IOT Greenhouse Monitoring System

Raj Basnet
Western Michigan University, rajaditya2056@gmail.com

Follow this and additional works at: https://scholarworks.wmich.edu/honors_theses

Part of the Computer and Systems Architecture Commons

Recommended Citation
https://scholarworks.wmich.edu/honors_theses/3472

This Honors Thesis-Open Access is brought to you for free and open access by the Lee Honors College at ScholarWorks at WMU. It has been accepted for inclusion in Honors Theses by an authorized administrator of ScholarWorks at WMU. For more information, please contact wmu-scholarworks@wmich.edu.
IoT Greenhouse Monitoring System

Raj Basnet, Wesley Erby, & Bradley Heydel

Advisor: Dr. Dean Johnson

Sponsor: Erby Innovations

ECE 4820 ELECTRICAL & COMPUTER ENGINEERING DESIGN II

12/10/2021
2a. Abstract:

Our project is a greenhouse monitoring system. The customer states that they need a complete monitoring system for their greenhouse. There are a lot of items within the greenhouse that need to be watered at the right time and kept at a certain temperature. The customer is not always around to check the status of these items due to their busy lifestyle. They would like a system to monitor all these items so they can check it on their smartphone no matter how far away they are from the greenhouse. The customer wants this to be a low-cost and energy-efficient system. The system our group will develop for the customer will be an IoT greenhouse monitoring system. This will incorporate low-cost sensors to communicate to a phone application over Wi-Fi. Each sensor will be independently powered with its own battery. There will be various sensors included that will need to be used to monitor the greenhouse. A soil moisture sensor will be created to monitor the moisture of the soil and a temperature sensor will be created to monitor the temperature and humidity of the greenhouse. The phone app will include settings for “good” to “bad” temperature and soil moisture levels, as well as notifications that will be populated when the values are the “bad” range. The app will also include a time that the user sensors can be put into “deep sleep” mode to conserve battery power.
2b. Disclaimer:

DISCLAIMER
This report was generated by a group of engineering seniors at Western Michigan University. It is primarily a record of a project conducted by these students as part of curriculum requirements for being awarded an engineering degree. Western Michigan University makes no representation that the material contained in this report is error free or complete in all respects. Therefore, Western Michigan University, its faculty, its administration or the students make no recommendation for use of said material and take no responsibility for such usage. Thus persons or organizations who choose to use said material for such usage do so at their own risk.
2c. Senior Design Project Report Release Form:

WESTERN MICHIGAN UNIVERSITY  
COLLEGE OF ENGINEERING AND APPLIED SCIENCES  
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING  
KALAMAZOO, MICHIGAN 49008

SENIOR DESIGN PROJECT REPORT RELEASE FORM

In accordance with the "Policy on Patents and Release of Reports Resulting from Senior Design Projects" as adopted by the Executive Committee of the College of Engineering and Applied Sciences on Feb. 9, 1989, permission is hereby granted by the individuals listed below to release copies of the final report written for the Senior Design Project entitled:

PROJECT TITLE: **IOT Greenhouse Monitoring**

PROJECT SPONSOR* Did this project have a sponsor? YES _X_ (see footnote) NO ___

Contact person and email address &/or telephone Wesley Erby  erbyinnovations@gmail.com

Company Name  Erby Innovations

Design team has requested sponsor to verify in writing to course coordinator that all promised deliverables have been received. YES _X_ NO ___ (please check)

TEAM MEMBERS NAMES:

<table>
<thead>
<tr>
<th>NAME PRINTED</th>
<th>NAME SIGNED</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wesley Erby</td>
<td>Wesley Erby</td>
<td>12-10-2021</td>
</tr>
<tr>
<td>Bradley Heydel</td>
<td>Bradley Heydel</td>
<td>12-10-2021</td>
</tr>
<tr>
<td>Raj Basnet</td>
<td>Raj</td>
<td>12-10-2021</td>
</tr>
</tbody>
</table>

* Those teams with a sponsor must have sponsor provide the course coordinator with written evidence that they have provided the sponsor with a copy of the final project report as well as with other items that the team has promised to the sponsor. The evidence could be a short note via email, fax or US mail from the sponsor indicating receipt of a copy of the report and all promised deliverables.

Form Dr. John Senk  Version 4 September 2013  © 2013 Dennis A. Miller
2d. Acknowledgements / Permissions: Erby Innovations has given full permission to release this project and its contents to the public.

