

HYDROFLORA (A CONTEXT-AWARE WATERING CAN)

By

Delilah Dzulkaflī

Idris Ispandi

Charis Wang

Final Report for ECE 445, Senior Design, **Spring 2026**

TA: **Mingrui Liu**

Professor: **Viktor Gruev**

May 06 2026

Team 55

Abstract

This report details the motivation, design, and implementation of Hydroflora, a wireless plant water monitoring and control system. Hydroflora uses multiple sensor nodes to collect soil moisture data through capacitive moisture sensors. These readings are transmitted wirelessly to a main control node, which processes the data and uses a feedback loop to control a motor-driven watering mechanism. Based on the measured soil moisture level, the system dispenses an appropriate amount of water to maintain optimal plant moisture conditions. Throughout the project, four iterations of PCB design were developed and tested, allowing improvements in circuit layout, power distribution, component selection, and system reliability. Key engineering concepts explored include communication between multiple integrated circuits, power regulation, sensor interfacing, motor control, and wireless data transfer. The project demonstrates how embedded systems, circuit design, and feedback control can be integrated into a practical automated watering solution. The challenges and failures are also explored and discussed extensively in the following report.

Contents

1. Introduction.....	4
1.1 Problem.....	4
1.2 Solution.....	4
1.3 Visual Aid.....	5
1.4 Block Diagram.....	6
1.5 Design Changes.....	6
1.6 High Level of Requirements.....	7
2. Design.....	8
2.1 Design Procedure.....	8
2.2 Design Details.....	8
2.2.1 Watering Can.....	8
2.2.2 Sensor Node.....	12
3. Verification.....	15
3.1 Sensing Subsystem.....	15
3.2 User Interface Subsystem.....	16
3.3 Control Subsystem.....	16
3.4 Water Dispensing Subsystem.....	17
3.5 Power Subsystem.....	17
4. Cost and Schedule.....	19
4.1 Cost.....	19
4.1.1 Parts Cost.....	19
4.1.2 Labor Cost.....	19
4.1.3 Total Cost.....	19
4.2 Schedule.....	19
5. Conclusion.....	20
5.1 Accomplishments.....	20
5.2 Uncertainties.....	20
5.3 Ethical considerations.....	20
5.4 Future work.....	21
References.....	22
Appendix A Requirement and Verification Table.....	24
Appendix B Parts Cost Table.....	31
Appendix C Schedule.....	34

1. Introduction

1.1 Problem

Many people care for multiple houseplants with different watering needs, but watering is typically done by intuition and inconsistent habits. Because plant type, pot size, soil type, and moisture all affect how much water a plant actually needs, manual watering often results in overwatering or underwatering. Overwatering can lead to root rot, fungus gnats, and wasted water, while underwatering causes plant stress, slowed growth, and wilting. However, existing reminders or generic schedules don't adapt to real-time soil conditions, and fully automated irrigation systems can be too expensive, complex, or impractical for small indoor plant collections. Thus, there is a need for a simple, low-effort tool that helps users deliver the correct amount of water per plant based on measured soil dryness and plant/ot-specific requirements, without requiring a permanent installed system.

From a broader perspective, our project addresses environmental sustainability by reducing water waste because only the exact optimal amount of water needed for the plants will be dispensed to avoid overwatering. This project also relates to economic efficiency by preventing the loss of expensive indoor greenery as problems relating to watering that might cause the death of plants can be avoided. Furthermore, studies [10] have shown that indoor gardening has become a vital component of psychological well-being. By providing a tool that ensures plant health, our project directly supports the mental wellness of the society.

1.2 Solution

Our solution is a Smart Watering Can System. We say that the watering can is context-aware because it does the watering job based on the feedback received from the components surrounding it. The project consists of modular, low-power sensor nodes and a centralized intelligent dispensing unit. The sensor nodes are placed permanently in plant pots to monitor soil moisture levels using capacitive sensing, which avoids the corrosion issues typical of resistive probes. These nodes broadcast real-time data to the watering can via a wireless protocol (Bluetooth).

The user interacts with a microcontroller-based interface on the watering can. When the user selects a plant through the buttons/rotary knob, the system compares the current moisture reading against a pre-set threshold for that specific plant species. Upon activation, a peristaltic pump dispenses a precise, calculated volume of water. This implementation moves plant care from being "scheduled-based" to

"data-driven", ensuring that the exact lack of moisture is replenished with accurate calculations instead of human estimation.

1.3 Visual Aid

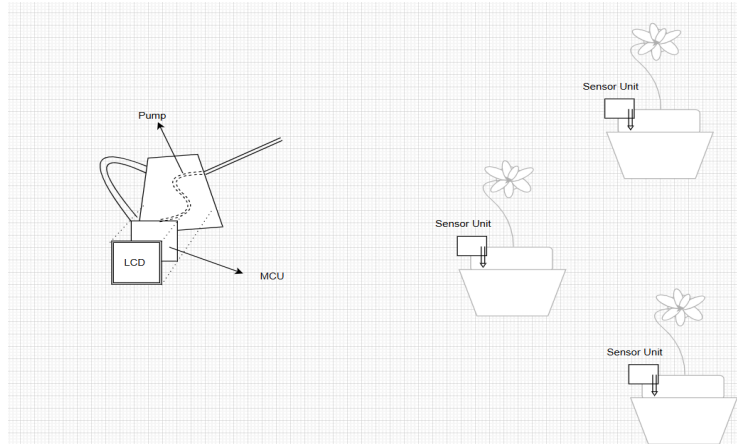


Figure 1: Project Visual Aid

1.4 Block Diagram

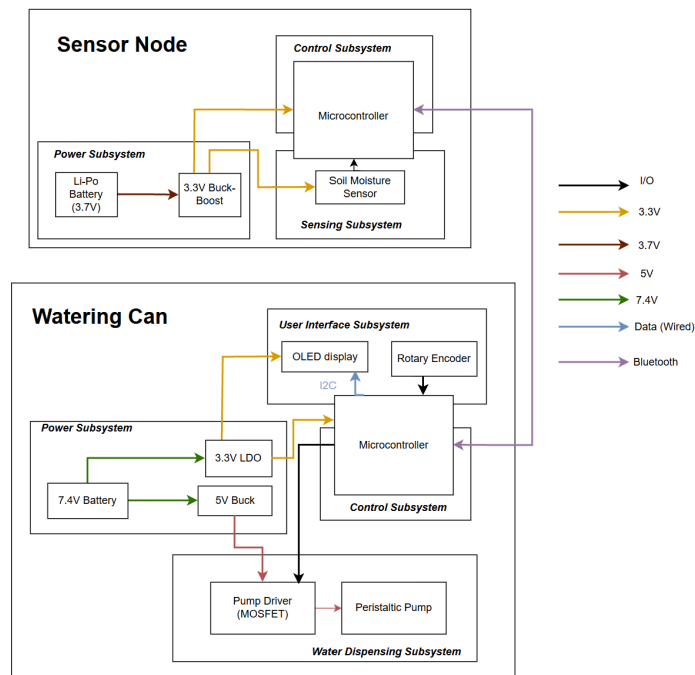


Figure 2: System Block Diagram

1.5 Design Changes

Looking at Figure 2, we have the block diagram which is broken down into two main blocks serving different subsystems, namely the sensor node block and the main control unit at the watering can. The sensor system is mainly for recording how much moisture is within the plant. Using Bluetooth, we are able to transmit the data to the watering can subsystem that is able to calculate the proper amount of water and output it to the plant to reach its target moisture. We had mainly changed the power subsystem updating along with changing some minor changes to the watering can subsystem. Within the sensor subsystem module, the power system within it changed from having a 3.3V LDO to a 3.3V buck-boost converter to allow maximum efficiency. Furthermore, we had also changed the power system within the watering can to power the motor with a 7.4V battery that has less voltage sag than the 3.7V battery that we were initially using. Additionally, instead of using selection buttons for the watering can module, we will be using an encoder.

