Bangabandhu Sheikh Mujibur Rahman
Digital University

Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh

Faculty of Engineering
Department of Information and Communication Technology
B.Sc. in Internet of Things

Course Title: Industrial IoT
Course Code: IoT 4215

PROJECT REPORT

Project Title: IoT Based Soil Moisture System

Submitted To,
Farzana Akter
Head of the Department,
Department of ICT, BDU.

Submitted By,
Group No: 04

Sumon Ahemed	Nusrat Jahan Akhi	T. M. Mehrab Hasan
ID: 1801042	ID: 1901040	ID: 1901049
Session: 2018-19	Session: 2019-20	Session: 2019-20

Date of Submission: 26 July, 2022.

Project Name: IoT Based Soil Moisture System

ACKNOWLEDGEMENT

We have taken many efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to my teammate. We are highly indebted to Farzana Akter mam for her guidance and constant supervision as well as for providing necessary information regarding the effective project titled "**IoT Based Soil Moisture System**".

ABSTRACT

Different countries use different types of soil moisture system. In our country, our farmers use typical method for soil. They face the use of unnecessary fertilizers, unnecessary water consumption, unnecessary expenses for more yields etc. As a result, they don't get expected crops, spend too much money for unnecessary fertilizers etc. This system will help them to overcome these types of problems and will also ensure good yields by maintaining the fertility of their soils.

INTRODUCTION

Purpose:

An easily accessible IoT based soil moisture system is very useful to help our farmers in their daily lives. This system measures or estimates the amount of water in the soil. This system is stationary or portables such as handheld probes. Stationary sensors are placed at the predetermined locations and depths in the field, whereas portable soil moisture probes can measure soil moisture at several locations.

Scope:

The future scope of this project is huge. In future, this system is more upgradable and developable by adding various features. We can add auto watering system, humidity & temperature monitoring system etc. By implementing those features a farmer can easily avoid the loss and get more crops.