3. Table of Contents:

Contents
1. Front Page................................................................. 1
2a. Abstract.............................................................................. 2
2b. Disclaimer........................................................................... 3
2c. Senior Design Project Report Release Form ......................... 4
2d. Acknowledgements / Permissions........................................ 5
3. Table of Contents .................................................................. 5
4. Report Body........................................................................... 7
   4.1 Summary ........................................................................... 7
       4.1.2 Introduction ................................................................. 7
   4.2 Discussion .......................................................................... 7
       4.2.1 Background .................................................................. 7
       4.2.2 Need statement ............................................................ 7
       4.2.3 High Level System Diagram .......................................... 8
       4.2.4 Specifications ............................................................... 8
           4.2.4.1 Physical Characteristics ......................................... 8
           4.2.4.2 Electrical Design ................................................... 8
           4.2.4.3 Functionality ......................................................... 8
           4.2.4.4 Economic .............................................................. 9
           4.2.4.5 Health and Safety ................................................... 9
           4.2.4.6 Environmental Effects and Sustainability .................. 9
           4.2.4.7 Ethical ................................................................. 9
           4.2.4.8 Manufacturability .................................................. 9
       4.2.5 Deliverables ............................................................... 9
       4.2.6 Parts List / Bill of Materials ........................................ 10
       4.2.7 Comprehensive Description ........................................ 10
       4.2.8 Initial Approach ........................................................ 10
       4.2.9 Design Implementation/Changes .................................... 12
       4.2.10 Design Analysis ....................................................... 12
4. Report Body

4.1 Summary: For our project we wanted to create a relatively low-cost system for plant monitoring that includes remote functionality and an intuitive app-like user interface. This project was designed to aid greenhouse managers who cannot always be present to monitor their greenhouses. We were able to read data and send an object that can be read on two other software programs and parse accordingly for the completion of our IoT network. We were able to include a low power management system within the board to be able to support a longer battery life. In addition, we were able to test and calibrate soil moisture to have a moisture level reading of one sq. ft. and a pH reading from the same plot of soil. Our app allows users to have a user-friendly UI (user interface) through the use of charts and graphs.

4.1.2 Introduction: The following document is a report on the design, construction, and evaluation of an IoT greenhouse monitoring system (IoT GMS) which provides wireless monitoring and control of greenhouse environments. The IoT GMS incorporates low-cost sensors to communicate to a phone application over Wi-Fi. Each PCB is powered with its own battery. Various sensors have been included that are used to monitor the greenhouse. A PCB was created to monitor the moisture and acidity of the soil as well as temperature and humidity of the greenhouse. The phone app includes settings for ideal and out of range values of soil moisture and soil acidity. Notifications that are populated when the values are out of the ideal range. The sensor includes sleep functionality to preserve battery life.

4.2 Discussion:

4.2.1 Background: Erby Innovations is a company that wants to use IoT technology to develop a monitor and control system for greenhouse environments.

4.2.2 Need statement: There are a lot of items within the greenhouse that need to be watered at the right time and kept at a certain temperature. Greenhouse managers are not always around to check the status of these items due to their busy lifestyle. They would like a system to monitor all these items so they can check it on their smartphone no matter how far away they are from the greenhouse. This innovative monitoring system will include a database that will keep track of past data to show the history of the plant’s environment. These kinds of customers also want this to be a low-cost and energy-efficient system. Therefore, IoT GMS will provide wireless monitoring of greenhouse environments to allow users to remotely monitor their greenhouse.
4.2.3 High Level System Diagram:

![High Level System Diagram](image)

4.2.4 Specifications: The input is the moisture of the soil and the temperature of the surrounding air. The temperature and soil moisture sensor’s data will be the input. The output will be the data received by the Wi-Fi module and displayed on that app. The data will be displayed on a chart and will trigger notifications when the data drop into danger zone. The following are the specifications for the IoT GMS:

4.2.4.1 Physical Characteristics

4.2.4.1.a Size:

- Each PCB must be small enough with sensor circuit and Wi-Fi module to fit beside plants. Preferably smaller than 6cm x 15cm.

4.2.4.1.b Durability:

- Each PCB integrated with both the circuit and Wi-Fi module must be able to withstand being submerged in wet to damp soil without shorting.

4.2.4.2 Electrical Design

4.2.4.2.a Wi-Fi Module:

- The Wi-Fi module microcontroller (ESP32) must be able read the sensor data and publish this information to the cloud server using the IoT protocol MQTT.