1.6 High Level of Requirements

- **Accuracy:** The system must dispense the calculated volume of water with a precision of $\pm 20\%$ of the target volume.
- **Connectivity:** The sensor node must maintain a reliable wireless connection and update soil moisture data to the Main Control Unit at a range of at least 3 meters in an indoor environment.
- **Longevity:** The sensor node must operate continuously for a minimum of 24 hours on a single battery charge
- **Variation:** The system must be able to recognize different types of plants according to user input and dispense the correct amount of water based on the current plant's data.

2. Design

2.1 Design Procedure

The watering can module has 4 main subsystems: user interface subsystem, control subsystem, power subsystem, and water dispensing subsystem. The user interface subsystem allows users to see the percentage of moisture on the selected plant and the target percentage that the plant needs for optimal hydration. Additionally, because there are multiple sensor modules and plants, we have a rotary encoder to select the desired plant. The control subsystem is to ensure that we are sending the right signals to the OLED display and the pump driver. This control subsystem from the watering can will also communicate with the control subsystem in the sensor node in order to get the proper data. The power subsystem mainly supplies the microcontroller and the pump. The water dispensing subsystem will have a pump that will be controlled with a pump driver that will allow it to appropriately dispense water.

The sensor node module has 3 main subsystems: sensing subsystem, control subsystem, and power subsystem. The sensing subsystem contains a soil moisture sensor that gathers data upon how much moisture is in the soil. This data gets sent to the control subsystem which will then communicate with the other control subsystem on the watering can module. The power subsystem supplies power to the control and sensing subsystem.

2.2 Design Details

2.2.1 Watering Can

2.2.1.1 Water Dispensing Subsystem

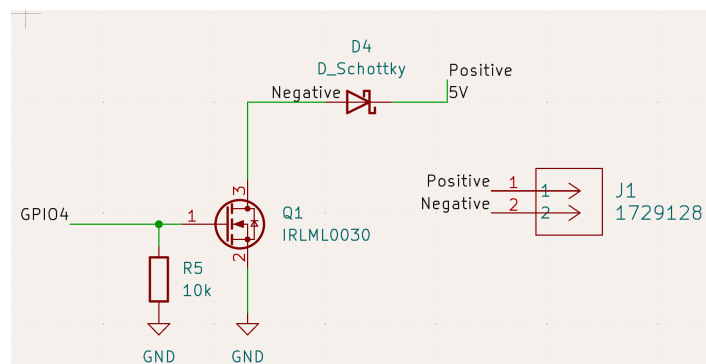


Figure 3: Water Dispensing Subsystem Schematic

Figure 3 is a circuit diagram of Water Dispensing Subsystem where on the left hand side we drive the pump using the microcontroller GPIO pin and have the positive and negative label where the pump is connected to. This physical actuator block takes the low-power logic signal from the Control Subsystem and uses a MOSFET to switch a 5V supply to the pump, moving water from the reservoir to the spout of the watering can. There is also a Schottky diode incorporated into the system as a flyback diode placed across the pump's terminals. This diode allows a safe path for the high-voltage spike to dissipate whenever the MOSFET turns off to prevent it from frying the ESP32.

The pump that we used is a 5V-6V DC Power Peristaltic Liquid Pump with Silicone Tubing (Adafruit 3910) [4]. This specific pump was chosen for a better precision level because a peristaltic pump works by using rollers to “squeeze” or compress a flexible tube, and the rollers will move a fixed volume of water per rotation. Besides, this will eliminate the risk of the pump itself touching the water since only the pump tubing will be in contact with the water.

2.2.1.2 Power Subsystem

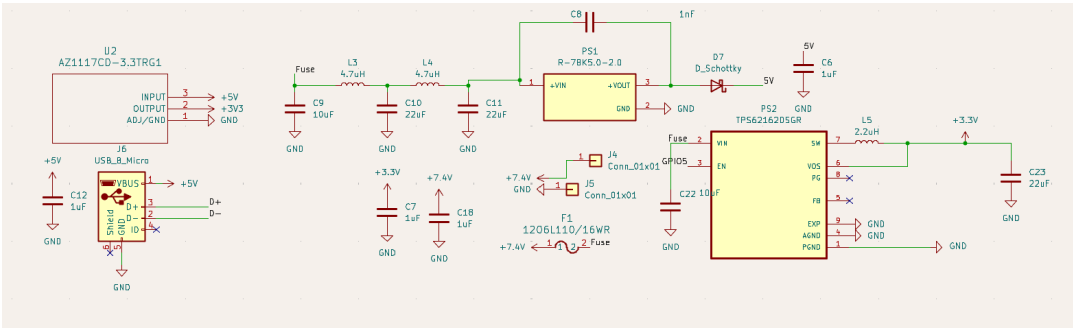


Figure 4: Power Subsystem Schematic - Watering Can Module

As shown in figure 4, the power subsystem that supplies the watering can module contains a 7.4V battery that will have a fuse attached to it just in case. Additionally, it will be connected to a 3.3V buck converter as well as a 5V buck converter to supply the controlling subsystem as well as the water dispensing subsystem. In order to program the microcontroller we simply had a micro usb that goes into a 3.3V LDO that supplies the microcontroller without the battery supply.

2.2.1.3 User Interface Subsystem

The user interface system handles all the interactions between users and the watering can. It first consists of an OLED module, figure 5, to display the plant types and their corresponding real-time soil moisture levels, driven by the control subsystem.

An OLED is preferred over LCD in order to achieve a better readability under any lighting conditions, because it should work well in both bright and dark environments. Since the OLED's individual pixel is its own light source, when a pixel is black, it will be completely off. This results in a high-contrast text that is easier to read, as compared to a LCD that has a backlight that constantly stays on.

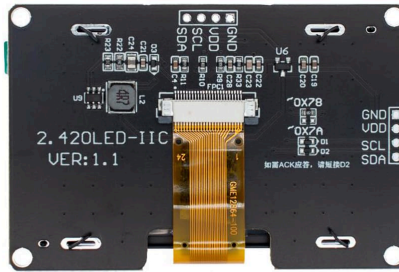


Figure 5: OLED Display

We are using the HiLetgo 2.42" SSD1309 OLED Display Module as shown in Figure 5 that uses the I2C protocol. The four pins needed are SDA (for sending data), SCL (for clock), VDD, GND, all of which will be driven by the control subsystem.

Furthermore, the user interface subsystem consists of a rotary encoder knob for the user's selection of plants. Since we implemented a sliding window method for the display, each plant will get a page on the display. Users can turn the encoder knob in order to view or select another plant profile. The encoder button can then be pressed to start the watering process. All five pins on this rotary encoder are also driven by the control subsystem through the GPIO pins.

