runlinc Project 104 I11: Temperature and humidity Graph (E32W Version)

Contents

Introduction

Problem

Temperature and humidity are two important parameters affecting the growth of crops. We want to keep track on the current temperature and humidity of environment around the crops. And We want to monitor the effect of temperature and humidity on the growth of crops.

Background

High temperatures affect plant growth in numerous ways. The most obvious are the effects of heat on photosynthesis, in which plants use carbon dioxide to produce oxygen, and respiration, an opposite process in which plants use oxygen to produce carbon dioxide. Experts at Colorado State University Extension explain that both processes increase when temperatures rise. However, when temperatures reach uncomfortably high limits (which depends on the plant), the two processes become unbalanced. Tomatoes, for example, get into trouble when temperatures exceed about 96 degrees F. (36 C.). The effect of temperature on plants varies widely and is influenced by factors such as exposure to sunlight, moisture drainage, elevation, difference between day and night temperatures, and proximity to surrounding rock structure (thermal heat mass).

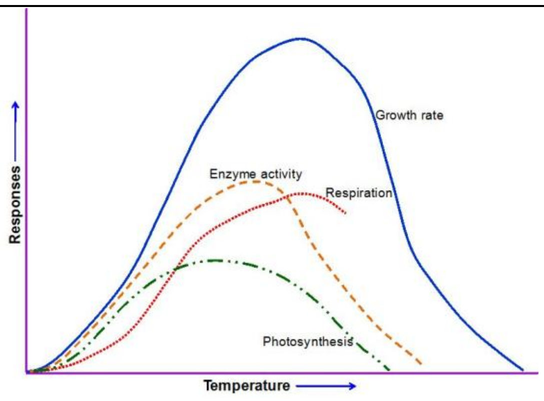


Figure 1 Temperature effects on plant growth

Humidity is the amount of water vapor in the air. Relative humidity is given as a percentage and tells you how close the air is to being saturated.

However, relative humidity is dependent on air temperature, too. If the water vapor content stays the same and the temperature drops, the relative humidity increases. If the water vapor content stays the same and the temperature rises, the relative humidity decreases. This is because colder air doesn't require as much moisture to become saturated as warmer air.

If the humidity is high, this moisture doesn't evaporate from your body very fast. In areas that are very dry, such as Arizona, the humidity is so low that when you sweat, the water evaporates so quickly that you may not even feel it. In this case, you must be careful to stay hydrated because the water loss goes unnoticed.

Humidity can also affect plant turgor pressure, which is an indicator of the amount of water in plant cells. When humidity is low, and dew points are in the 50s and low 60s, moisture evaporates from plants very quickly. When this happens, plants can wilt rapidly if too much water is pulled out of plant cells through transpiration. Conversely, when humidity and temperature are both high, plants can get overheated because transpiration is reduced, thus restricting evaporative cooling.

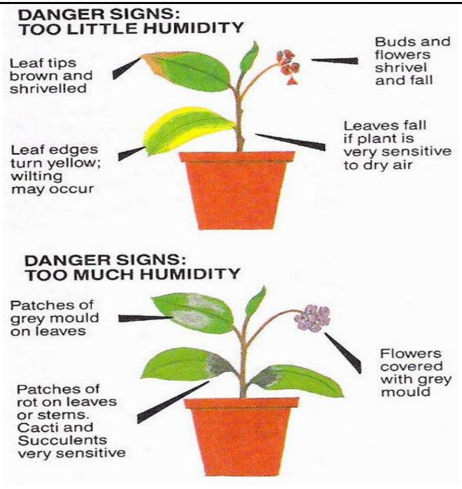

DANGER SIGNS:
TOO LITTLE HUMIDITY
Leaf tips brown and shrivelled
Buds and flowers shrivel and fall
Leaves fall if plant is very sensitive to dry air
Leaf edges turn yellow; wilting may occur
DANGER SIGNS:
TOO MUCH HUMIDITY
Patches of grey mould on leaves
Flowers covered with grey mould
Patches of rot on leaves or stems. Cacti and Succulents very sensitive

Figure 2 Humidity effects on plants (Image from Cool Garden)

Ideas

How can we measure the temperature and humidity? How can the sensor measure the real-time temperature and humidity and display the data on runlinc webpage?

Plan

We have DHT11–temperature and humidity sensor in our kits which we can use to check the temperature and humidity around the sensor. We want to use the DHT11 sensor to monitor the temperature and humidity around the crops.