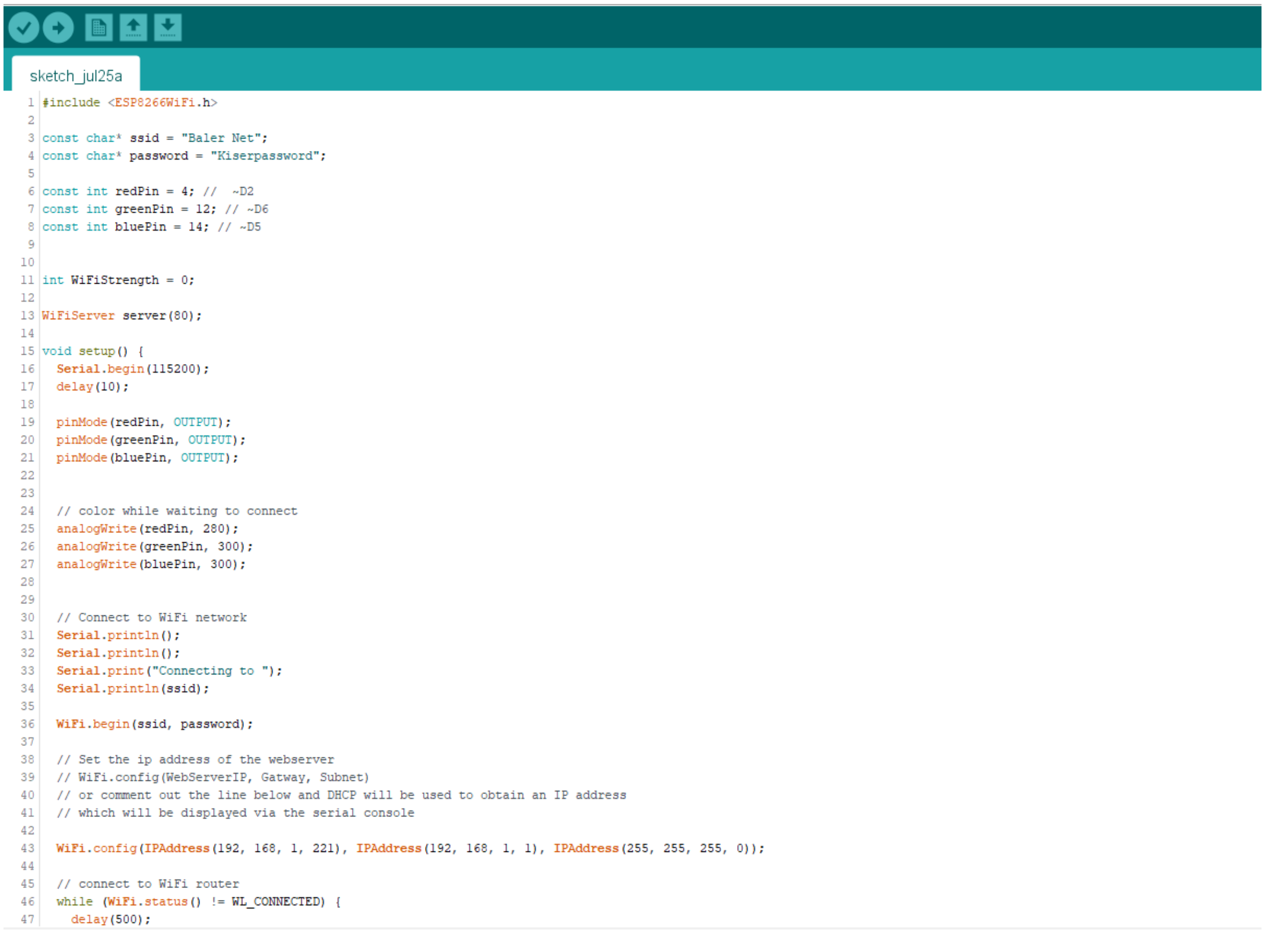
OBJECTIVE

The main objective of this project is to solve the watering issues. The other objectives are,

- Reduce water consumption
- Increase soil fertility
- Can estimate the water levels without being physically present
- Reduce human intervention
- Reduce extra manpower needed to maintain the crops
- Reduce motor power consumption

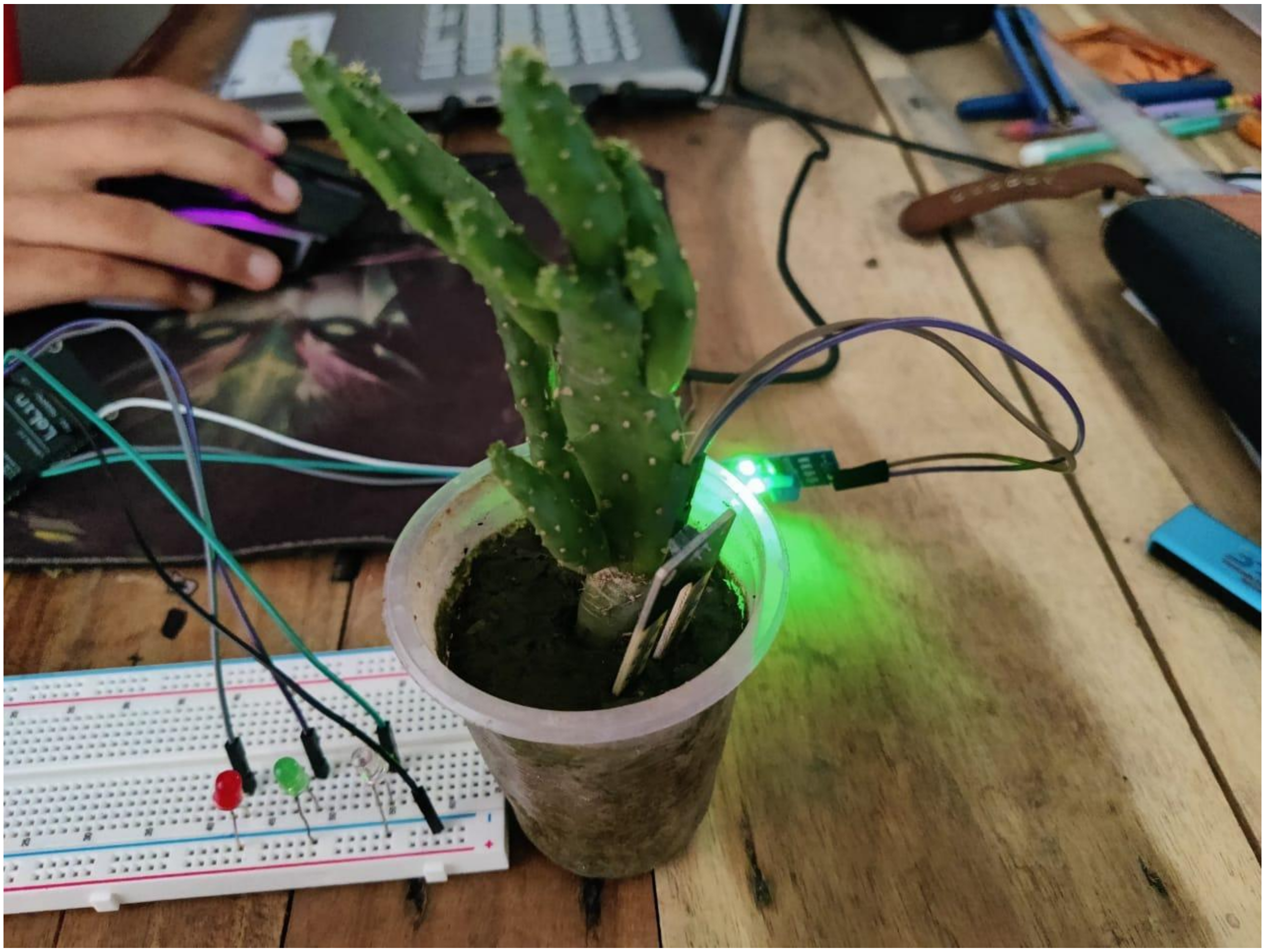
Input /Output interface:

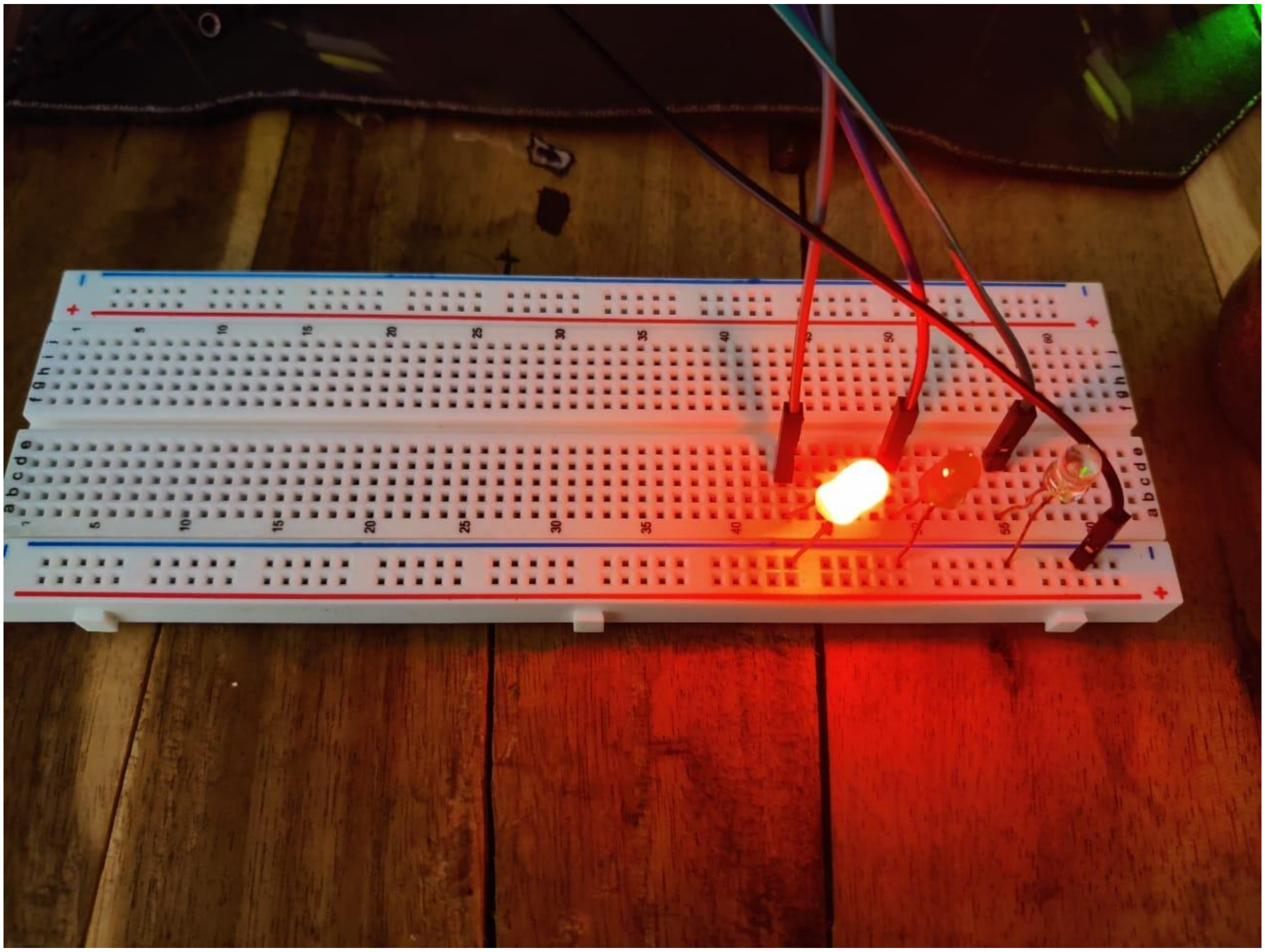
Input:

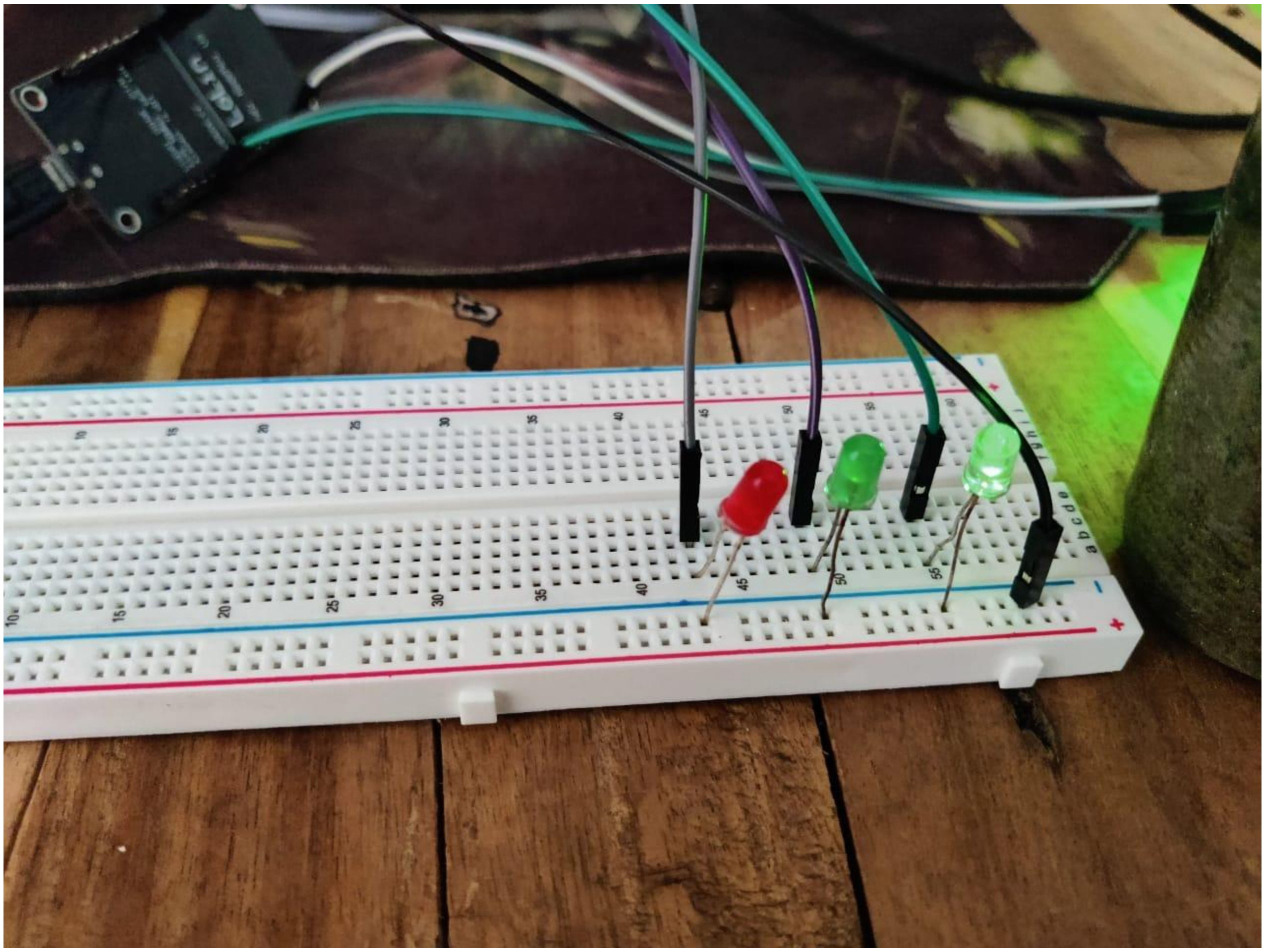


```
#include <ESP8266WiFi.h>

const char* ssid = "Baler Net";
const char* password = "Kiserpassword";

const int redPin = 4; //  ~D2
const int greenPin = 12; // ~D6
const int bluePin = 14; // ~D5


int WiFiStrength = 0;

WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);

  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);


  // color while waiting to connect
  analogWrite(redPin, 280);
  analogWrite(greenPin, 300);
  analogWrite(bluePin, 300);


  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  // Set the ip address of the webserver
  // WiFi.config(WebServerIP, Gatway, Subnet)
  // or comment out the line below and DHCP will be used to obtain an IP address
  // which will be displayed via the serial console

  WiFi.config(IPAddress(192, 168, 1, 221), IPAddress(192, 168, 1, 1), IPAddress(255, 255, 255, 0));

  // connect to WiFi router
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
```