2.2.1.4 Main Control Subsystem

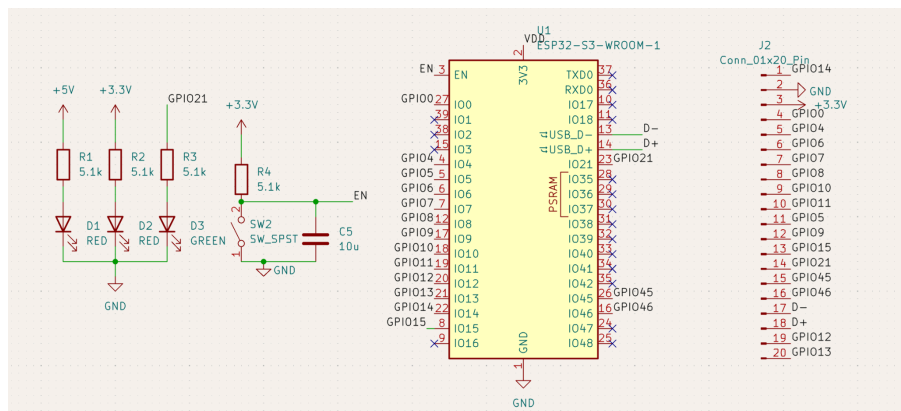


Figure 6: Control Subsystem Schematic

The control subsystem shown in figure 6 for the watering can is responsible for communicating with the sensor node controller, storing moisture level data for different plants, and calculating the timing for pump. The microcontroller that we used is the ESP32-S3-WROOM-1. We are using a dynamic watering logic because the amount of water to be dispensed is different according to different types of plants. For each plant, we have a pre-defined target moisture level percentage that determines how much volume of water is needed to meet that threshold using the formula:

$$Pump\ time\ (s) = \frac{Volume\ needed\ to\ meet\ target\ moisture\ (mL)}{Pump\ flow\ rate\ (mL/s)} \quad (1)$$

This subsystem is also a receiver node of the data packets containing the soil moisture readings sent by the transmitter sensor node via Bluetooth Low Energy (BLE). It then updates the moisture value on the user interface and does the pump timing calculation logic using (1). If the controller could not establish a connection with the sensor node and does not receive a moisture value, this subsystem will send a control signal that disables the pump for safety reasons.

As the central subsystem in charge of the overall functionality of the project, the following flowchart is implemented to ensure smooth integration between all subsystems.

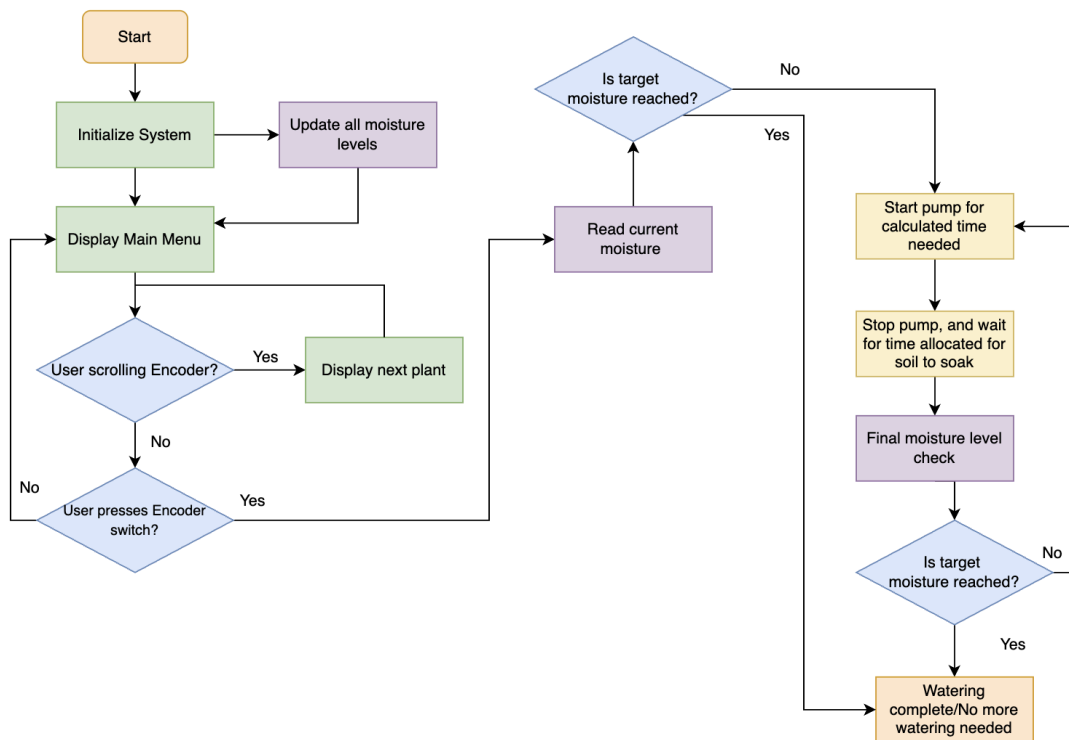


Figure 7: Control system flowchart

2.2.1.5 PCB Design

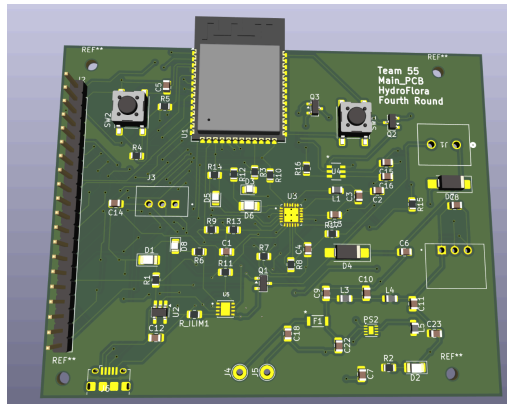


Figure 8: Main PCB

In Figure 8, we have the main PCB of the watering can module that will then contain all the above mentioned subsystems. The encoder and the OLED display are integrated via the connector pins on the left hand side.

2.2.2 Sensor Node

2.2.2.1 Sensing Subsystem

The sensing subsystem is the source of data for the whole system. It measures the soil moisture using a capacitive probe and converts it into a digital value using the ESP32's GPIO pins with internal ADC. The sensor that we use is a capacitive soil moisture sensor (DFRobot SEN0193) [6]. The advantage of this sensor unit compared to a resistive sensor is that the probes are made of corrosion resistant material which allows it to have a longer service life.

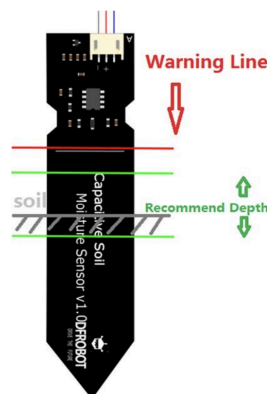


Figure 9: Sensor diagram

It is important to ensure that the probe is not submerged in soil or water beyond the warning line in Figure 9 to avoid damaging the electronics components above it. Once the sensor values are calibrated, we can use the readings and convert them to a soil moisture percentage with the following formula:

$$\text{Moisture \%} = \frac{\text{Value}_{\text{Raw}} - \text{Value}_{\text{Dry}}}{\text{Value}_{\text{Wet}} - \text{Value}_{\text{Dry}}} \times 100 \quad (2)$$

The dry value is the sensor reading when in open air while the wet value is when the probe is fully submerged in water, in order to compare the actual moisture reading relative to both ends of the spectrum.

2.2.2.2 Sensor Control Subsystem

Figure 10 shows the control subsystem which is responsible for transmitting the soil moisture percentage level obtained from the sensor to the main controller via Bluetooth Low Energy (BLE).