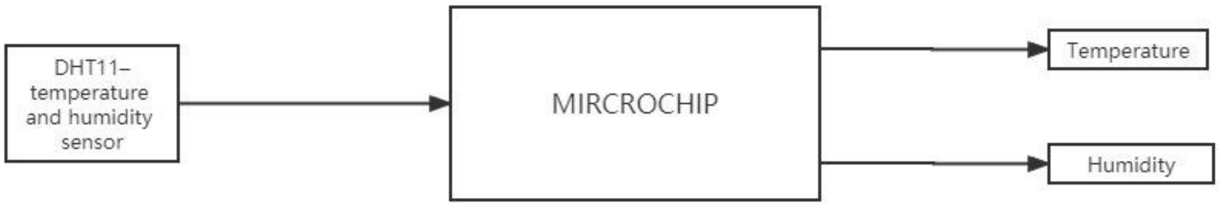


Figure 3: Block diagram of Microchip outputs

runlinc Background

runlinc is a web page inside a Wi-Fi chip. The programming is done inside the browsers compare to programming inside a chip. The runlinc web page inside the Wi-Fi chip will command the microchips to do sensing, control, data logging Internet of Things (IoT). It can predict and command.

Part A: Design the Circuit on runlinc

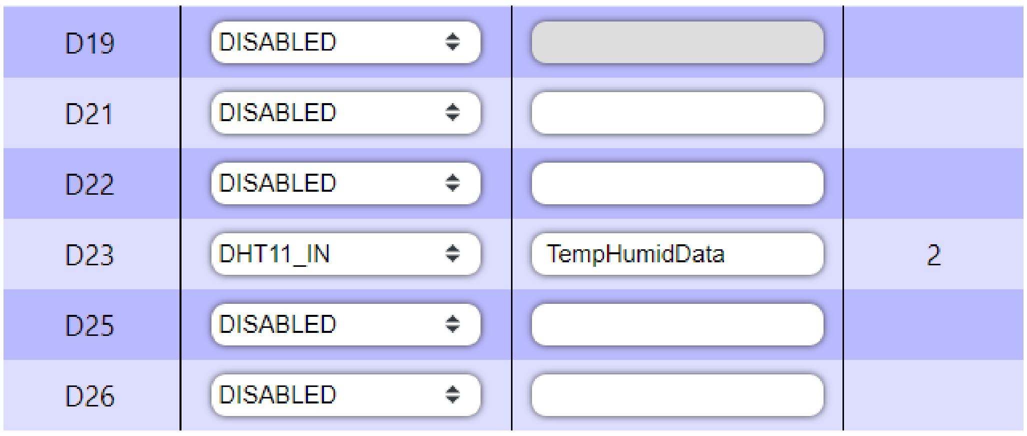
Note: Refer to runlinc Wi-Fi Setup Guide document to connect to runlinc

Upgrade your runlinc V1.2 to V2.0 with reference to the file 'runlinc V2.0 Upgrade'.

Use the left side of the runlinc web page to construct an input/output (I/O).

For port D23 name it TempHumidData and set it as DHT11_IN.

In our circuit design, we will be using the DHT11–temperature and humidity sensor. We happen to have this in our kits, so these can be used on our circuit design, as per the plan.



D19	DISABLED		
D21	DISABLED		
D22	DISABLED		
D23	DHT11_IN	TempHumidData	2
D25	DISABLED		
D26	DISABLED		

Figure 4: I/O configurations connections

Part B: Build the Circuit

Use the STEMSEL E32W board to connect the hardware. For this project we are using both the left and right I/O ports, with **negative port (-ve)** on the outer side, **positive port (+ve)** on the middle and **signal port (s)** on the inner side (as shown below).