Output:







Code Implementation

Implementation Environment

Challenges identified for the successful design and implementation of this project are dominated by: Complexity, reliability/availability, transparent data access. The project was a result of a Group consensus. The team was having two members. A Session is maintained throughout the system when a particular user enters the system. The Session is regularly checked whenever it is required. Proper validation is placed as and when it is required.

Limitations and Future Enhancement

LIMITATIONS:

The main limitation of this system is that this project is now at initial stage, and it is implemented in a small environment. In future, this project will become complex when it will be implemented for bigger environment.

Another limitation will be the use of Wi-Fi. If the wi-fi signal strength is low than we will get slightly different from the accurate value.

FUTURE SCOPE OF THE PROJECT:

Our project will be able to implement in the future after making some changes and modifications as we make our project at a very low level. So, the modifications that can be done in our project are: In the future one change can be done by adding the auto watering system based on soil moisture. And one more major change which can be done in this project is to add a notification system in mobile device.

Conclusion

So, this project is used to monitor soil moisture under wheat crop cultivation practices for using intelligent irrigation system. The tensiometers and Watermarks were less responsive to the soil drying between irrigations than regular used methods.

References:

- https://www.instructables.com/ESP8266-Soil-Moisture-Sensor-With-Arduino-IDE/

- https://create.arduino.cc/projecthub/hubmartin/soil-sensor-with-arduino-esp8266-esp32-2bc942