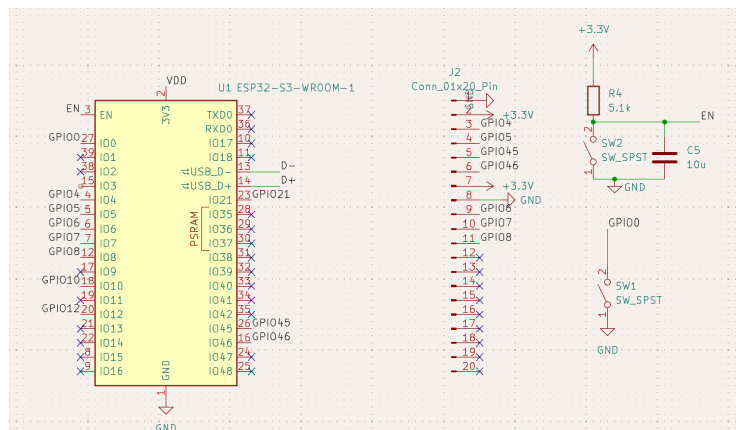


Figure 10: Control Subsystem schematic

The control subsystem shown in figure 10 has the ESP32-S3-WROOM-1 microcontroller which advertises the BLE data packets to the main watering can receiver. The advertising and sampling interval is every 10 seconds to allow some time for water to spread across the soil if necessary. The moisture percentages calculated by (2) will then be used by the main control subsystem to determine whether a plant needs watering and the volume of water needed. As shown in the schematic above, the soil moisture sensor is connected with the microcontroller via the GPIO pin. Each plant corresponds to a different sensor controller unit, and for this project we are incorporating three sensor nodes.

2.2.2.3 Power Subsystem

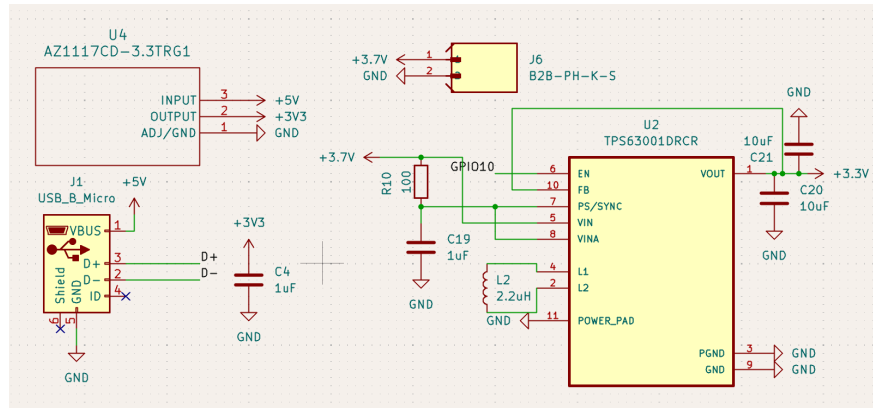


Figure 11: Power Subsystem Schematic

Figure 11 shows the power subsystem supplies a 3.7V battery in which we use a 3.3V buck-boost converter in order to supply power to the control and sensing subsystem. Additionally, in order to program/power the battery we had used a 3.3V LDO that is connected to a Micro-USB.

2.2.2.4 PCB Design

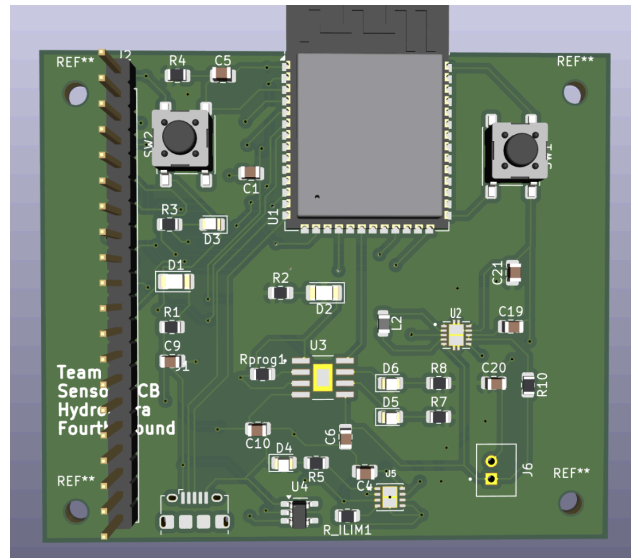


Figure 12: Sensor PCB

Figure 12 is the Sensor Node Module within the Sensor PCB which also contains the subsystems within the module mentioned above.

3. Verification

3.1 Sensing Subsystem

We have verified that the sensor node could remain operational for at least 24 hours without recharge.

Reading #	Measured DC Current (mA)
1	18.94
2	21.37
3	19.62
4	20.81
5	17.88
6	22.46
7	19.15
8	20.28
9	21.04
10	18.73

Table 1: Current measure across battery

From the results above, we get that the average current used is 20.03 mAh which is way less than the battery capacity of 2000 mAh capacity.

Next, to test if the capacitive soil moisture sensor can resist corrosion, we recorded its condition from Day 1 and after Day 30 of submerging the probe in moist soil with periodic watering. The following pictures indeed verified that the sensor did not face any water-induced degradation.



Figure 13.1: Day 1 Sensor condition



Figure 13.2: Day 30 Sensor condition

To test if the sensor node is able to transmit a valid moisture reading within an indoor range of at least 3 meters, we measured the distance between the sensor node and the watering can's controller to be 3

meters and checked whether the Bluetooth connection is established and that the sensor readings are successfully updated. This is successfully achieved as shown in the demo.

3.2 User Interface Subsystem

To test if the OLED display is functional under any lighting conditions, we placed the display in two different room settings. The following results verify that the display clearly shows the current moisture percentage of all plants in bright or dark conditions.

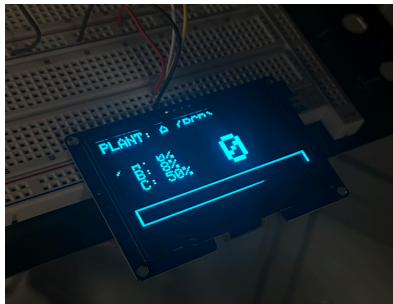


Figure 14.1: Dark condition

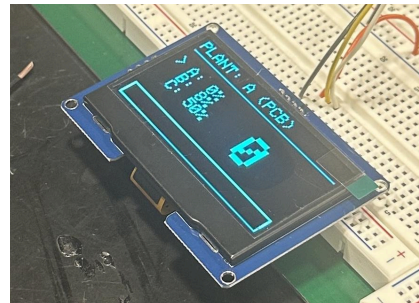


Figure 14.2: Bright condition

To test whether user input interaction will cause any unintended pump activation, we monitored the voltage at the pump MOSFET's gate as shown in Table 2 below and observed that they are all below the gate voltage threshold.