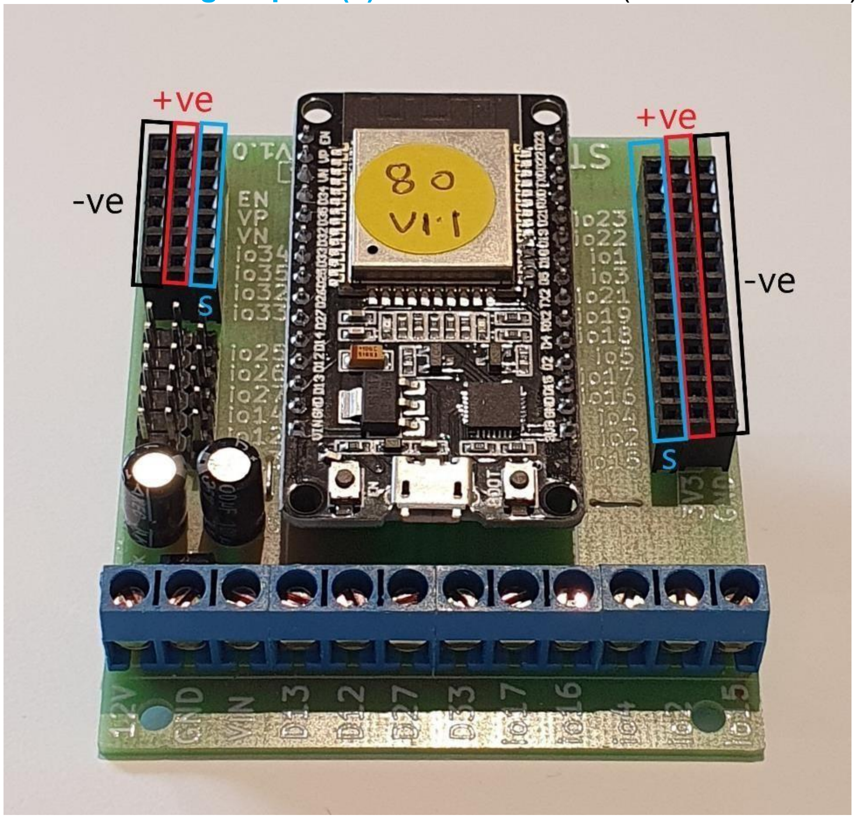


Figure 5: Negative, Positive and Signal port on the E32W board

There is one I/O part we are using for this project, a DHT11–temperature and humidity sensor, their respective pins are shown in the figure below.

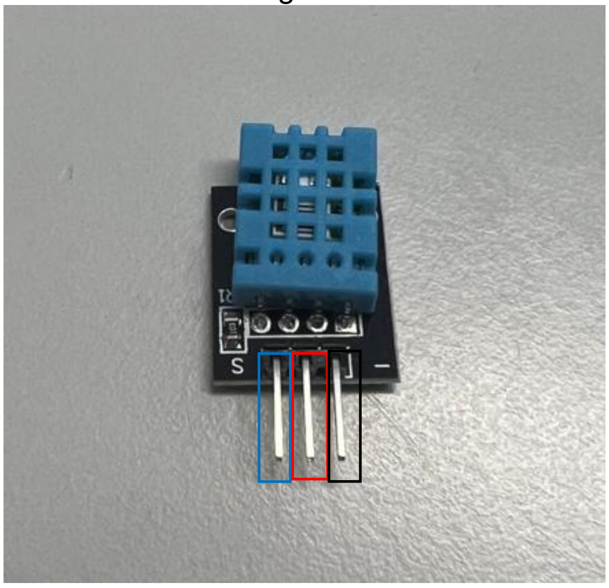


Figure 6: I/O part with negative, positive and signal pins indicated

Wiring instructions

a) Plug in the DHT11–temperature and humidity sensor to io23 on the E32W board.

b) Make sure the (-ve) pin are on the GND (outer) side of the I/O ports.