4.2.4.2.b Circuit:

- The PCB must incorporate the capacitive sensor circuit and Wi-Fi module while running off a Lithium-ion battery.

4.2.4.3 Functionality

4.2.4.3.a Speed:
• Device must be able to read and publish sensor data in real time (About one measurement per one second).

4.2.4.3.b  User Interface:

• User interface must be easy to navigate and clear to read.
• Must provide graphs and charts to analyze data over periods of time.
• Must include option to query to show past data points.
• Must include notifications and clear indicators when data is performing in bad or good range.

4.2.4.4 Economic

• Greenhouse monitoring system must use low-cost components.

4.2.4.5 Health and Safety

• Greenhouse monitoring system must be safe to use, and it should offer no health or safety related issues to the user operating this system.
• Sensors will be sealed to prevent shock and saturation.

4.2.4.6 Environmental Effects and Sustainability

• System must contain a low power management setting to conserve battery life.
• System promotes using the minimum number of resources needed to grow and sustain the plants.

4.2.4.7 Ethical

• Must use original designs and concepts.
• The health and safety of the consumer must be the highest priority.

4.2.4.8 Manufacturability

• Parts used in design must be commercially available and be easily integrated into the design to construct the prototype system.

4.2.5 Deliverables:

• One PCB integrated Wi-Fi module that will be able to read data and publish it to a cloud server. The PCB can measure soil acidity, soil moisture, temperature of surrounding environment, and humidity of surrounding environment.
• App that includes the sensor data that has been recorded over various lengths of time and crop/soil performance data.
### 4.2.6 Parts List / Bill of Materials:

<table>
<thead>
<tr>
<th>Designator</th>
<th>Quantity Per Unit</th>
<th>Total Quantity Purchased</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2</td>
<td>1</td>
<td>1</td>
<td>1.15 µf 50V 50V Ceramic Capacitor X5R 0603 (1608 Metric)</td>
</tr>
<tr>
<td>C3, C7, C8, C11, C12</td>
<td>1</td>
<td>10</td>
<td>10 µf 0805 50V X7R 0603 (1608 Metric)</td>
</tr>
<tr>
<td>C4</td>
<td>1</td>
<td>1</td>
<td>1.000pF ±10% 50V Ceramic Capacitor X8R 0603 (1608 Metric)</td>
</tr>
<tr>
<td>C5</td>
<td>1</td>
<td>1</td>
<td>470pF ±2% 100V Ceramic Capacitor C0G, NP0 0603 (1608 Metric)</td>
</tr>
<tr>
<td>C6, C9</td>
<td>1</td>
<td>2</td>
<td>4.7uF ±10% 25V Ceramic Capacitor X5R 0603 (1608 Metric)</td>
</tr>
<tr>
<td>D1</td>
<td>1</td>
<td>1</td>
<td>2CO4 DC GEN FLR 100V 15MA 500123</td>
</tr>
<tr>
<td>J1</td>
<td>1</td>
<td>3</td>
<td>2 Position Receptacle Connector 0.100&quot; (2.54mm) Surface Mount, Right Angle Gold</td>
</tr>
<tr>
<td>J2</td>
<td>1</td>
<td>2</td>
<td>BNC Connector Jack, Female Socket 50Ohm Panel Mount, Through Hole, Right Angle Silver</td>
</tr>
<tr>
<td>J3</td>
<td>1</td>
<td>2</td>
<td>Battery Contact Clip AA 1 Cell SMD (SMT) Tab</td>
</tr>
<tr>
<td>J4</td>
<td>1</td>
<td>2</td>
<td>Conn Header 2 POS 2.54mm</td>
</tr>
<tr>
<td>R2</td>
<td>1</td>
<td>2</td>
<td>2 RES 1M OHM 1% 1/10W 0603</td>
</tr>
<tr>
<td>R3, R9</td>
<td>1</td>
<td>4</td>
<td>4 RES SMD 1K OHM 1% 1/10W 0603</td>
</tr>
<tr>
<td>R4</td>
<td>1</td>
<td>2</td>
<td>2 RES SMD 330 OHM 1% 1/10W 0603</td>
</tr>
<tr>
<td>R5</td>
<td>1</td>
<td>2</td>
<td>2 RES SMD 1KX OHM 1% 1/10W 0603</td>
</tr>
<tr>
<td>R6, R7, R9, R10, R11, R12, R13</td>
<td>1</td>
<td>14</td>
<td>14 RES SMD 51K OHM 1% 1/10W 0603</td>
</tr>
<tr>
<td>SW3</td>
<td>1</td>
<td>2</td>
<td>Switch Slide DIP 300mA 6V</td>
</tr>
<tr>
<td>U1</td>
<td>1</td>
<td>2</td>
<td>2C OSC SGL. Timer 2.1MHz B-55IC</td>
</tr>
<tr>
<td>U2</td>
<td>3</td>
<td>6</td>
<td>3C OPAMP GP Circuit, High Output Current, BOCX</td>
</tr>
<tr>
<td>U3</td>
<td>2</td>
<td>4</td>
<td>2C REG Linear Fixed 5.3V, 200mA Max Dropout Voltage, 50MAA 6T6DN</td>
</tr>
<tr>
<td>U4</td>
<td>2</td>
<td>4</td>
<td>2ESP32-S2-WROOM-1 Series Transceiver, 802.11 b/g/n (Wi-Fi, WIF, WIL, WIL) Evaluation Board</td>
</tr>
<tr>
<td>U4 SOCKET</td>
<td>2</td>
<td>1</td>
<td>2P1 Position Receptacle Connector 0.100&quot; (2.54mm) Through Hole Tin</td>
</tr>
<tr>
<td>CI</td>
<td>1</td>
<td>2</td>
<td>2 Battery 10200 (Rechargeable) Lithium-Ion 3.7V 2.6Ah</td>
</tr>
<tr>
<td>T&amp;H Sensors (not included on schematic)</td>
<td>1</td>
<td>2</td>
<td>2cs DHT11 Humidity Sensor Module Digital Temperature Humidity Sensor with Wires for Arduino</td>
</tr>
<tr>
<td>pH Sensor Probe (not included on schematic)</td>
<td>1</td>
<td>2</td>
<td>2GAOKSH PH0.14 Value Detect Sensor Module + PH Electrode Probe BNC For Arduino</td>
</tr>
<tr>
<td>Battery</td>
<td>1</td>
<td>1</td>
<td>SparkFun Electronics</td>
</tr>
<tr>
<td>Total Cost</td>
<td>$96.10</td>
<td>$99.81</td>
<td></td>
</tr>
</tbody>
</table>