Reading #	DC Voltage (V)
1	0.009752
2	0.016709
3	0.015905
4	0.012913
5	0.013267
6	0.012716
7	0.011986
8	0.011117
9	0.010427
10	0.010577

Table 2: Voltage measure across MOSFET gate

3.3 Control Subsystem

To test if the main controller is able to store and manage data for three different plants, we turned on all sensor nodes and transmitted all sensor readings. We verified that the controller received the new

readings and updated the OLED display. The main controller stores data for each plant in a struct containing all details as Figure 15 below.

```
struct PlantNode {
    String name;
    String species;
    int moisture;
    int threshold;
    int maxPumpTime;
    BLEAdvertisedDevice* device;
    BLEClient* client;
    bool active;
};
```

Figure 15: Plant node struct

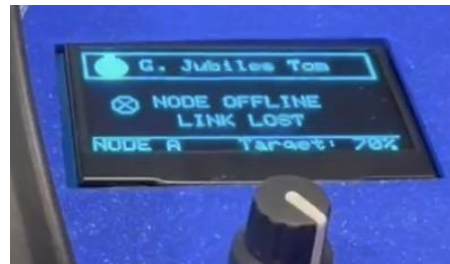


Figure 16: Node offline display

Besides, we tested if the pump will be disabled when communication with a node fails. We turned off one of the sensor nodes and pressed the button to start the watering process. However, the pump remained disabled and the display is as shown in Figure 16.

3.4 Water Dispensing Subsystem

To verify the functionality of this physical actuator block, we first tested out the pump’s flow constant rate. We programmed the pump to run for 60 seconds and measured the volume of water dispensed. The recorded pump flow rate is 57mL/min which is equivalent to 0.95 mL/s.

Furthermore, to test the accuracy of the volume of water dispensed by the pump, we programmed the pump to dispense 100mL of water and measure the actual volume of water in the beaker. This is repeated for 5 trials and we verified that all readings were within $\pm 20\%$ as shown in Table 3 below.

Entry	Reading (mL)	Deviation from 100 mL
1	103	+3
2	106	+6
3	111	+11
4	105	+5
5	104	+4

Table 3: Water volume readings

3.5 Power Subsystem

To determine stability within the power subsystem we want to maintain minimal ripple as possible. This means that we would want $V \pm 5\%$ within the rated output voltage.

Entry	LDO Voltage Output (V)
1	3.21
2	3.24
3	3.19
4	3.25
5	3.25

Table 4: Voltage LDO output for sensor node

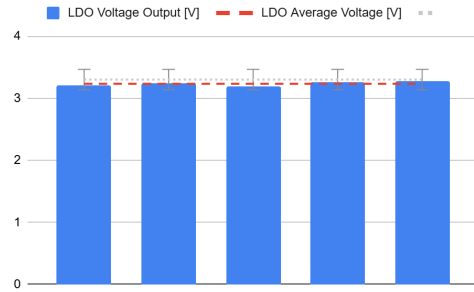


Figure 17: Voltage Output graph

In order to get the data we used a DMM and turned the mode to DC. We had triggered the pump and we had probed the voltage of the LDO output. Here we have a table and graph that contains 5 readings of the watering can PCB 3.3V LDO which is $3.3V \pm 5\%$ showing minimal deviation. The results are shown on table 4 and figure 17.

Entry	Input Voltage (V)	Voltage Output (V)
1	7.54	5.04
2	7.59	5.02
3	7.45	5.001
4	7.48	5.03
5	7.41	4.99

Table 5: Voltage Readings of Buck Converter

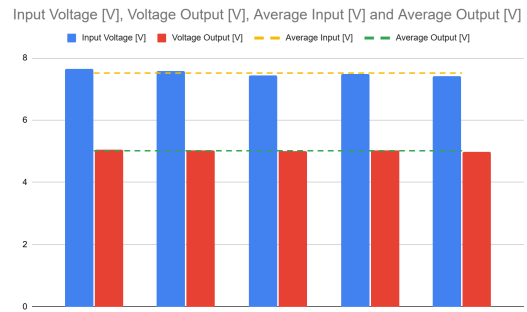


Figure 18: Voltage Output and Input of Buck

We have measured the Buck converter input and output voltage readings making sure that it has a stable output while the pump is discharging shown in figure 18 and table 5. In order to get the data we used a DMM and turned the mode to DC. Then, we had triggered the pump and during this duration we had probed the input and output voltage of the buck converter. We had used two DDM for this process.

Entry	Input Voltage (V)	Voltage Output (V)
1	4.12	3.29
2	4.09	3.301
3	3.81	3.28
4	3.67	3.29
5	3.64	3.29

Table 6: Voltage Readings of Buck-Boost Converter

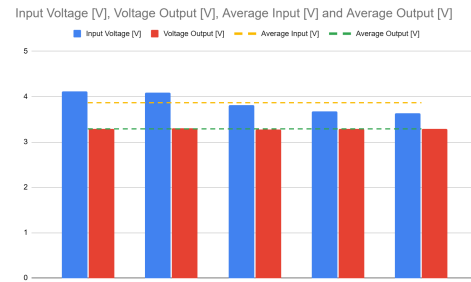


Figure 19: Voltage Output and Input of Buck-Boost

We have repeated the same procedure as the buck, except that we had the sensor running. We found that we are $3.3V \pm 5\%$ shown at table 6 and figure 19.

4. Cost and Schedule

4.1 Cost

4.1.1 Parts Cost

The table for each component used and its price is listed out in Appendix B. The total parts cost sums up to be \$228.64.

4.1.2 Labor Cost

Assuming an average hourly salary of an ECE graduate from University of Illinois to be \$45, the total hours spent on this project to be 7 hours per week with a total of 9 weeks, the labor cost will be $(\$45/\text{hour}) * 2.5 * 7 * 9 = \7087.50 per person.

Since we have three members, the total labor cost will be $\$7087.50 * 3 = \21262.50 .

4.1.3 Total Cost

Summing up $\$228.64 + \21262.50 from both parts, the total cost including labor and materials turns out to be **\$21491.14**.

4.2 Schedule

Throughout the semester, we have broken down the overall system into individual smaller tasks to be completed weekly and slowly integrated all members' parts together. The full schedule that we followed is listed in Appendix C.

5. Conclusion

5.1 Accomplishments

The final HydroFlora prototype achieved several of the main project goals. The sensor node was able to collect soil moisture data and transmit the readings wirelessly to the watering can controller over an indoor distance of at least 3 meters. The sensor node was also verified to operate for at least 24 hours on battery power, satisfying the longevity requirement. The user interface subsystem successfully displayed plant moisture information under both bright and dark lighting conditions, and the rotary encoder allowed the user to select between different plant profiles. The control subsystem was able to manage data from multiple plant nodes.

5.2 Uncertainties

One of the main concerns we had with full deployment is the dependency of accuracy on calibration. Currently the way we have it set up is that we precalibrate each sensor node manually. For large scale integration we would ideally want a more robust detection system. The pump flow rate may also vary due to battery voltage, tube wear, water level, or mechanical resistance in the tubing. Therefore, even though pump timing can be calculated from the measured flow rate, the dispensed volume may drift over time. Another uncertainty is long-term wireless reliability, especially if the system is used in a larger room, near other Bluetooth devices, or with more sensor nodes added. The 24-hour battery test confirms short-term operation, but further testing is needed to characterize battery life over multiple days or weeks.

5.3 Ethical considerations

HydroFlora was designed with safety, reliability, and environmental responsibility in mind in accordance with IEEE Std 7000-2021 [15], which provides a model process for addressing ethical concerns during system design, the project considered how stakeholder values such as safety, usability, sustainability, and reliability could be translated into engineering requirements. Since the system uses rechargeable lithium batteries and water near electronic components, proper electrical isolation, voltage regulation, and protection circuitry are important to reduce the risk of overheating, short circuits, or unsafe operation. The system also disables the pump when sensor communication fails, preventing uncontrolled watering due to missing or invalid data. From an environmental perspective, HydroFlora supports more efficient water usage by dispensing water based on measured plant needs rather than fixed schedules or guesswork. The design also avoids unnecessary automation that could mislead users into ignoring plant

care entirely; instead, it assists the user while still allowing human control through the watering can interface.