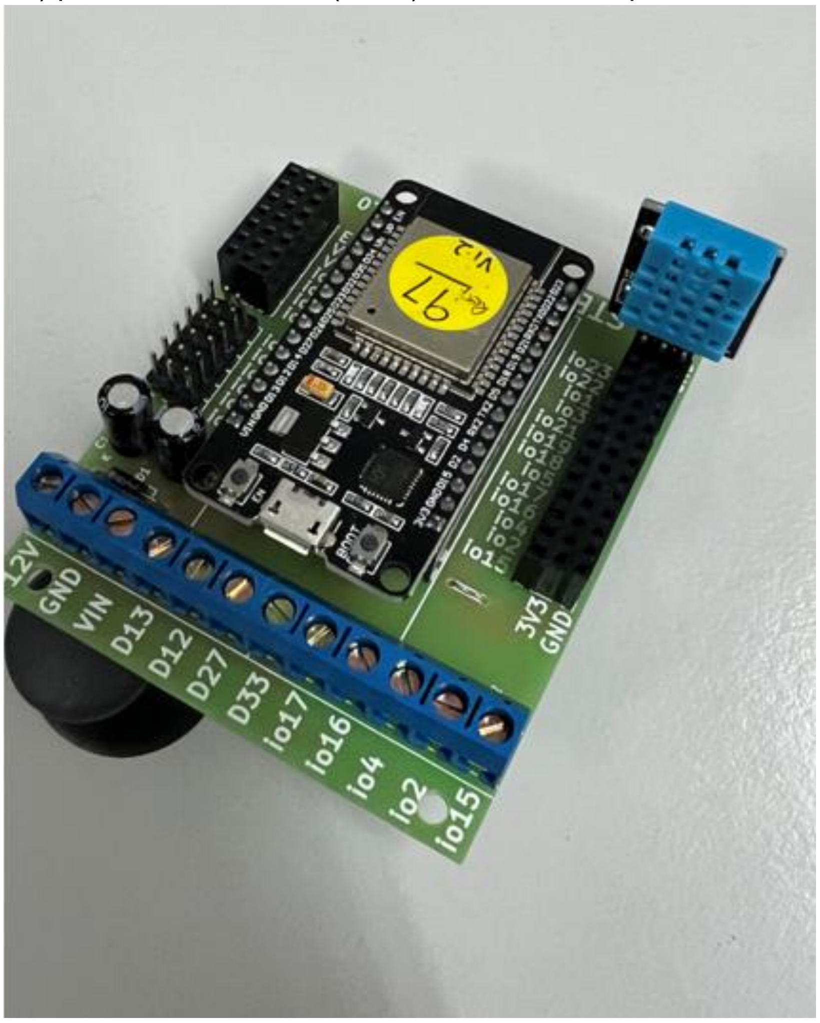


Figure 7: Circuit board connection with I/O part (side view)

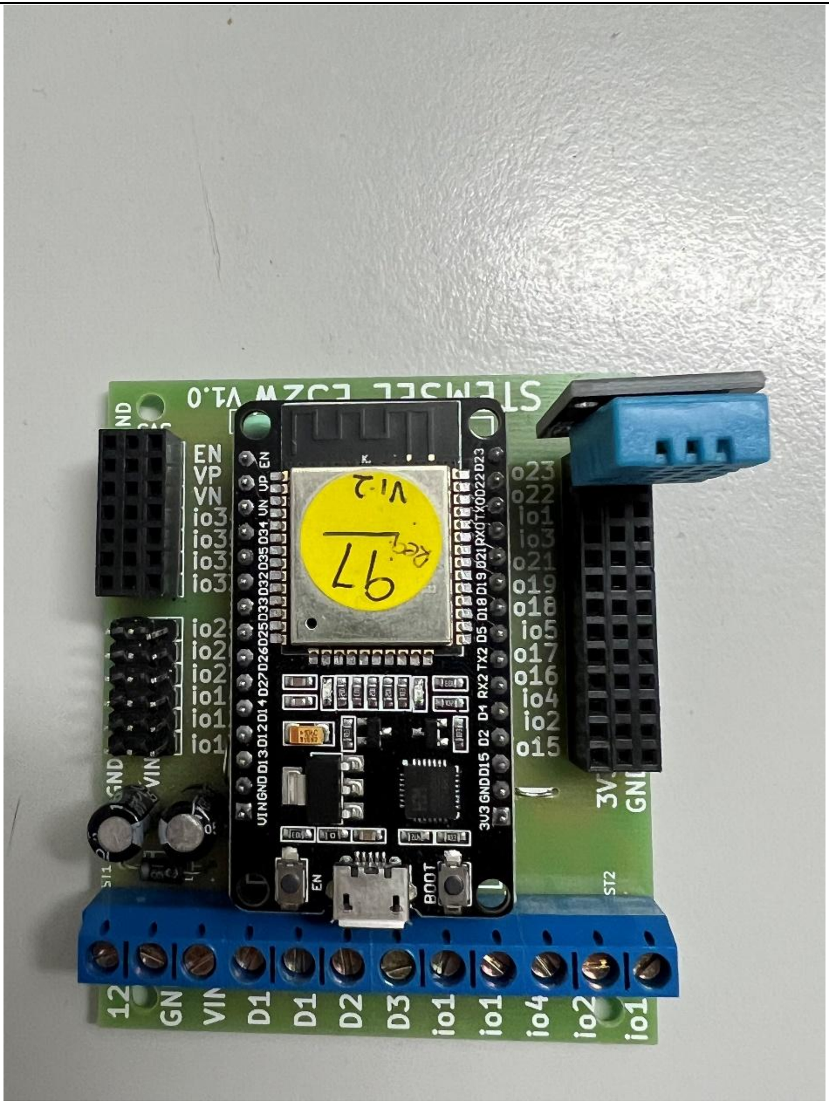


Figure 8: Circuit board connection with I/O part (top view)

Step 1: Graph temperature and humidity separately.

