

PROJECT 6

Soil Moisture sensor using webserver

1.INTRODUCTION:-

We all have home gardens at our home. To keep the plants more healthy, we need to water the plants at perfect time at perfect range.

Using this Soil moisture sensor with the IOT Technology, You can be able to know the soil moisture and get notification through your Smart phone. When the Soil get enough moisture, you can stop watering your plants.\

Through out this Activity you can prevent your plants from excess watering and keep the Soil Moisture from falling into dry condition.

COMPONENTS:-

1. **WEMOS**
2. **SOIL MOISTURE**

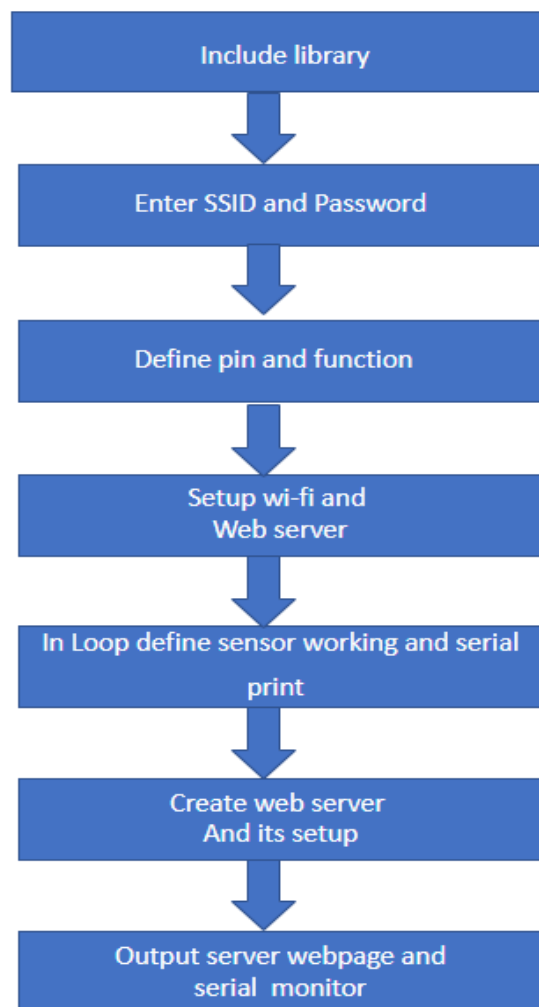
APPLICATION:-

Soil moisture sensors are used in numerous research applications, e.g. in agricultural science and horticulture including irrigation planning, climate research, or environmental science including solute transport studies and as auxiliary sensors for soil respiration measurements.

OBJECTIVES:-

Soil moisture sensors measure the water content in the soil and can be used to estimate the amount of stored water in the soil horizon. Soil moisture sensors do not measure water in the soil directly. Instead, they measure changes in some other soil property that is related to water content in a predictable way.

