

# IOT Project Codes and Video link

## Smart farming System

Moiz k.(19BEE0165)

Sarthak Mittal(19BEE0107)

Parth Sharma(19BEE0392)

### **Code in Arduino:**

```
#include <DHT.h>

#define DHTTYPE DHT11

#include <ESP8266WiFi.h>

#include <ESPAsyncTCP.h>

#include <ESPAsyncWebServer.h>

#include <FS.h>

#include <Wire.h>


char* ssid = "Muffadal villa"; //WIFI SSID

const char* password = "7d08c10c17974bc"; //WIFI PASSWORD

int Relay1 = 12; //D6 light switch

int Relay2 = 13; //D7 motor switch

const int ldr_pin = 5; //D1

int lights=0;

float h =0;

float t =0;

int soil = A0;

int val=0;


int dht = 2;

DHT DHT(dht, DHTTYPE);

AsyncWebServer server(80);


String getHum() {

    delay(500);

    float h = DHT.readHumidity();

    Serial.print("Humidity:");

    Serial.println(h);
```

```
    delay(500);

    return String(h);
}

String getTem() {
    delay(500);

    float t = DHT.readTemperature();

    Serial.print("Temperature:");

    Serial.println(t);

    delay(500);

    if(t>35){
        //digitalWrite(Relay2,HIGH);
    }
    else{
        //digitalWrite(Relay2,LOW);
    }

    return String(t);
}
```

```
String getLdr(){
    if( digitalRead( ldr_pin ) == 1){
        //Serial.println("Lights ON");

        lights=1;

        digitalWrite(Relay1,LOW);
    }
    else{
        //Serial.println("Lights OFF");

        lights=0;

        digitalWrite(Relay1,HIGH);
    }

    //Serial.println(lights);

    return String(lights);

    delay(500);
}
```

```
String getMoisture(){
```

```

val = analogRead(soil);

Serial.print("Analog Value : ");

int moisture=map(val,1024,500,0,100);

//Serial.println(moisture);

if(moisture<40){

  Serial.print("Motor on");

  Serial.println("("+String(moisture)+"%");

  digitalWrite(Relay2,LOW);

}

else if(moisture>70){

  Serial.print("Too much water(Motor off)");

  Serial.println("("+String(moisture)+"%");

  digitalWrite(Relay2,HIGH);

}

else{

  Serial.print("Good amount");

  Serial.println("("+String(moisture)+"%");

  digitalWrite(Relay2,LOW);

}

return String(moisture);

delay(500);

}

void setup () {

  Serial.begin (9600);

  /* DHT.begin();*/

  pinMode(ldr_pin,INPUT);

  pinMode(Relay1, OUTPUT);

  pinMode(Relay2, OUTPUT);

  if (!SPIFFS.begin ()) {

    Serial.println ("An Error has occurred while mounting SPIFFS");

    return;

  }

  // Connect to Wi-Fi

  WiFi.begin(ssid, password);

```

```
while (WiFi.status() != WL_CONNECTED) {  
    delay(1000);  
    Serial.print(".");  
}  
  
// Print ESP32 Local IP Address  
Serial.println(WiFi.localIP());  
  
// Route for web page  
server.on ("/", HTTP_GET, [] (AsyncWebServerRequest * request) {  
    request-> send (SPIFFS, "/index.html");  
});  
  
server.on ("/temp", HTTP_GET, [] (AsyncWebServerRequest * request) {  
    request-> send_P (200, "text / plain", getTem().c_str());  
});  
server.on ("/hum", HTTP_GET, [] (AsyncWebServerRequest * request) {  
    request-> send_P (200, "text / plain", getHum().c_str());  
});  
  
server.on ("/lights", HTTP_GET, [] (AsyncWebServerRequest * request) {  
    request-> send_P (200, "text / plain", getLdr().c_str());  
});  
  
server.on ("/moisture", HTTP_GET, [] (AsyncWebServerRequest * request) {  
    request-> send_P (200, "text / plain", getMoisture().c_str());  
});  
  
// start server  
server.begin ();  
}  
void loop() {  
}
```