And then, we want to display the real-time temperature and humidity separately around the DHT11 sensor.

Program the Circuit

CSS:

Under CSS section, we will be creating a class that changes the font size to 25 pixels

```
div.a{
    font-size: 25px;
}
```

HTML:

Under HTML, we will be displaying the title of the webpage, Temperature and Humidity Graph. And then we will graph the real time measurement data of temperature and humidity respectively around the DHT11 sensor.

```
<div style = "text-align:center">
<h1> <font id = "Title" color = "darkblue"> Temperature + Humidity Graph </font> </h1>
<div class = "a">
<td> <font id = "Temperature"> </font> </td><br>
<td> <font id = "Humidity"> </font> </td>
</div>
<br>
<svg height = "350" width = "600">
    <polyline id = "tempGraph" style = "fill: none; stroke: darkred; stroke-width:3" />
    <polyline id = "humidityGraph" style = "fill: none; stroke: blue; stroke-width:3" />
    <text x = "95" y = "315" fill = "black">0</text>
    <text x = "65" y = "15" fill = "black">100</text>
    <text x = "70" y = "72" fill = "black">80</text>
    <text x = "70" y = "130" fill = "black">60</text>
    <text x = "70" y = "189" fill = "black">40</text>
    <text x = "70" y = "247" fill = "black">20</text>
    <text x = "320" y = "340" fill = "darkblue">Time</text>
    <text x = "0" y = "150" fill = "darkblue" id = "yTitleOne">Sensor</text>
    <text x = "0" y = "170" fill = "darkblue" id = "yTitleTwo">Levels</text>
    <line x1 = "100" y1 = "300" x2 = "600" y2 = "300" stroke = "black"/>
    <line x1 = "100" y1 = "0" x2 = "100" y2 = "300" stroke = "black"/>
</svg>
</div>
```

JavaScript:

```
// Graph display parameters (You'll still need to alter the position of the y axis numbers
manually)
var Height = 300; // Height of the graph
var Width = 500; // Width of the graph
var MaxValue = 103; // The maximum value that you want the graph to have within its
bounds
var StartingWidth = 100; // The starting width of the graph
var widthPerSample = 5; // The horizontal distance between 2 graph points
var maxWidthSample = Width / widthPerSample;
var i = 0;

// Sensor values + points
var currentTempHumidity;

var tempValues = [];
var humidityValues = [];

var tempPoints = [];
var humidityPoints = [];

// Function to update the graph
function updatedGraph(){
    // If the amount of points is greater than the width of the graph
    if (tempValues.length > maxWidthSample || humidityValues.length >
maxWidthSample) {
        // Clear the temperature and humidity points
        tempPoints = [];
        humidityPoints = [];
        tempValues.shift();
        humidityValues.shift();

        // Set the new points
        for (var j = 0; j < maxWidthSample; j++) {
           tempPoints[j] = ((StartingWidth - 5 * i) + widthPerSample * j) + "," + (Height - 
tempValues[j] * (Height / MaxValue));
           humidityPoints[j] = ((StartingWidth - 5 * i) + widthPerSample * j) + "," + (Height - 
humidityValues[j] * (Height / MaxValue));
        }
    }
    else {
        // Append a point to the array
        tempPoints[i] = StartingWidth + "," + (Height - tempValues[i] * (Height / MaxValue));
        humidityPoints[i] = StartingWidth + "," + (Height - humidityValues[i] * (Height / 
MaxValue));

        StartingWidth += 5;
        i++;
    }

    // Set the points on the polyline
    var tempGraph = document.getElementById('tempGraph');
    var updatedTempPoints = tempPoints.join(" ");
    tempGraph.setAttribute('points', updatedTempPoints);

    var humidityGraph = document.getElementById('humidityGraph');
    var updatedHumidityPoints = humidityPoints.join(" ");
    humidityGraph.setAttribute('points', updatedHumidityPoints);
}

// Simple function to display the values of the sensors
function displayValues(){
    document.getElementById('Temperature').innerHTML = "Current Temperature: " + 
currentTempHumidity[0] + " °C";
```

```
    document.getElementById('Humidity').innerHTML = "Current Humidity: " +
currentTempHumidity[1] + "%";
}
```