### 4.2.7 Comprehensive Description: For the soil moisture sensor, the probe is placed in the soil and the reading is sent to the ESP32 Wi-Fi module. That value is then sent to the cloud server and processed on the UI (user interface) of the app, showing various values and information. A similar process is conducted for the soil acidity sensor. The probe on this sensor, however, requires the soil sample to be mixed with distilled water, which has a pH value of 7.0 in a 1:4 ratio respectively. Multiple readings are recommended due to the nature of the probe. These values are to be averaged out in order to receive a more accurate response. The soil acidity probe inputs a voltage reading into the ESP32 and then Node-Red takes that value and converts it to a pH value which is displayed on the UI.

### 4.2.8 Initial Approach: Our initial approach was to create two PCBs, one with a soil moisture sensor and one with a soil acidity sensor. Both PCBs included a temperature sensor and humidity sensor.
Soil Moisture Sensor:

Soil Acidity Sensor:

```
.dc VpH_Sensor list 120m 114.29m 108.57m 102.86m 97.14m 91.43m 85.71m 80m 74.29m 68.57m 62.86m 57.14m 51.43m 45.71m 40m 34.29m 28.57m 22.86m 17.14m 11.43m 5.71m
.include ADTL082.sub
```
4.2.9 **Design Implementation/Changes:** Our original design for the soil moisture circuit did not account for capacitor recharging. We also had to revise our Altium schematics to account for the ESP32 traces. Our final design combines both sensors into one schematic, eliminating the need for two separate PCBs. The soil acidity sensor was overhauled for better performance and to adjust to the new board design. The original design went from having a stand-alone chip to having a full ESP-32-S2-WROOM-1 development board. This updated ESP32 development board allowed us to implement a visual indicator to show processes such as sleep mode, wake-up, and data publishing sequences.

4.2.10 **Design Analysis:**

4.2.10.1 Hardware:
Soil Moisture Sensor: We used a TLC555 timer chip which runs at 1.5kHz. A low dropout 3.3V voltage regulator supplies a voltage to power the TLC555 timer whose output signal feeds a low pass filter (10kΩ resistor and the moisture sensing capacitor). This creates a peak-to-peak voltage of the waveform depending on the effective dielectric in the soil. A peak voltage detector provides the analog output signal that is read through the GPIO pin.
**Soil Acidity Sensor:** R10 and R12 form a voltage divider and U2A acts as a voltage buffer in order to not alter the division ratio of the voltage divider. This voltage divider takes the supply voltage of 3.3V and divides it in half resulting in 1.65V. This voltage then appears at R7. U2C is a gain amplifier in a non-inverting configuration with multiple inputs. By matching R6 and R7 and using super position, the output of U2C is equal to the sensor voltage + an offset voltage of 1.65V. The gain of this circuit is 2V/V.

