

# **GOVERNMENT COLLEGE OF ENGINEERING**

## **ERODE**

(FORMERLY KNOWN AS IRTT)

### **BE., ELECTRONICS AND COMMUNICATION ENGINEERING**

**TECHNOLOGY NAME : INTERNET OF THINGS**

#### **ENVIRONMENTAL MONITORING**

#### **AUTOMATIC SOIL IRRIGATION SYSTEM**

**NAME OF THE STUDENTS**

**REGISTER NUMBER**

**TEAM LEADER:**

P.DINESHKUMAR

731121106010

**TEAM MEMBERS:**

- |                    |              |
|--------------------|--------------|
| 1. V.JANANI        | 731121106018 |
| 2. M.DEVAKI        | 731121106004 |
| 3. S.DHANALAKSHIMI | 731121106005 |
| 4. P.JEEVITHA      | 731121106306 |

## **PHASE 3**

Under the mentor of

**DR.M.POONGOTHAI ME, Phd.,**  
**DEPARTMENT OF INFORMATION TECHNOLOGY(IT)**

Government College of Engineering, Erode

Near Vasavi College Post,

TamilNadu – 638316.

Affiliated to Anna University , Chennai.

# **AUTOMATIC SOIL IRRIGATION SYSTEM**

## **INTRODUCTION:**

Automatic soil irrigation is the use of a device to operate irrigation structures so the change of flow of water from bays can occur in the absence of the irrigator.

Automation can be used in a number of ways:

- To start and stop irrigation through supply channel outlets
- To start and stop pumps
- To cut off the flow of water from one irrigation area either a bay or a section of channel and directing the water to another area.

These changes occur automatically without any direct manual effort, but you may need to spend time preparing the system at the start of the irrigation and maintaining the components so it works properly.

## **PYTHON PROGRAM:**

```
#include <ESP8266WiFi.h>

#include <SPI.h>

#include <Wire.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include <DHT.h> // Including library for dht

#include <Fonts/FreeSerif9pt7b.h> // Add a custom font

#include <Fonts/FreeMono9pt7b.h> // Add a custom font

#define SCREEN_WIDTH 128 // OLED display width, in pixels

#define SCREEN_HEIGHT 64 // OLED display height, in pixels

#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)
```

```

#define DHTPIN D4      //pin where the dht11 is connected

DHT dht(DHTPIN, DHT11);


String apiKey = "*****"; // Enter your Write API key from ThingSpeak

const char *ssid = "*****"; // replace with your wifi ssid and wpa2 key

const char *pass = "*****";

const char* server = "api.thingspeak.com";


Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);


const int AirValue = 790; //you need to replace this value with Value_1
const int WaterValue = 390; //you need to replace this value with Value_2

const int SensorPin = A0;

int soilMoistureValue = 0;

int soilmoisturepercent=0;

int relaypin = D5;


WiFiClient client;


void setup() {

  Serial.begin(115200); // open serial port, set the baud rate to 9600 bps

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); //initialize with the I2C addr 0x3C
  (128x64)

  display.clearDisplay();