JavaScript Loop:

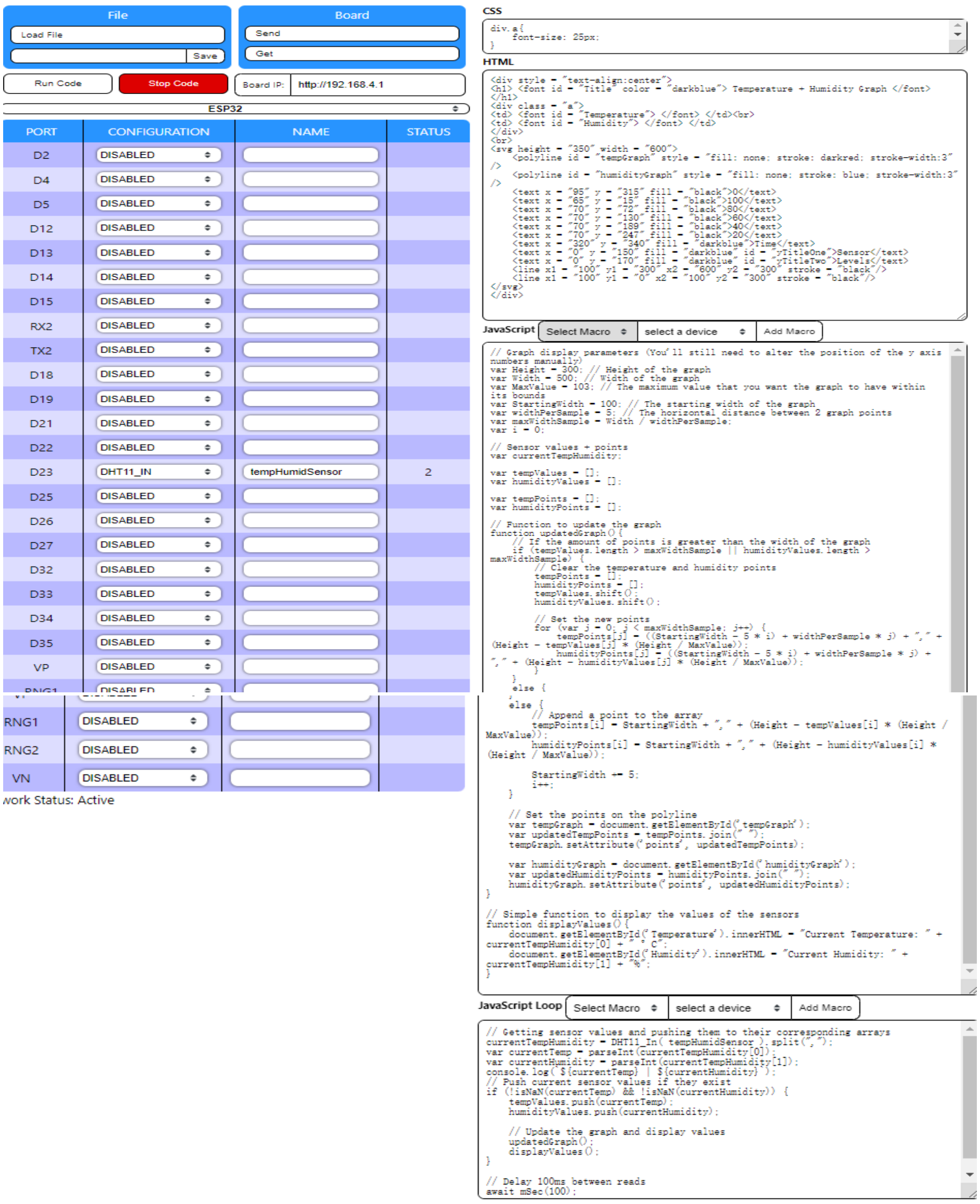
```
// Getting sensor values and pushing them to their corresponding arrays
currentTempHumidity = DHT11_In( tempHumidData ).split(",");
var currentTemp = parseInt(currentTempHumidity[0]);
var currentHumidity = parseInt(currentTempHumidity[1]);
console.log(`${currentTemp} | ${currentHumidity}`);
// Push current sensor values if they exist
if (!isNaN(currentTemp) && !isNaN(currentHumidity)) {
    tempValues.push(currentTemp);
    humidityValues.push(currentHumidity);

    // Update the graph and display values
    updatedGraph();
    displayValues();
}

// Delay 100ms between reads
await mSec(100);
```