FLOW CHART :-



PROGRAMMING:-

```
#include <ESP8266WiFi.h>

//Connect to BSSID
const char* ssid  = "YourNetworkSSID";
const char* password = "YourNetworkPasskey";
WiFiClient client;

// Domoticz server info
const char * domoticz_server = "YourDomoticzIP";
int port = 8080; //Domoticz port
int idx = 37; //IDX for this virtual sensor, found in Setup -> Devices


const int powersoil = D8; // Digital Pin 8 will power the sensor, acting
as switch
const long interval = 2000;
float humidity = 0;
const unsigned reading_count = 10; // Numbber of readings each
time in order to stabilise
unsigned int analogVals[reading_count];
unsigned int counter = 0;
unsigned int values_avg = 0;
```

```
const unsigned sleepTimeS = 60; // Seconds to sleep between
readings

const unsigned sleepTimeMin = 15; // Minutes to sleep between
readings

int HumStat = 0; // Status Variable for humidity- It can be Dry, Wet,
Normal.

void setup() {
    // initialize serial communication at 115200 bits per second:
    Serial.begin(115200);
    delay(10);
    Serial.println();
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
```

```
pinMode(powersoil,OUTPUT);  
delay(2000);  
}
```

```
void loop() {  
  Serial.print("Starting variables.");  
  delay(500);  
  Serial.print("..  
  delay(250);  
  Serial.println("...");  
  delay(250);  
  values_avg = 0;  
  humidity = 0;  
  counter = 0;  
  
  Serial.print("Average values: ");  
  Serial.println(values_avg);  
  Serial.print("Humidity values: ");  
  Serial.println(humidity);  
  Serial.print("Counter values: ");  
  Serial.println(counter);  
  
  Serial.println("Powering module ON");  
  digitalWrite(powersoil, HIGH);
```

```
delay(1000);
// read the input on analog pin 0:
for( counter = 0; counter < reading_count; counter++){
  Serial.println("Reading probe value...:");
  analogVals[reading_count] = analogRead(A0);
  delay(100);

  values_avg = (values_avg + analogVals[reading_count]);
  Serial.println(analogVals[reading_count]);
  Serial.print("Total Readings value...:");
  Serial.println(values_avg);
}
values_avg = values_avg/reading_count;
Serial.print("Average Readings value...:");
Serial.println(values_avg);
// make average value
humidity = map(values_avg,0,400,0,100);
// print out the value the sensor reads:
Serial.print("Average humidity value...:");
Serial.println(humidity);
delay(interval);
Serial.println("Powering module off");
digitalWrite(powersoil, LOW);
delay(interval);
```

```
HumStat = HumStatFunction(humidity);  
printlnInfo();
```

```
Serial.print("Entering deep-sleep mode for Xmin*60 sec..");  
delay(1000);  
Serial.print("...");  
delay(1000);  
Serial.println("...");  
delay(1000);  
Serial.println("ESP8266 in sleep mode");  
ESP.deepSleep(sleepTimeMin * sleepTimeS * 1000000);
```

```
}
```

```
int HumStatFunction(int humidity){  
    int HumStat = 0;  
    if( humidity >40 && humidity <80){  
        HumStat = 0;//Normal Soil  
  
    }  
    else if( humidity >=0 && humidity <=40){  
        HumStat = 2;//Dry Soil  
    }  
    else if( humidity >= 80){  
        HumStat = 3;//Wet Soil
```



```

    }
    Serial.println(HumStat);

    return HumStat;

}

void printInfo()
{
    // Domoticz format
    /json.htm?type=command&param=udevice&idx=IDX&nvalue=0&svalue=HUM;HUM_STAT

    if (client.connect(domoticz_server,port)) {
        Serial.println("Sending Values to server...");
        Serial.println(HumStat);

        client.print("GET
/json.htm?type=command&param=udevice&idx=");
        client.print(idx);
        client.print("&nvalue=");
        client.print(humidity);
        client.print("&svalue=");
        client.print(HumStat);
        client.println(" HTTP/1.1");
        client.print("Host: ");
        client.print(domoticz_server);
    }
}

```

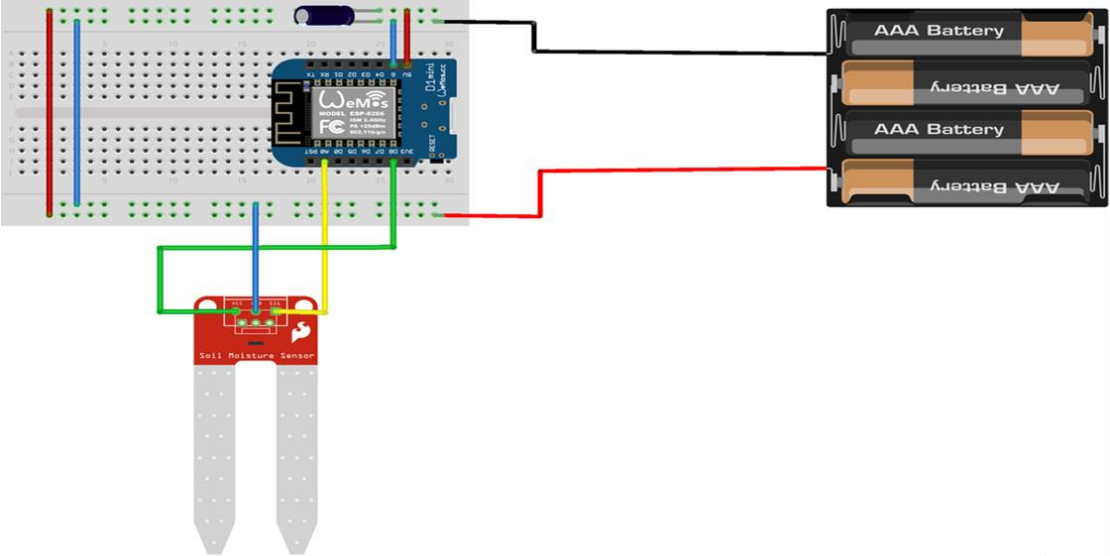
```
client.print(":");  
client.println(port);  
client.println("User-Agent: Arduino-ethernet");
```

```
client.println("Connection: close");  
client.println();  
client.stop();  
Serial.println(humidity);  
}  
}
```

HARDWARE CONNECTION:-

1. Connect pin sensor to wemos
2. Connect pin GND to GNG
3. Connect pin 5V to 5V
4. Connect pin D0 to D8

CIRCUIT DIAGRAM:-



fritzing