5.4 Future work

In the future we would like to explore the possibility of more advanced features for an overall cheaper cost. In this project we were using the ESP32 which led to high cost. To cut cost while still maintaining full functionality we could explore other Microcontroller Units that have a more specific capability to the scope of this project. Also in the future we would like to further develop the power management system for a more robust and safe operation.

References

- [1] IEEE, "IEEE Code of Ethics," *IEEE.org*, 2024. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8>. [Accessed: Feb. 11, 2026].
- [2] Association for Computing Machinery, "ACM Code of Ethics and Professional Conduct," *ACM.org*, 2018. [Online]. Available: <https://www.acm.org/code-of-ethics>. [Accessed: Feb. 11, 2026].
- [3] Underwriters Laboratories, "UL 1642 Standard for Lithium Batteries," *LiPol Battery*, 2023. [Online]. Available: https://www.lipolbattery.com/UL1642_Certification_of_Lithium-ion_Battery.html. [Accessed: Feb. 11, 2026].
- [4] Adafruit Industries, "Peristaltic Liquid Pump with Silicone Tubing," *DigiKey Electronics*, 2024. [Online]. Available: <https://www.digikey.com/en/products/detail/adafruit-industries-llc/1150/5638299>. [Accessed: Feb. 13, 2026].
- [5] DFRobot, "Capacitive Soil Moisture Sensor - Corrosion Resistant," *DigiKey Electronics*, 2024. [Online]. Available: <https://www.digikey.com/en/products/detail/dfrobot/SEN0193/6588605>. [Accessed: Feb. 13, 2026].
- [6] DFRobot, "Capacitive Soil Moisture Sensor v1.0," *SEN0193 Datasheet*, 2024. [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/2107/SEN0193_Web.pdf. [Accessed: Feb. 13, 2026].
- [7] Espressif Systems, "ESP32-C3 Series Datasheet," *Espressif Documentation*, v1.0, 2024. [Online]. Available: https://documentation.espressif.com/esp32-c3_datasheet_en.pdf. [Accessed: Feb. 13, 2026].
- [8] Adafruit, "Monochrome 128x64 OLED Graphic Display Module Kit," *Adafruit Industries*, 2024. [Online]. Available: <https://www.adafruit.com/product/4741>. [Accessed: Feb. 13, 2026].
- [9] Missouri Botanical Garden, "Overwatering," *Help for the Home Gardener*, 2024. [Online]. Available: <https://www.missouribotanicalgarden.org/gardens-gardening/your-garden/help-for-the-home-gardener/advice-tips-resources/insects-pests-and-problems/environmental/overwatering>. [Accessed: Feb. 13, 2026].
- [10] Perennial Wellness, "The Therapeutic Power of Indoor Gardening for Mental Wellness," *Perennial Wellness Blog*, 2023. [Online]. Available: <https://www.perennialwellnesswny.com/post/the-therapeutic-power-of-indoor-gardening-for-mental-wellness>. [Accessed: Feb. 13, 2026].
- [11] NOAA, "Environmental Stewardship Societal Impact: Common Measure Definitions," *NOAA Education Council*, 2023. [Online]. Available:

<https://www.noaa.gov/office-education/noaa-education-council/monitoring-resources/common-measure-definitions/stewardship-definitions>. [Accessed: Feb. 13, 2026].

[12] The Grainger College of Engineering, "Computer Engineering," *University of Illinois Urbana-Champaign*, 2024. [Online]. Available: <https://grainger.illinois.edu/academics/undergraduate/majors-and-minors/computer-engineering>. [Accessed: Feb. 27, 2026].

[13] Watson-Marlow Fluid Technology Solutions, "Peristaltic pumps: how they work," *W-M Support: Pump Principles*, 2024. [Online]. Available: <https://www.wmfts.com/en-us/support/pump-principles/peristaltic-pumps-how-they-work/>. [Accessed: Feb. 27, 2026].

[14] Texas Instruments, "TPS6103x Efficient Boost Converter With 4-A Switch," *TPS61030 Datasheet, SLVS454C*, 2023. [Online]. Available: <https://www.ti.com/lit/ds/symlink/tps61030.pdf>. [Accessed: Feb. 27, 2026].

[15] IEEE Standards Association, "IEEE Std 7000-2021: IEEE Standard Model Process for Addressing Ethical Concerns during System Design," IEEE, 2021. [Online]. Available: <https://standards.ieee.org/standard/7000-2021.html>. [Accessed: May 6, 2026].

Appendix A Requirement and Verification Table

1. Sensing Subsystem

Requirements	Verification	Verification status (Y or N)
<ul style="list-style-type: none"> The sensing node shall remain operational for at least 24 hours without recharge under normal conditions 	<p><u>Equipment:</u> Digital Multimeter (DMM)</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> Charge the sensor node's battery to full capacity of 4.1 V±5%. Set the DMM to current mode (mA) and put it in series with the battery. Measure the current during sensor node's transmission ($I_{transmit}$). Calculate the average current draw over 10 30s spaced sampled values $I_{average} = \frac{I_{transmit}}{10}$ <ol style="list-style-type: none"> Confirm that $I_{average}$ is less than (Battery Capacity/24h). <p><u>Presentation of Results:</u> Calculated battery life based on measured current draw</p>	Y
<ul style="list-style-type: none"> The sensing probe shall resist water-induced degradation for at least 30 days of continuous exposure. 	<p><u>Equipment:</u> DMM</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> Record the physical condition of the new sensor and measure its output voltage in both dry and wet soil condition. Submerge the probe in soil for 30 days with periodical watering. Perform weekly inspections and check for any oxidation or corrosion of the sensor probe. On day 30, record the 	Y

	<p>physical condition of the sensor and measure its output voltage in dry and wet soil and compare it to the Day 1 readings.</p> <p><u>Presentation of Results:</u> Pictures of sensor's physical condition, Table of voltage readings on Day 1 and Day 30.</p>	
<ul style="list-style-type: none"> The sensing node shall transmit a valid moisture reading within an indoor range of at least 3 meters. 	<p><u>Equipment:</u> Measuring tape, Serial monitor</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> Place the sensor node and the watering can's controller at a measured distance of 3 meters apart. Monitor the serial output of the watering can's controller for incoming packets. Confirm that 10 consecutive readings are received successfully without corruption. <p><u>Presentation of Results:</u> Table of success rate of readings at 3 meters</p>	Y

2. User Interface Subsystem

Requirements	Verification	Verification status (Y or N)
<ul style="list-style-type: none"> The display shall show the current moisture percentage and selected 	<p><u>Equipment:</u> Visual inspection</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> Place the display in two different room settings. Trigger the sensor's moisture update by adding water to the soil. 	Y

<p>plant name under typical indoor lighting conditions.</p>	<p>3) Verify that the current moisture percentage and plant name are clearly visible from a 0.5m distance.</p> <p><u>Presentation of Results:</u> Confirmation of visual legibility</p>	
<ul style="list-style-type: none"> The subsystem shall not cause unintended pump activation during input interaction. 	<p><u>Equipment:</u> DMM</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Connect DMM to the Pump MOSFET Gate pin. 2) Using the rotary knob, cycle through at least 3 different plant profiles at both slow and fast pace. 3) Monitor the DMM to see if the voltage stays at $0V \pm 0.1V$. <p><u>Presentation of Results:</u> A voltage log during user input interaction</p>	<p>Y</p>