Expected Result

Once you filled the codes in their respective section in the runlinc control page and once everything was ready. When we run the code, the webpage will display as shown in Figure 10. However, the program will run continuously.



PORT	CONFIGURATION	NAME	STATUS
D2	DISABLED		
D4	DISABLED		
D5	DISABLED		
D12	DISABLED		
D13	DISABLED		
D14	DISABLED		
D15	DISABLED		
RX2	DISABLED		
TX2	DISABLED		
D18	DISABLED		
D19	DISABLED		
D21	DISABLED		
D22	DISABLED		
D23	DHT11_IN	tempHumidSensor	2
D25	DISABLED		
D26	DISABLED		
D27	DISABLED		
D32	DISABLED		
D33	DISABLED		
D34	DISABLED		
D35	DISABLED		
VP	DISABLED		
RNG1	DISABLED		
RNG2	DISABLED		
VN	DISABLED		

work Status: Active

CSS

```
div.a{
    font-size: 25px;
}
```

HTML

```
<div style = "text-align:center">
<h1> <font id = "Title" color = "darkblue"> Temperature + Humidity Graph </font>
</h1>
<div class = "a">
<td> <font id = "Temperature"> </font> </td><br>
<td> <font id = "Humidity"> </font> </td>
</div>
<br>
<svg height = "350" width = "600">
    <polyline id = "tempGraph" style = "fill: none; stroke: darkred; stroke-width:3"
/>
    <polyline id = "humidityGraph" style = "fill: none; stroke: blue; stroke-width:3"
/>
    <text x = "95" y = "315" fill = "black">0</text>
    <text x = "65" y = "15" fill = "black">100</text>
    <text x = "70" y = "72" fill = "black">80</text>
    <text x = "70" y = "130" fill = "black">60</text>
    <text x = "70" y = "189" fill = "black">40</text>
    <text x = "70" y = "247" fill = "black">20</text>
    <text x = "320" y = "340" fill = "darkblue">Time</text>
    <text x = "0" y = "150" fill = "darkblue" id = "yTitleOne">Sensor</text>
    <text x = "0" y = "170" fill = "darkblue" id = "yTitleTwo">Levels</text>
    <line x1 = "100" y1 = "300" x2 = "600" y2 = "300" stroke = "black"/>
    <line x1 = "100" y1 = "0" x2 = "100" y2 = "300" stroke = "black"/>
</svg>
</div>
```

JavaScript

```
// Graph display parameters (You'll still need to alter the position of the y axis
numbers manually)
var Height = 300; // Height of the graph
var Width = 500; // Width of the graph
var MaxValue = 103; // The maximum value that you want the graph to have within
its bounds
var StartingWidth = 100; // The starting width of the graph
var widthPerSample = 5; // The horizontal distance between 2 graph points
var maxWidthSample = Width / widthPerSample;
var i = 0;

// Sensor values + points
var currentTempHumidity;

var tempValues = [];
var humidityValues = [];

var tempPoints = [];
var humidityPoints = [];

// Function to update the graph
function updatedGraph(){
    // If the amount of points is greater than the width of the graph
    if (tempValues.length > maxWidthSample || humidityValues.length >
maxWidthSample) {
        // Clear the temperature and humidity points
        tempPoints = [];
        humidityPoints = [];
        tempValues.shift();
        humidityValues.shift();

        // Set the new points
        for (var j = 0; j < maxWidthSample; j++) {
            tempPoints[j] = ((StartingWidth - 5 * i) + widthPerSample * j) + "," +
(Height - tempValues[j] * (Height / MaxValue));
            humidityPoints[j] = ((StartingWidth - 5 * i) + widthPerSample * j) +
"," + (Height - humidityValues[j] * (Height / MaxValue));
        }
    }
    else {
        // Append a point to the array
        tempPoints[i] = StartingWidth + "," + (Height - tempValues[i] * (Height /
MaxValue));
        humidityPoints[i] = StartingWidth + "," + (Height - humidityValues[i] *
(Height / MaxValue));

        StartingWidth += 5;
        i++;
    }

    // Set the points on the polyline
    var tempGraph = document.getElementById('tempGraph');
    var updatedTempPoints = tempPoints.join(" ");
    tempGraph.setAttribute('points', updatedTempPoints);

    var humidityGraph = document.getElementById('humidityGraph');
    var updatedHumidityPoints = humidityPoints.join(" ");
    humidityGraph.setAttribute('points', updatedHumidityPoints);
}

// Simple function to display the values of the sensors
function displayValues(){
    document.getElementById('Temperature').innerHTML = "Current Temperature: " +
currentTempHumidity[0] + " ° C";
    document.getElementById('Humidity').innerHTML = "Current Humidity: " +
currentTempHumidity[1] + "%";
}
```

