

UCF Senior Design 2 Final Report

Functioning Lighting and Optimal Watering for Environmental Regulation System (F.L.O.W.E.R.S.)



Group 20

James Ventimiglia
*Electrical
Engineering*

Noah Weeks
*Electrical
Engineering*

Ashley Hanzelka
*Computer
Engineering*

Junior Jestine
*Electrical
Engineering*

Reviewers:

Dr. Sonali Das
Dr. Mike Borowczak
Dr. Saleem Sahawneh

Mentors:

Dr. Chung Yong Chan



COLLEGE OF
ENGINEERING &
COMPUTER
SCIENCE

Table of Contents

1.0 Executive Summary	1
2.0 Project Description	2
2.1 Background and motivation	2
2.2 Existing Products	3
2.3 Goals and Objectives	5
2.4 Features and Functionalities	7
2.5 Specifications	10
2.6 House of Quality	12
2.7 Block Diagrams	13
3.0 Research	14
3.1 Technology Comparison	14
3.1.1 Moisture Sensing Technology	15
3.1.1.1 Volume Water Content Sensor	15
3.1.1.2 Soil Water Tension Sensor	16
3.1.1.3 Single-Point Moisture Sensor	16
3.1.1.4 Capacitive Sensing	17
3.1.1.5 Moisture Sensing Technology Selection	19
3.1.2 Lighting	20
3.1.2.1 LEDs	21
3.1.2.2 Fluorescent	21
3.1.2.3 Incandescent	22
3.1.2.4 Infrared	22
3.1.2.5 Lighting Selection	23
3.1.3 LED Drivers	24
3.1.3.1 Dimming Technologies	25
3.1.3.2 LED Driver Comparison	26
3.1.3.3 Philips Hue Lightstrip Plus LED Driver	26
3.1.3.4 WLED for ESP32	26
3.1.3.5 Shelly RGBW2	27
3.1.3.6 LED Driver Conclusion Part Comparison Conclusion	27
3.1.4 Light Sensing Technology	28
3.1.4.1 Photodiode	28
3.1.4.2 Photoresistor	28
3.1.4.3 Phototransistor	29
3.1.4.4 Light Sensing Selection	30
3.1.4.5 Photodiode part selection	30

3.1.4.6 Adafruit BH1750 Ambient Light Sensor	31
3.1.4.7 Adafruit VEML7700 Ambient Light Sensor	31
3.1.4.8 TSL2591	31
3.1.4.9 Photodiode Selection	32
3.1.5 Soil Types	32
3.1.5.1 Potting Mix	33
3.1.5.2 Garden Soil	33
3.1.5.3 Succulent Soil	33
3.1.6 Water Delivery to Plants: Pumps	34
3.1.6.1 Types of Water Pump Technology	34
3.1.6.2 Diaphragm Pumps	34
3.1.6.3 Peristaltic Pumps	35
3.1.6.4 Centrifugal Pumps	35
3.1.6.5 Continuous Flow Pumps	36
3.1.6.6 Demand Pumps	36
3.1.6.7 Water Pump Part Comparison	36
3.1.6.8 Water Pump Constraints	37
3.1.7 Water Delivery to Plants: Solenoid Valves	38
3.1.7.1 Solenoid Valve Types	38
3.1.7.2 Direct Wired Solenoid Valves	39
3.1.7.3 Wireless Controlled Solenoid Valves	39
3.1.7.4 Solenoid Valve Actuation using wireless LED driver	39
3.1.7.5 Solenoid Valve Part Comparison	40
3.1.7.6 Solenoid Valve Constraints	40
3.1.8 MCU Parts Comparison	41
3.1.8.1 ESP32-Wroom-32	42
3.1.8.2 ATmega328P	42
3.1.8.3 MSP430	43
3.1.8.4 RP2040	43
3.1.9 Total System Energy Sources	43
3.1.9.1 Lead-Acid Battery	43
3.1.9.2 Lithium-Ion Battery	44
3.1.9.3 CR2032 Coin Cell Battery	45
3.1.9.4 PC IFR18650-1500 Battery	46
3.1.9.5 Line Power	46
3.1.9.6 Power Selection Table	47
3.1.9.7 Power DistributionTable	48
3.1.10 Water Delivery to Plants: Pump and Valves Power Source	48