- https://diyi0t.com/soil-moisture-sensor-tutorial-for-arduino-and-esp8266/

Appendix

Source Code:

```
#include <ESP8266WiFi.h>
const char* ssid = "Baler Net";
const char* password = "Kiserpassword";
const int redPin = 4; //  ~D2
const int greenPin = 12; // ~D6
const int bluePin = 14; // ~D5
int WiFiStrength = 0;
WiFiServer server(80);
void setup() {
Serial.begin(115200);
 delay(10);
```

```
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
 // color while waiting to connect
  analogWrite(redPin, 280);
  analogWrite(greenPin, 300);
  analogWrite(bluePin, 300);
 // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
 WiFi.begin(ssid, password);
// Set the ip address of the webserver
  // WiFi.config(WebServerIP, Gatway, Subnet)
  // or comment out the line below and DHCP will be used to obtain an IP address
  // which will be displayed via the serial console
 WiFi.config(IPAddress(192, 168, 1, 221), IPAddress(192, 168, 1, 1), IPAddress(255, 255, 255, 0));
 // connect to WiFi router
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
Serial.println("");
  Serial.println("WiFi connected");
// Start the server
  server.begin();
  Serial.println("Server started");
 // Print the IP address
  Serial.print("Use this URL to connect: ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");

}
double analogValue = 0.0;
double analogVolts = 0.0;
unsigned long timeHolder = 0;
void loop() {
  WiFiStrength = WiFi.RSSI(); // get dBm from the ESP8266
  analogValue = analogRead(A0); // read the analog signal

  analogVolts = (analogValue * 3.08) / 1024;
  int chartValue = (analogValue * 100) / 400;
  chartValue = 100 - chartValue;
  if (millis() - 15000 > timeHolder)
  {
    timeHolder = millis();
 if (chartValue <= 25) {  // 0-25 is red "dry"
     analogWrite(redPin, 1000);
      analogWrite(greenPin, 0);
      analogWrite(bluePin, 0);

    } else if (chartValue > 25 && chartValue <= 75) // 26-75 is green
    {
     analogWrite(redPin, 0);
      analogWrite(greenPin, 1000);
      analogWrite(bluePin, 0);
  }
    else if (chartValue > 75 ) // 76-100 is blue
    {
     analogWrite(redPin, 0);
      analogWrite(greenPin, 0);
      analogWrite(bluePin, 1000);

    }

    delay(1000); // this is the duration the LED will stay ON

    analogWrite(redPin, 0);
    analogWrite(greenPin, 0);
    analogWrite(bluePin, 0);

  }

  // Serial data
  Serial.print("Moisture Level: ");
  Serial.println(analogValue);
  Serial.print("Analog V: ");
  Serial.println(analogVolts);
  Serial.print("ChartValue: ");
  Serial.println(chartValue);
  Serial.print("TimeHolder: ");
  Serial.println(timeHolder);
  Serial.print("millis(): ");
  Serial.println(millis());
  Serial.print("WiFi Strength: ");
  Serial.print(WiFiStrength); Serial.println("dBm");
```

```
 Serial.println(" ");
 delay(1000); // slows amount of data sent via serial

 // check to for any web server requests. ie - browser requesting a page from the webserver
 WiFiClient client = server.available();
 if (!client) {
   return;
 }
// Wait until the client sends some data
 Serial.println("new client");

 // Read the first line of the request
 String request = client.readStringUntil('\r');
 Serial.println(request);
 client.flush();

 // Return the response
 client.println("HTTP/1.1 200 OK");
 client.println("Content-Type: text/html");
 client.println(""); //  do not forget this one
 client.println("<!DOCTYPE HTML>");

 client.println("<html>");
 client.println(" <head>");
 client.println("<meta http-equiv=\"refresh\" content=\"60\">");
 client.println(" <script type=\"text/javascript\"
src=\"https://www.gstatic.com/charts/loader.js\"></script>");
 client.println("  <script type=\"text/javascript\">");
 client.println("    google.charts.load('current', {'packages':['gauge']});");
 client.println("    google.charts.setOnLoadCallback(drawChart);");
 client.println("   function drawChart() {");

 client.println("      var data = google.visualization.arrayToDataTable([ ");
 client.println("        ['Label', 'Value'], ");
 client.print("       ['Moisture',  ");
 client.print(chartValue);
 client.println(" ], ");
 client.println("       ]); ");
 // setup the google chart options here
 client.println("    var options = {");
 client.println("      width: 400, height: 120,");
 client.println("      redFrom: 0, redTo: 25,");
 client.println("      yellowFrom: 25, yellowTo: 75,");
 client.println("      greenFrom: 75, greenTo: 100,");
 client.println("       minorTicks: 5");
 client.println("    };");

 client.println("   var chart = new
google.visualization.Gauge(document.getElementById('chart_div'));");
 client.println("  chart.draw(data, options);");
 client.println("  setInterval(function() {");
 client.print("  data.setValue(0, 1, ");
 client.print(chartValue);
 client.println("    );");
 client.println("    chart.draw(data, options);");
 client.println("    }, 13000);");

client.println("  }");
 client.println(" </script>");
```

```
  client.println("  </head>");
  client.println("  <body>");

  client.print("<h1 style=\"size:12px;\">ESP8266 Soil Moisture</h1>");

  // show some data on the webpage and the guage
  client.println("<table><tr><td>");

  client.print("WiFi Signal Strength: ");
  client.println(WiFiStrength);
  client.println("dBm<br>");
  client.print("Analog Raw: ");
  client.println(analogValue);
  client.print("<br>Analog Volts: ");
  client.println(analogVolts);
  client.println("<br><a href=\"/REFRESH\"\"><button>Refresh</button></a>");

  client.println("</td><td>");
  // below is the google chart html
  client.println("<div id=\"chart_div\" style=\"width: 300px; height: 120px;\"></div>");
  client.println("</td></tr></table>");

  client.println("<body>");
  client.println("</html>");
  delay(1);
  Serial.println("Client disonnected");
  Serial.println("");


}
```