3. Control Subsystem

Requirements	Verification	Verification status (Y or N)
<ul style="list-style-type: none"> The controller shall store and manage data for at least 3 independent sensor nodes. 	<p><u>Equipment:</u> Serial monitor</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Power on the watering can and all 3 sensor nodes. 2) Select each node one by one. 3) Verify on the display and serial monitor the unique ID and current moisture reading for each of the 3 nodes. <p><u>Presentation of Results:</u> Table showing each node IDs and their moisture readings before</p>	<p>Y</p>

	and after restarting the watering can controller.	
<ul style="list-style-type: none"> Wireless reception shall reliably operate within 3m indoor range. 	<p><u>Equipment:</u> Measuring tape, Serial monitor</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> Place the sensor node and the watering can's controller at a measured distance of 3 meters apart. Monitor the serial output of the watering can's controller for incoming readings. Confirm that 10 consecutive readings are received successfully without corruption. <p><u>Presentation of Results:</u> Table of success rate of readings at 3 meters</p>	Y
<ul style="list-style-type: none"> If communication with a selected node fails, the pump shall remain disabled. 	<p><u>Equipment:</u> DMM</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> Select a plant profile on the watering can. Power off the corresponding sensor node to simulate a communication failure. Attempt to trigger the watering action via user interface. Measure the voltage at the pump's MOSFET Gate pin with DMM. Confirm that the voltage remains at $0V \pm 0.1V$. Repeat for all plant profiles. <p><u>Presentation of Results:</u> Confirmation of $0V \pm 0.1V$ output during the simulated link failures.</p>	Y

4. Water Dispensing Subsystem

Requirements	Verification	Verification Status (Y or N)
<ul style="list-style-type: none"> The dispensed volume should be stable around plus or minus 20 percent volume accuracy. 	<p><u>Equipment:</u> Beaker</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Program the system to dispense 100mL of water 2) Collect water from the pump in the beaker 3) Repeat for 5 times 4) Confirm that all trials fall within $\pm 20\%$ of 100mL. <p><u>Presentation of Results:</u> Table of volume of water dispensed across all 5 trials</p>	Y
<ul style="list-style-type: none"> Water shall begin flowing within $3s \pm 1s$ of activation. 	<p><u>Equipment:</u> Stopwatch, beaker</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Position the pump tubing in a beaker. 2) Trigger the pump and start the stopwatch simultaneously. 3) Stop the stopwatch once the first drop of water enters the beaker. 4) Record the time taken on stopwatch and confirm that it is within $3s \pm 1s$. 5) Repeat for 5 times. <p><u>Presentation of Results:</u> Table of time taken across 5 trials</p>	Y
<ul style="list-style-type: none"> The subsystem shall operate continuously 	<p><u>Equipment:</u> Infrared thermometer</p>	Y

<p>for up to 60 seconds without overheating or shutdown.</p>	<p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Run the pump for a continuous 60-second cycle. 2) Immediately measure the temperature of the MOSFET and the pump motor. 3) Confirm that the temperature of the pump is below 40 degrees Celsius. <p><u>Presentation of Results:</u> Table showing recorded temperature of components after 60s</p>	
--	--	--

5. Power Subsystem

Requirements	Verification	Verification status (Y or N)
<ul style="list-style-type: none"> • The logic supply voltage shall remain within plus or minus 5 percent of nominal (3.3V) during pump activation. 	<p><u>Equipment:</u> DMM</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Connect the probe to the 3.3V logic rail. 2) Trigger the pump. 3) Monitor the DMM for any sag or surge. 4) Verify the voltage stays within $\pm 5\%$ of 3.3V during pump activation. <p><u>Presentation of Results:</u> DDM trace showing 3.3V stability under load</p>	<p>Y</p>
<ul style="list-style-type: none"> • The buck converter shall step down the (7.4-8.2V) battery voltage to a stable $5.0V \pm 5\%$ to ensure the peristaltic pump maintains a 	<p><u>Equipment:</u> DMM, battery</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Connect the 7.4V battery to the input of the buck converter. 2) Measure the output voltage of the buck converter with DMM. 	<p>Y</p>

<p>consistent flow state regardless of battery discharge level.</p>	<p>3) Record the readings and confirm it is within $5.0V \pm 0.1V$.</p> <p><u>Presentation of Results:</u> Table showing the input voltages and output voltages of the buck converter</p>	
<ul style="list-style-type: none"> The buck-boost converter shall step down the (3.7-4.2V) battery voltage to a stable $3.3V \pm 5\%$ to ensure the microcontroller is powered 	<p><u>Equipment:</u> DMM, battery</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Connect the 3.7V battery to the input of the buck-boost converter. 2) Measure the output voltage of the buck-boost converter with DMM. 3) Record the readings and confirm it is within $3.3V \pm 0.1V$. <p><u>Presentation of Results:</u> Table showing the input voltages and output voltages of the buck-boost converter</p>	<p>Y</p>

Appendix B Parts Cost Table

Description	Manufacturer	Part #	Quantity	Unit Price	Extended Price
RF TXRX MODULE BT PCB TRACE SMD	Espressif Systems	ESP32-S3-WRO0 M-1-N16	4	\$5.92	\$23.68
Peristaltic Liquid Pump with Silicone Tubing - 5V to 6V DC Power	Adafruit Industries LLC	3910	1	\$24.95	\$24.95
Capacitive Soil Moisture Sensor	DFRobot	SEN0193	3	\$5.90	\$17.70
3.7V 1000mAh LiPo Battery Rechargeable 4 Pack	MakerHawk	102050	1	\$18.99	\$18.99
1-A Single Cell Li-Ion Battery Linear Charger	3PEAK	TPB4056B2X-ES1 R	3	\$0.50	\$1.50
MOSFET N-CH 30V 5.3A SOT23	Infineon Technologies	IRLML0030TRPBF	1	\$0.50	\$0.50
DIODE SCHOTTKY 40V 5A DO214AC	Comchip	CDBA540-HF	1	\$0.45	\$0.45
IC REG LINEAR 3.3V 1A SOT-223-3L	UMW	LD1117-3.3	4	\$0.30	\$1.20
HiLetGo 2.42" 128x64 OLED Graphic Display	HiLetGo	4741	1	\$22.50	\$22.50
ENCODER MECH QUAD VERT PC PIN	Adafruit	377	1	\$4.50	\$4.50
IC REG BOOST ADJ 3.6A 16HTSSOP	Texas Instruments	TPS61030PWP	1	\$3.86	\$3.86
CONN RCPT USB2.0 MICRO B SMD R/A	Amphenol	Micro USB-B	1	\$0.49	\$0.49
CAP CER 10UF 50V X7R 0805	SAMSUNG	10uF	1	\$0.11	\$0.11
CAP CER 4.7UF 50V X7R 0805	YAGEO	4.7uF	2	\$0.38	\$0.76
CAP CER 0.1UF 50V X7R 0805	YAGEO	0.1uF	2	\$0.10	\$0.20
LED GREEN CLEAR CHIP SMD	LITE-ON	GREEN LTST-C171GKT	2	\$0.11	\$0.22
LED RED CLEAR 1206 SMD	LITE-ON	RED LTST-C150CKT	2	\$0.15	\$0.30
RES 5.1K OHM 5% 1/8W 0805	Stackpole Electronics	RMCF0805JT5K1 0	3	\$0.10	\$0.30