![Soil Acidity Sensor Circuit Diagram](image)

**Temperature and Humidity Sensor:** For our temperature and humidity sensor we are using a DHT11. This sensor is connected to its own PCB which we connect the three pins into our female 3pin connector on our PCB. We read in the temperature and humidity in a separate function from the “main loop” and return a structure with both variables. These values are then included with the data object over MQTT with topic, “topic/data/publish”.

![Temperature and Humidity Sensor Circuit Diagram](image)

**4.2.10.2 Software:**

**Node Red Frontend Flow Description:** Node Red is an IoT application that includes nodes for input/output, code functions, and data display for user interface. The main
nodes used in our project is network, dashboard, function, and common for debugging and injecting. We are using the MQTT network nodes for publishing and subscribing data from the ESP32, cloud server, and on Node Red itself. The function node stores code for manipulation of data. The dashboard nodes are used to build our user interface. The complete flow below begins with the “dropdown” dashboard node. When a crop is selected, that string is stored in a global variable that we can use to dynamically set the data displays throughout the user interface. New data sent from our ESP32 is subscribed to and sent through a JSON and Function node to parse and populate the data on the user interface.

**Node Red Frontend UI:** The user can navigate through various panels. The "Welcome" panel allows the user to select the crop that will be growing in the soil they wish to monitor. The "Data Readings" panel shows soil pH level, soil moisture, temperature, and humidity of the surrounding environment. This panel also shows a
color indicator based on the measured value. The "Charts and Graphs" panel allows
the user to view Soil Moisture and Soil Acidity over a variety of time increments.

Node Red User Interface Gauge Design: On our soil moisture and soil acidity
panel in the app, we have two gauges that are dynamically loaded based on the
chosen crop. The soil moisture reading is sent from the sensor and then converted to
the output value on the ESP32. These values range from 4000 to 8300, where 4000
is Fully Dry (0%) shown below in red and 8300 is Fully Wet (100%) shown below
in green.