```

```
pinMode(relaypin, OUTPUT);
```

```
dht.begin();
```

```
WiFi.begin(ssid, pass);
```

```
while (WiFi.status() != WL_CONNECTED)
```

```
{
```

```
    delay(500);
```

```
    Serial.print(".");
```

```
}
```

```
    Serial.println("");
```

```
    Serial.println("WiFi connected");
```

```
    delay(4000);
```

```
}
```

```
void loop()
```

```
{
```

```
    float h = dht.readHumidity();
```

```
    float t = dht.readTemperature();
```

```
    Serial.print("Humidity: ");
```

```
    Serial.print(h,0);
```

```
    Serial.println("%");
```

```
Serial.print("Temperature: ");

Serial.print(t,0);

Serial.println("% ");

delay(100);

soilMoistureValue = analogRead(SensorPin); //put Sensor insert into soil

//Serial.println(soilMoistureValue);


soilmoisturepercent = map(soilMoistureValue, AirValue, WaterValue, 0, 100);


if(soilmoisturepercent >= 100)
{
    Serial.print("soilmoisture: ");
    Serial.println("100% ");

    display.setCursor(0,20); //oled display
    display.setTextSize(0);
    display.setFont(&FreeMono9pt7b); // Set a custom font
    display.setTextColor(WHITE);
    display.print("Moist:");
    display.setTextSize(0);
    display.print("100");
    display.println("% ");
    display.setCursor(0,40); //oled display
    display.setTextSize(0);
```

```
display.print("Humid:");
display.setTextSize(0);
display.print(h,0);
display.println("% ");
display.setCursor(0,60); //oled display
display.setTextSize(0);
display.print("Temp:");
display.setTextSize(0);
display.print(t,0);
display.println("*C");
display.display();

delay(100);
display.clearDisplay();
}

else if(soilmoisturepercent <=0)
{
  Serial.print("soilmoisture: ");
  Serial.println("0% ");

  display.setCursor(0,20); //oled display
  display.setTextSize(0);
  display.setFont(&FreeMono9pt7b); // Set a custom font
```

```

display.setTextColor(WHITE);
display.print("Moist:");
display.setTextSize(0);
display.print("0");
display.println("%");
display.setCursor(0,40); //oled display
display.setTextSize(0);
display.print("Humid:");
display.setTextSize(0);
display.print(h,0);
display.println("%");
display.setCursor(0,60); //oled display
display.setTextSize(0);
display.print("Temp:");
display.setTextSize(0);
display.print(t,0);
display.println("*C");
display.display();

delay(100);
display.clearDisplay();
}

```

```

else if(soilmoisturepercent >=0 && soilmoisturepercent <= 100)

```



```
{  
  
  Serial.print("soilmoisture : ");  
  
  Serial.print(soilmoisturepercent);  
  
  Serial.println("% ");  
  
  
  display.setCursor(0,20); //oled display  
  
  display.setTextSize(0);  
  
  display.setFont(&FreeMono9pt7b); // Set a custom font  
  
  display.setTextColor(WHITE);  
  
  display.print("Moist:");  
  
  display.setTextSize(0);  
  
  display.print(soilmoisturepercent);  
  
  display.println("% ");  
  
  display.setCursor(0,40); //oled display  
  
  display.setTextSize(0);  
  
  display.print("Humid:");  
  
  display.setTextSize(0);  
  
  display.print(h,0);  
  
  display.println("% ");  
  
  display.setCursor(0,60); //oled display  
  
  display.setTextSize(0);  
  
  display.print("Temp:");  
  
  display.setTextSize(0);  
  
  display.print(t,0);  
  
  display.println("*C");
```

```
display.display();
```

```
delay(100);
```

```
display.clearDisplay();
```

```
}
```

```
if(soilmoisturepercent <= 49)
```

```
{
```

```
    digitalWrite(relaypin, LOW);
```

```
    Serial.println("Motor is ON");
```

```
    Serial.println("-----");
```

```
}
```

```
else if (soilmoisturepercent >49 && soilmoisturepercent <= 100)
```

```
{
```

```
    digitalWrite(relaypin, HIGH);
```

```
    Serial.println("Motor is OFF");
```

```
    Serial.println("-----");
```

```
}
```

```
if (client.connect(server, 80)) // "184.106.153.149" or api.thingspeak.com
```

```
{
```

```
    String postStr = apiKey;
```

```
    postStr += "&field1=";
```

```
    postStr += String(soilmoisturepercent);
```

```
    postStr += "&field2=";
```

```
    postStr += String(h);
    postStr += "&field3=";
    postStr += String(t);
    postStr += "&field4=";
    postStr += String(relaypin);
    postStr += "\r\n\r\n\r\n\r\n";

    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(postStr.length());
    client.print("\n\n");
    client.print(postStr);

}

client.stop();
}
```

## **PROGRAM DISCRIPTION:**

- This code appears to be controlling a motor based on the soil moisture level. It also sends the soil moisture level, temperature, and humidity data to a Thingspeak server.
- In the setup() function, the program initializes the serial communication and sets up the motor pin as an output. It also connects to the Wi-Fi network and initializes the Thingspeak client.
- In the loop() function, the program reads the soil moisture level, temperature, and humidity values. It then checks the soil moisture level and turns on or off the motor accordingly. It also creates a string with the data to be sent to the Thingspeak server and sends an HTTP POST request to update the fields.
- The code snippet at the end of the loop() function stops the Thingspeak client after sending the data.
- Overall, this code controls a motor based on the soil moisture level and sends data to a Thingspeak server.