil moisture percentages.

```
if((moisture > 0) && (moisture < 25))        // 0% - 25%
    {
      digitalWrite(Green, LOW);   // Green LED Off
      digitalWrite(Yellow, LOW);  // Yellow LED Off
      digitalWrite(Red, LOW);     // Red LED Off
      digitalWrite(Buzzer, HIGH); // Buzzer On
    }

  if((moisture > 25) && (moisture < 50))     // 25% - 50%
    {
      digitalWrite(Green, HIGH);  // Green LED On
      digitalWrite(Yellow, LOW);  // Yellow LED Off
      digitalWrite(Red, LOW);     // Red LED Off
      digitalWrite(Buzzer, LOW);  // Buzzer Off
    }

  if((moisture > 50) && (moisture < 75))     // 50% - 75%
    {
      digitalWrite(Green, HIGH);  // Green LED On
      digitalWrite(Yellow, HIGH); // Yellow LED On
      digitalWrite(Red, LOW);     // Red LED Off
      digitalWrite(Buzzer, LOW);  // Buzzer Off
    }

  if((moisture > 75) && (moisture < 100))    // 75% - 100%
    {
      digitalWrite(Green, HIGH);  // Green LED On
      digitalWrite(Yellow, HIGH); // Yellow LED On
      digitalWrite(Red, HIGH);    // Red LED On
      digitalWrite(Buzzer, LOW); // Buzzer Off
    }
```

Here, the converted soil moisture percentage values are continuously compared against 4 different thresholds with minimum and maximum percentage range. These ranges are 0-25%, 25-50%, 50-75% and finally 75-100% soil moisture content.

The Green, the Yellow and the Red LEDs as well as the Buzzer are turned On or Off according to how much soil moisture percentage is detected for the present condition.

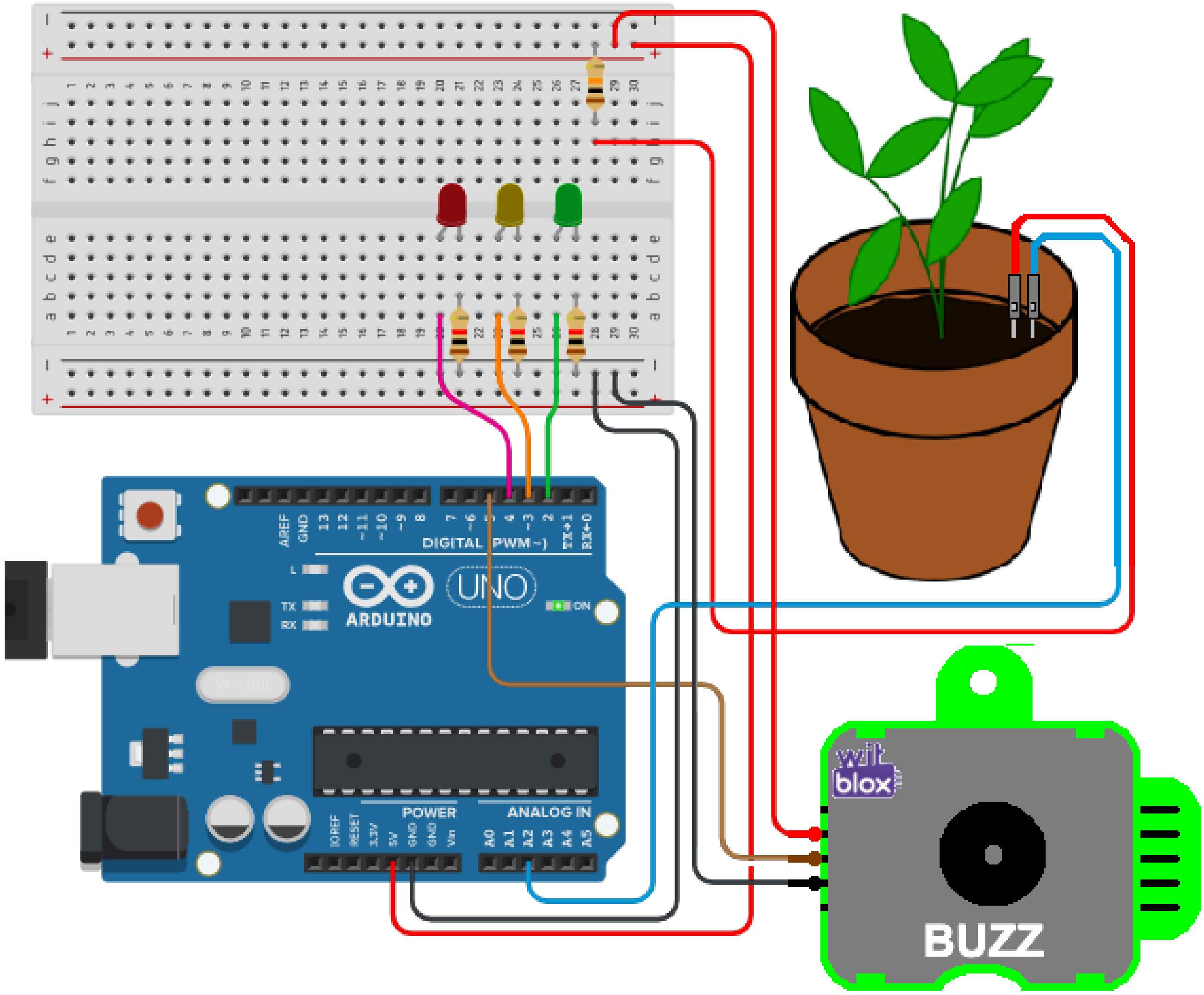
Here is how the distribution works:

1. If the measured soil moisture content is between 0% and 25%, then all LEDs are turned Off and the Buzzer is turned On, indicating the need for immediate attention and watering.

2. If the measured soil moisture content is between 25% and 50%, then only the Green LED is turned On while other LEDs are turned Off and the Buzzer is turned Off as well. So, now only 1 out of 3 LEDs are On.
3. If the measured soil moisture content is between 50% and 75%, then the Green and the Yellow LEDs are turned On and the Red LED is turned Off and the Buzzer is also turned Off. So, now 2 out of 3 LEDs are On.

4. If the measured soil moisture content is between 75% and 100%, then all the LEDs are turned On and the Buzzer is turned Off. So, now all 3 LEDs are turned On.

Outcome and Observations:



1. When the code is compiled and uploaded to the Arduino, take a plant pot with extremely dry soil and insert the metal tips of the wires into the soil as shown in the above image. Now power the system up by connecting the USB.

2. Initially all the LEDs will remain Off and the Buzzer will remain On continuously. Now start adding water to the dry soil in small quantities using a small cup.

3. After sometime the Buzzer turns Off and the Green LED turns On.

4. Keep adding more water and after sometime both the Green and the Yellow LEDs turn On.

5. Keep adding more water and after sometime all the three LEDs, Green, Yellow and Red turn On indicating that we can stop watering the soil now.