pH is a scale used to specify the acidity or basicity of an aqueous solution. This
scale ranges from 0 to 14 where 0 is highly acidic and 14 is highly basic. Distilled
water is known to be 7.0pH. When it comes to plants and the soil they are in, lower
pH values are more dangerous than high pH values. Because of this, our soil
acidity/pH gauge has a less than conventional color layout. Low pH readings
are shown in red, indicating that the soil requires immediate attention. Ideal pH readings
are shown in green, indicating that no additional attention is needed. Finally, high
pH readings are shown in yellow, indicating that the user should consider inspecting
the soil.
ESP32 Visual Indicators: We were able to make use of the RGB LED on the ESP32 to show the different modes of the ESP32 using colored light indicators. The first mode is “Wi-Fi and broker connection”. The flashing green LED indicates that we are trying to connect to Wi-Fi and the broker. After the connection is successfully completed, the green LED shows up and it goes to the second mode which is “Data reading and publishing”. The data is read and published every two seconds, while no data is being read or published, the LED glows a dim magenta. While data is being read and published, the LED glows bright magenta. When the ESP32 is in “Modem sleep”, our power saving mode, the LED glows a bright red, dimming continuously until it reaches low brightness. Our final mode is the “Wake up” mode which is when the ESP32 wakes up again, glowing bright blue from the LED and starts reading and publishing data again.

```cpp
if(count > 30) {
    setModemSleep();
    //Red
    pixels.setPixelColor(0, pixels.Color(255, 0, 0));
    Serial.println("Count: " + String(count));
    for(int i = 100; i > 0; i--){
        pixels.setBrightness(i);
        pixels.show();
        delay(10);
    }
    count++;
}

while(count > 30){
    delay(1000);
    Serial.println("Count: " + String(count));
    count++;
    if(count > 60){
        wakeModemSleep();
        pixels.setPixelColor(0, pixels.Color(0, 255, 0));
        for(int i = 1; i <=100; i++){
            pixels.setBrightness(i);
            pixels.show();
            delay(10);
        }
        count = 0;
    }
}
```
Cloud Server: Our cloud server is a virtual Ubuntu EC2 server instance that is hosted in Amazon Web Services. It runs the broker in the background of the server to support the publish and subscribe architecture for MQTT. We are running a python script in the server that reads the published messages from both the ESP32 and Node Red frontend. Our code checks two topics: "topic/data/publish" and "topic/data/query". The first topic, “topic/data/publish”, subscribes on incoming messages sent from the ESP32. The snippet below is the function that reads the payload from the topic which parses the data object into variables for an SQLite database entry that is also hosted on the cloud server. This function also checks for the second topic, “topic/data/query”, to perform a query on the database. The string payload value that is sent on “topic/data/query” goes to another function to perform the query that corresponds with the date request. The query data is published to our Node Red interface.

```
# include topic and payload message from broker
# parse data depending on topic value
def process_message(mess_topic, mess_payload):
    if mess_topic == "topic/data/publish":
        mqtt_val = json.loads(mess_payload)
        temp = float(mqtt_val["temperature"])  # temperature
        humid = float(mqtt_val["humidity"])    # humidity
        mois = float(mqtt_val["moisture"])    # moisture
        acid = float(mqtt_val["acidity"])     # acidity
        readData = convertTime()
        date_entry = readData[0]
        time_entry = readData[1]
        project = (temp, humid, date_entry, time_entry, mois, acid)
        id_num = create_project(project)
        print(temp, humid, mois, acid, date_entry, time_entry)
        print(id_num)
        # client.publish("topic/verification", "message recieved from test\r\rtopic/status")
```
4.2.11 **Description of Performance Testing:** We made sure that the dimensions of our PCB comply with the required specification. The sensors were built in such a way that they will not be shorted when submerged in damp soil and sealed to prevent electric shock to the user. The microcontroller was designed in such a way that it can receive and publish data in real time (once every two seconds) to the cloud server using IoT protocols. Our PCB is powered by a rechargeable Lithium-ion battery and implements modem sleep in order to conserve battery life. We simulated and tested all components with software before ordering and testing our PCBs. Sensor readings were sent to the cloud server and displayed on the Node Red UI using easy to read graphs.

4.2.12 **Testing Results:**

**Soil Acidity Sensor Testing on LTspice:**

![LTspice Diagram]

Results shown for an input reading of 8.0pH (-0.05714V or -57.14mV).
Table 1. Soil Acidity LTspice Simulation Results