## Code for Chart local webpage:

```
<html>

<head>

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <script src="https://code.highcharts.com/highcharts.js"></script>

  <style>

    body {

      min-width: 310px;

      max-width: 800px;

      height: 400px;

      margin: 0 auto;

    }

    h2 {

      font-family: Arial;

      font-size: 2.5rem;

      text-align: center;

    }

  </style>

</head>

<body>

  <h2>Plots of Sensors</h2>

  <div id="chart-DHT11" class="container"></div>

  <div id="chart-light" class="container"></div>

  <div id="chart-moisture" class="container"></div>

</body>


<script>

var chartTH = new Highcharts.Chart({

  chart:{ renderTo : 'chart-DHT11' },

  title: { text: 'DHT11 data' },

  series: [{

    name:'Temperature',
```

```

data:[],
    tooltip:{valueSuffix:'\xB0C'}
},{
    name:'Humidity',
    data:[],
    tooltip:{valueSuffix:'%'}
}
],
plotOptions: {
    line: { animation: false,
        dataLabels: { enabled: true }
    },
    series: [{ color: '#059e8a' }, {color:'#83hr93'}]
},
xAxis: { type: 'datetime',
    dateTimeLabelFormats: { second: '%H:%M:%S' }
},
yAxis: {
    title: { text: 'DHT11 data' }
},
credits: { enabled: false }
});

setInterval(function ( ) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            var x = (new Date()).getTime(),
                y = parseFloat(this.responseText);
            //console.log(this.responseText);
            if(chartTH.series[0].data.length > 40) {
                chartTH.series[0].addPoint([x, y], true, true, true);
            } else {
                chartTH.series[0].addPoint([x, y], true, false, true);
            }
        }
    }
}

```

```

    }
};

xhttp.open("GET", "/temp", true);
xhttp.send();

var xhttp1 = new XMLHttpRequest();
xhttp1.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        var x = (new Date()).getTime(),
            y = parseFloat(this.responseText);
        //console.log(this.responseText);
        if(chartTH.series[1].data.length > 40) {
            chartTH.series[1].addPoint([x, y], true, true, true);
        } else {
            chartTH.series[1].addPoint([x, y], true, false, true);
        }
    }
};

xhttp1.open("GET", "/hum", true);
xhttp1.send();
}, 1000 ) ;

```

```

var chartL = new Highcharts.Chart({
    chart:{ renderTo : 'chart-light' },
    title: { text: 'Lights Status' },
    series: [{
        showInLegend: false,
        data: [],
        tooltip:{
            formatter:function(){
                if(this.point.y == 1)
                    return 'ON'
                else

```

```

        return 'OFF'
    }
}

}},
plotOptions: {
    line: { animation: false,
        dataLabels: { enabled: true }
    },
    series: { color: '#059e8a' }
},
xAxis: { type: 'datetime',
    dateTimeLabelFormats: { second: '%H:%M:%S' }
},
yAxis: {
    title: { text: 'Light Status' }
},
credits: { enabled: false }
});

setInterval(function ( ) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            var x = (new Date()).getTime(),
                y = parseFloat(this.responseText);
            //console.log(this.responseText);
            if(chartL.series[0].data.length > 40) {
                chartL.series[0].addPoint([x, y], true, true, true);
            } else {
                chartL.series[0].addPoint([x, y], true, false, true);
            }
        }
    };
    xhttp.open("GET", "/lights", true);
    xhttp.send();

```



```
}, 1000 ) ;
```

```
var chartM = new Highcharts.Chart({  
    chart:{ renderTo : 'chart-moisture' },  
    title: { text: 'Moisture level' },  
    series: [{  
        showInLegend: false,  
        data: []  
    }],  
    plotOptions: {  
        line: { animation: false,  
            dataLabels: { enabled: true }  
        },  
        series: { color: '#059e8a' }  
    },  
    xAxis: { type: 'datetime',  
        dateTimeLabelFormats: { second: '%H:%M:%S' }  
    },  
    yAxis: {  
        title: { text: 'Moisture' }  
    },  
    credits: { enabled: false }  
});  
  
setInterval(function ( ) {  
    var xhttp = new XMLHttpRequest();  
    xhttp.onreadystatechange = function() {  
        if (this.readyState == 4 && this.status == 200) {  
            var x = (new Date()).getTime(),  
                y = parseFloat(this.responseText);  
            //console.log(this.responseText);  
            if(chartM.series[0].data.length > 40) {  
                chartM.series[0].addPoint([x, y], true, true, true);  
            } else {  
                chartM.series[0].addPoint([x, y], true, false, true);  
            }  
        }  
    }  
};
```

```
}  
  
}  
  
};  
  
xhttp.open("GET", "/moisture", true);  
xhttp.send();  
  
, 1000 ) ;  
</script>  
</html>
```

**Video Link:**

<https://drive.google.com/folderview?id=187FCOdMxcBei6KEc-DVgALZmPAPmD9bp>