SWITCH TACTILE SPST-NO 0.05A 12V	C&K	Tactile	2	\$0.36	\$0.72
Watering Can	Etarewilt	-	1	\$12.99	\$12.99
BATTERY LITH-ION 3.7V 2AH	2011	1528-1857-ND	3	\$12.50	\$37.50
CONN HEADER VERT 2POS 2MM	B2B-PH-K-S	455-1704-ND	3	\$0.11	\$0.33
2A DC/DC-CONVERTER 'INNOLINE' SI	R-78K5.0-2.0	945-R-78K5.0-2.0- ND	1	\$5.21	\$5.21
PTC RESET FUSE 16V 1.1A 1206	1206L110/16WR	18-1206L110/16 WRCT-ND	1	\$1.09	\$1.09
XT60 CONNECTORS MALE/FEMALE PAIR	10474	1568-10474-ND	1	\$2.50	\$2.50
GOLDBAT 2S Lipo Battery 5000mAh 50C 7.4V Pack Hardcase with T Plug for RC Evader BX Car, Traxxas Slash Buggy RC Truck, RC Truggy RC Airplane UAV Drone FPV	GOLDBAT	9067160597	1	\$14.99	\$14.99
IC REG BUCK BST 3.3V 1.6A 10VSON	Texas Instruments	TPS63001DRCR	3	\$2.44	\$7.32
IC REG BUCK 3.3V 1A 8WSON	Texas Instruments	TPS62162DSGR	1	\$1.57	\$1.57
LOW-POWER MONITOR AND PROTECTOR	Texas Instruments	296-BQ76905RG RRCT-ND	1	\$1.72	\$1.72
IC OR CTRLR SRC SELECT 8SON	Texas Instruments	TPS2113ADRBR	4	\$2.28	\$9.12
IC BAT BALANCER LI-ION 2C 24VQFN	Texas Instruments	BQ25887RGER	1	\$5.00	\$5.00
CONN HEADER VERT 3POS 2.5MM	JST Sales America Inc.	B3B-XH-A	1	\$0.13	\$0.13
TERM BLK 2P SIDE ENT 5.08MM PCB	Phoenix Contact	277-1247-ND	1	\$1.33	\$1.33
BATTERY PROTECTION IC	ABLIC Inc.	1662-S-8252ACI- M6T1UCT-ND	1	\$1.80	\$1.80
RES 5.1K OHM 1% 1/10W 0603	YAGEO	311-5.10KHRCT- ND	10	\$0.10	\$1.00
RES 100 OHM 1% 1/16W 0402	YAGEO	311-100LRCT-ND	1	\$0.10	\$0.10
RES SMD 3K OHM 0.1% 1/16W 0402	Panasonic Industry	P3.0KDCCT-ND	1	\$0.11	\$0.11
RES 500K OHM 1% 300MW 1206	Vishay Dale	3579-CRMA1206 AF500KFKEFCT- ND	1	\$0.54	\$0.54

RES 10K OHM 1% 1/10W 0603	YAGEO	311-10.0KHRCT- ND	4	\$0.10	\$0.40
RES 470 OHM 1% 1/16W 0402	YAGEO	311-470LRCT-ND	2	\$0.10	\$0.20
RES 1K OHM 1% 1/16W 0402	YAGEO	311-1.00KLRCT-N D	2	\$0.10	\$0.20
THERMISTOR NTC 10KOHM 3977K BEAD	Vishay Beyschlag/Dralori c/BC Components	NTCLE100E3103 JB0	1	\$0.56	\$0.56
				Total	\$228.64

Appendix C Schedule

WEEK	HIGH LEVEL DESCRIPTION
1: 3/2 - 3/8	<p>Design Review - Monday</p> <ul style="list-style-type: none"> - Hardware <ul style="list-style-type: none"> - Send in non E-shop components order form - Solder and test First PCB board (if arrived) - Update Control Unit PCB for second round order - Complete and Send in PCB design for sensor node - Software <ul style="list-style-type: none"> - Prototype BLE data communication using Dev board - Drive a simple display
2: 3/9 - 3/15	<p>Breadboard Demo</p> <ul style="list-style-type: none"> - Hardware <ul style="list-style-type: none"> - Solder and test Both PCB board - Update Control Unit PCB Design for third round order <ul style="list-style-type: none"> - Update to include Rechargeable LiPo - Complete and Send in PCB design for both PCB board - Define the PCB dimensions for PCB housing and pass dimensions to machine shop - Software <ul style="list-style-type: none"> - Verify BLE data communication ESP-ESP <ul style="list-style-type: none"> - Try out Wi-Fi communication to see Power Tradeoff - Drive a simple display and display communicated data <p>BREADBOARD DEMO ONE Description</p> <ul style="list-style-type: none"> - Demo ESP-ESP communication - Demo Displaying communication data <p>Ideal Case - breakout PCB arrived and working Minimal Case - Functional requirements as above using dev board</p>
3: 3/16 - 3/22	<p style="text-align: center;">SPRING BREAK</p>
4: 3/23 - 3/29	<ul style="list-style-type: none"> - Hardware <ul style="list-style-type: none"> - Solder and test Both PCB board (if arrived) - Update Both PCB board for FINAL round order <ul style="list-style-type: none"> - Mainly reducing breakouts for compactness - Verify power Unit and see if need to swap different boost converter - Complete and Send in PCB design for both PCB board - Software <ul style="list-style-type: none"> - Generate sensor reading and send to Main MCU - Display data on simple display

5: 3/30 - 4/5	<p>Individual progress reports</p> <ul style="list-style-type: none"> - Hardware <ul style="list-style-type: none"> - Solder and test Both PCB board - Debugging <ul style="list-style-type: none"> - Test Voltage Output over time - Test Water Volume being Output by peristaltic pump - Software <ul style="list-style-type: none"> - Port previously developed code and try running on PCB MCU <ul style="list-style-type: none"> - Develop communication pipeline for Moisture sensor (On Chip -> Wireless transmit) - Debugging
6: 4/6 - 4/12	<p>Progress Demo</p> <ul style="list-style-type: none"> - Hardware <ul style="list-style-type: none"> - Solder and test Both PCB board - Debugging - Software <ul style="list-style-type: none"> - Test code and try running on PCB MCU - Debugging - Integrated Testing <ul style="list-style-type: none"> - Compare Actual water volume outputted vs Ideal water outputted - Compare Sensor node reading vs Control Unit reading.
7: 4/13 - 04/19	<ul style="list-style-type: none"> - Hardware <ul style="list-style-type: none"> - Solder and test Both PCB board - Debugging - Software <ul style="list-style-type: none"> - Test code and try running on PCB - Debugging - Refinement <ul style="list-style-type: none"> - Refine the code based on previous actual values. <ul style="list-style-type: none"> - ie non ideal buffering/ error correction
8: 4/20 - 4/26	<p>Mock demo & Mock Presentation</p> <ul style="list-style-type: none"> - Hardware <ul style="list-style-type: none"> - Debugging - Software <ul style="list-style-type: none"> - Debugging
9: 4/27 - 5/3	<p>Final demo & Final Presentation</p> <ul style="list-style-type: none"> - Work on professional packaging - Prepare for demo and presentation
10: 5/4 - 5/7	<p>Final papers</p>