<table>
<thead>
<tr>
<th>pH Value</th>
<th>pH Sensor input (mV)</th>
<th>pH Sensor input (V in)</th>
<th>Voltage after Voltage Divider (Vdiv)</th>
<th>Vout [V]</th>
<th>Expected Voltage Vexp (V in Vdiv)</th>
<th>Vexp-Vout</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5</td>
<td>142.86</td>
<td>0.14286</td>
<td>1.65123</td>
<td>1.79056</td>
<td>1.79409</td>
<td>0.00343</td>
</tr>
<tr>
<td>4.6</td>
<td>157.14</td>
<td>0.15714</td>
<td>1.65123</td>
<td>1.78494</td>
<td>1.78837</td>
<td>0.00343</td>
</tr>
<tr>
<td>4.7</td>
<td>181.43</td>
<td>0.18143</td>
<td>1.65123</td>
<td>1.77928</td>
<td>1.78266</td>
<td>0.00343</td>
</tr>
<tr>
<td>4.8</td>
<td>125.71</td>
<td>0.12571</td>
<td>1.65123</td>
<td>1.77351</td>
<td>1.77694</td>
<td>0.00343</td>
</tr>
<tr>
<td>4.9</td>
<td>120</td>
<td>0.12000</td>
<td>1.65123</td>
<td>1.76780</td>
<td>1.77123</td>
<td>0.00343</td>
</tr>
<tr>
<td>5</td>
<td>114.29</td>
<td>0.11429</td>
<td>1.65123</td>
<td>1.76209</td>
<td>1.76552</td>
<td>0.00343</td>
</tr>
<tr>
<td>5.1</td>
<td>108.57</td>
<td>0.10857</td>
<td>1.65123</td>
<td>1.75637</td>
<td>1.75980</td>
<td>0.00343</td>
</tr>
<tr>
<td>5.2</td>
<td>102.86</td>
<td>0.10286</td>
<td>1.65123</td>
<td>1.75086</td>
<td>1.75409</td>
<td>0.00343</td>
</tr>
<tr>
<td>5.3</td>
<td>97.14</td>
<td>0.09714</td>
<td>1.65123</td>
<td>1.74495</td>
<td>1.74837</td>
<td>0.00343</td>
</tr>
<tr>
<td>5.4</td>
<td>91.43</td>
<td>0.09143</td>
<td>1.65123</td>
<td>1.73932</td>
<td>1.74266</td>
<td>0.00343</td>
</tr>
<tr>
<td>5.5</td>
<td>85.71</td>
<td>0.08571</td>
<td>1.65123</td>
<td>1.73351</td>
<td>1.73694</td>
<td>0.00343</td>
</tr>
<tr>
<td>5.6</td>
<td>80</td>
<td>0.08000</td>
<td>1.65123</td>
<td>1.72780</td>
<td>1.73123</td>
<td>0.00343</td>
</tr>
<tr>
<td>5.7</td>
<td>74.29</td>
<td>0.07429</td>
<td>1.65123</td>
<td>1.72209</td>
<td>1.72552</td>
<td>0.00343</td>
</tr>
<tr>
<td>5.8</td>
<td>68.57</td>
<td>0.06857</td>
<td>1.65123</td>
<td>1.71638</td>
<td>1.71980</td>
<td>0.00342</td>
</tr>
<tr>
<td>5.9</td>
<td>62.86</td>
<td>0.06286</td>
<td>1.65123</td>
<td>1.71075</td>
<td>1.71409</td>
<td>0.00342</td>
</tr>
<tr>
<td>6</td>
<td>57.14</td>
<td>0.05714</td>
<td>1.65123</td>
<td>1.70495</td>
<td>1.70837</td>
<td>0.00342</td>
</tr>
<tr>
<td>6.1</td>
<td>51.43</td>
<td>0.05143</td>
<td>1.65123</td>
<td>1.69954</td>
<td>1.70266</td>
<td>0.00312</td>
</tr>
<tr>
<td>6.2</td>
<td>45.71</td>
<td>0.04571</td>
<td>1.65123</td>
<td>1.69392</td>
<td>1.69694</td>
<td>0.00292</td>
</tr>
<tr>
<td>6.3</td>
<td>40</td>
<td>0.04000</td>
<td>1.65123</td>
<td>1.68831</td>
<td>1.69123</td>
<td>0.00342</td>
</tr>
<tr>
<td>6.4</td>
<td>34.19</td>
<td>0.03419</td>
<td>1.65123</td>
<td>1.68210</td>
<td>1.68552</td>
<td>0.00342</td>
</tr>
<tr>
<td>6.5</td>
<td>28.57</td>
<td>0.02857</td>
<td>1.65123</td>
<td>1.67638</td>
<td>1.67980</td>
<td>0.00342</td>
</tr>
<tr>
<td>6.6</td>
<td>22.86</td>
<td>0.02286</td>
<td>1.65123</td>
<td>1.67077</td>
<td>1.67409</td>
<td>0.00342</td>
</tr>
<tr>
<td>6.7</td>
<td>17.14</td>
<td>0.01714</td>
<td>1.65123</td>
<td>1.66495</td>
<td>1.66837</td>
<td>0.00342</td>
</tr>
<tr>
<td>6.8</td>
<td>11.43</td>
<td>0.01143</td>
<td>1.65123</td>
<td>1.65924</td>
<td>1.66266</td>
<td>0.00342</td>
</tr>
<tr>
<td>6.9</td>
<td>5.71</td>
<td>0.00571</td>
<td>1.65123</td>
<td>1.65352</td>
<td>1.65694</td>
<td>0.00342</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0.00000</td>
<td>1.65123</td>
<td>1.64782</td>
<td>1.65123</td>
<td>0.00341</td>
</tr>
<tr>
<td>7.1</td>
<td>-5.71</td>
<td>-0.00571</td>
<td>1.65123</td>
<td>1.64211</td>
<td>1.64552</td>
<td>0.00341</td>
</tr>
<tr>
<td>7.2</td>
<td>-11.43</td>
<td>-0.01143</td>
<td>1.65123</td>
<td>1.63639</td>
<td>1.63980</td>
<td>0.00341</td>
</tr>
<tr>
<td>7.3</td>
<td>-17.14</td>
<td>-0.01714</td>
<td>1.65123</td>
<td>1.63068</td>
<td>1.63409</td>
<td>0.00341</td>
</tr>
<tr>
<td>7.4</td>
<td>-22.86</td>
<td>-0.02286</td>
<td>1.65123</td>
<td>1.62496</td>
<td>1.62837</td>
<td>0.00341</td>
</tr>
<tr>
<td>7.5</td>
<td>-28.57</td>
<td>-0.02857</td>
<td>1.65123</td>
<td>1.61925</td>
<td>1.62266</td>
<td>0.00341</td>
</tr>
<tr>
<td>7.6</td>
<td>-34.29</td>
<td>-0.03429</td>
<td>1.65123</td>
<td>1.61353</td>
<td>1.61684</td>
<td>0.00341</td>
</tr>
<tr>
<td>7.7</td>
<td>-40</td>
<td>-0.04000</td>
<td>1.65123</td>
<td>1.60782</td>
<td>1.61123</td>
<td>0.00341</td>
</tr>
<tr>
<td>7.8</td>
<td>-45.71</td>
<td>-0.04571</td>
<td>1.65123</td>
<td>1.60211</td>
<td>1.60552</td>
<td>0.00341</td>
</tr>
<tr>
<td>7.9</td>
<td>-51.43</td>
<td>-0.05143</td>
<td>1.65123</td>
<td>1.59656</td>
<td>1.59980</td>
<td>0.00341</td>
</tr>
<tr>
<td>8</td>
<td>-57.14</td>
<td>-0.05714</td>
<td>1.65123</td>
<td>1.59098</td>
<td>1.59409</td>
<td>0.00341</td>
</tr>
</tbody>
</table>