JavaScript Loop

```
// Getting sensor values and pushing them to their corresponding arrays
currentTempHumidity = DHT11_In( tempHumidSensor ).split(",");
var currentTemp = parseInt(currentTempHumidity[0]);
var currentHumidity = parseInt(currentTempHumidity[1]);
console.log(`${currentTemp} | ${currentHumidity}`);
// Push current sensor values if they exist
if (!isNaN(currentTemp) && !isNaN(currentHumidity)) {
    tempValues.push(currentTemp);
    humidityValues.push(currentHumidity);

    // Update the graph and display values
    updatedGraph();
    displayValues();
}

// Delay 100ms between reads
await mSec(100);
```

Figure 10: Expect runlinc result screenshot

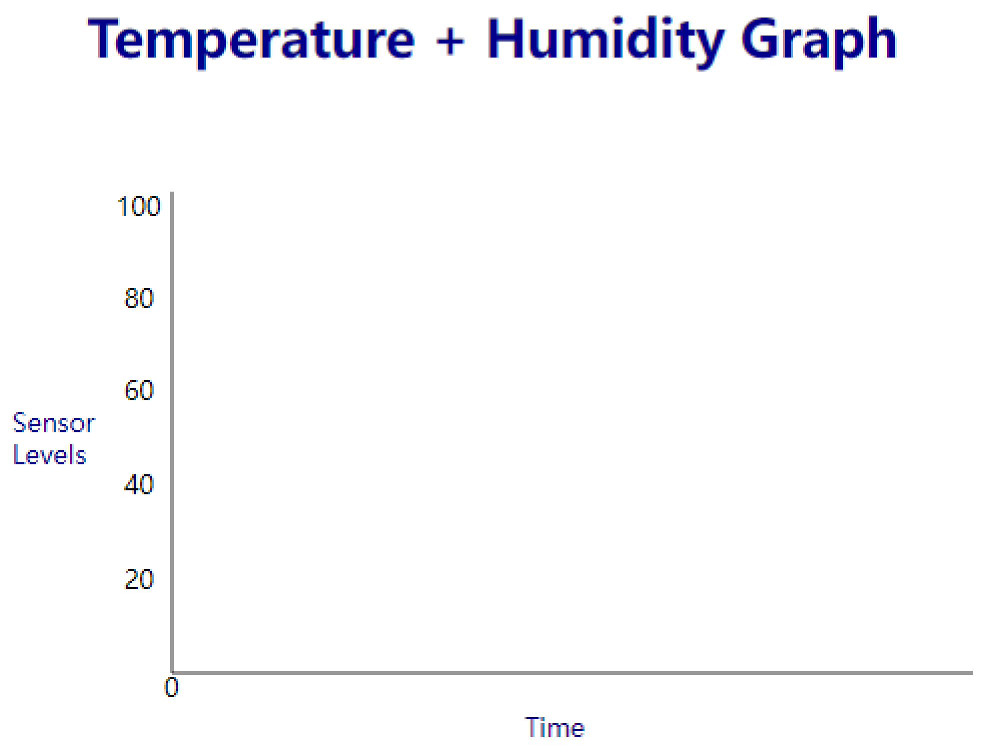


Figure 11 Expect webpage screenshot

Summary

Temperature and humidity are always playing crucial roles in the growth of green plants. Detecting the temperature and humidity of the crop growth environment in real time, farmers can better cultivate crops.