Table 1. Soil Acidity LTspice Simulation Results

Our circuit was tested in LTspice using voltage readings corresponding to pH values ranging from 4.5pH to 8.0pH. These readings were then compared to our expected values. In the table above, the column on the right shows the difference between our simulated values and our expected values. There is a consistent 3.4mV difference between these values allowing us to incorporate this difference and adjust our final design accordingly.
4.2.13 Quantified Summary of Specifications Met:

<table>
<thead>
<tr>
<th>Specifications</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCB must be no larger than 6cm x 15cm in size (excluding enclosure).</td>
<td>✓</td>
</tr>
<tr>
<td>Sensors must be sealed to prevent electric shock.</td>
<td>✓</td>
</tr>
<tr>
<td>Device must be able to read and publish sensor data in real time (one measurement per second).</td>
<td>✗ We achieved one measurement per two seconds</td>
</tr>
<tr>
<td>Application must provide graphs and charts to analyze data over period.</td>
<td>✓</td>
</tr>
<tr>
<td>ESP32 module reads the sensor data and publishes this information to the cloud server using the IoT protocol.</td>
<td>✓</td>
</tr>
<tr>
<td>Application must include notifications and clear indicators when data is performing in bad or good range.</td>
<td>✓</td>
</tr>
<tr>
<td>The PCB must incorporate the capacitive sensor circuit and Wi-Fi module while running off a Lithium-ion battery.</td>
<td>✓</td>
</tr>
<tr>
<td>System must contain a low power management setting to conserve battery life such as deep-sleep.</td>
<td>✗ We were only able to incorporate modem-sleep to conserve battery life.</td>
</tr>
<tr>
<td>Parts are commercially available.</td>
<td>✓</td>
</tr>
</tbody>
</table>

Our estimated specification compliance/total project completion at around 85%.

4.3 Conclusion: Our team created a greenhouse monitoring system that uses IoT. We created one PCB that includes sensors that monitor soil moisture, soil acidity, temperature, and humidity. An application using Node Red to display our sensor data allowing ease of access to crucial data the user might require was also created. Our team learned how to design and implement an IoT network using an AWS cloud server. During this process, we learned how to utilize MQTT publish/subscribe architecture over a cloud broker. We also learned about database management and design utilizing SQL for data logging. Additionally, our team gained experience constructing a user interface and how to display data with it.

4.4 Recommendations: Some recommendations we would make to others looking to improve or expand upon our project include further research into the pH sensor. A ground probe would improve the accuracy of the sensor immensely. We recommend making a testing a probe that has proper calibration. The IoT network can be configured to run on Linux based systems such as a raspberry pi 4 for local network usage.
4.5 Appendix:

https://www.gardenersnet.com/atoz/phlevel1.htm This is where we got the ranges for our soil acidity ranges.

https://oshpark.com/ This is the company who manufactured our PCBs.

https://www.espressif.com/sites/default/files/documentation/esp32_hardware_design_guidelines_en.pdf - Page 20: PCB capacitor probe trace were designed to these standards