3.1.10.1 AC to DC Converters	48
3.1.10.2 AC to DC Technology Comparisons	49
3.1.10.3 AC to DC Converter Part Comparison and Selection	50
3.1.10.4 DC to DC Converters	52
3.1.10.5 Linear Regulators	52
3.1.10.6 Switching Regulators	53
3.1.10.7 DC to DC converter Part Comparison	54
3.1.10.7.1 TPS613221A	54
3.1.10.7.2 LM2576	55
3.1.10.7.3 LTC3703	55
3.1.11 Software Development Technology	56
3.1.11.1 Embedded Software Technology Comparisons	57
3.1.11.2 App Development Technology Comparisons	59
3.1.11.3 Storage in the Mobile App	63
3.1.12 PCB Design Software	64
3.1.12.1 Fusion360	64
3.1.12.2 KiCad	65
3.1.12.3 Altium	65
3.1.12.4 PCB Software Selection	65
3.1.13 Temperature/Humidity Sensing Technology	66
3.1.13.1 Thermistor	66
3.1.13.2 Semiconductor-based	66
3.1.13.3 ADT 20	67
3.1.13.4 DHT 11	67
3.1.13.5 SHT 45	68
3.1.14 Communication Protocol	68
3.1.14.1 I2C	68
3.1.14.2 SPI	68
3.1.14.3 UART	69
4.0 Standards and Design Constraints	69
4.1 Standards	69
4.1.1 Health and Safety	69
4.1.1.1 Health and Safety Standards	70
4.1.1.2 Electrical & Safety Standards	70
4.1.1.3 PCB Design Standards	71
4.1.1.3.1 – Overview of IPC-2221 Design Standards	71
4.1.1.4 Wireless Communication & Connectivity Standards	74
4.1.1.4.1 Wireless Communication via the IEEE 802.11 Standard	76

4.1.1.4.2 Wireless Communication via ESP-NOW	78
4.1.1.5 Data Security & Privacy Standards	78
4.1.1.6 Smart Lighting Component Standards	79
4.1.1.7 Sustainability & Energy Efficiency Standards	79
4.1.1.8 Sensor & Environmental Standards	79
4.2 Constraints	80
4.2.1 Sustainability Constraints	80
4.2.1.1 Battery Life	80
4.2.2 Maximizing Battery Life	81
4.2.1 Maintenance and Reliability Constraint	81
4.2.1.1 Corrosion	81
4.2.1.1.1 Electrolytic Corrosion	82
4.2.1.1.2 Galvanic Corrosion	82
4.2.1.1.3 Corrosion Prevention	83
4.2.1.2 Salinity	84
4.2.3 Environmental Constraints	84
4.2.4 Security Constraints	85
5.0 Comparison of ChatGPT with Other AI	86
5.1 General Generative AI Assistance	86
5.2 Project Specific AI Generated Assistance	87
5.2.1 Example Prompt 1	87
1. Peristaltic Pump	87
2. Diaphragm Pump	88
3. Centrifugal Pump	88
Recommendation for Your Case-	89
5.2.2 Example Prompt 2	89
Shelly RGBW2	90
WLED (on ESP8266/ESP32)	91
Recommendation for This Project	92
5.3 AI Platform Comparisons	92
5.3.1 AI Comparison Example 2	93
5.5.3.1 ChatGPT	93
5.5.3.2 DeepSeek	93
5.5.3.3 Claude	94
5.5.4 AI comparison Conclusion	95
6.0 Hardware Design	95
6.1 MCU PCB Schematic	95
6.1.1 ESP32	95

6.1.2 Regulators	96
6.1.3 I2C Multiplexer	97
6.1.4 Sensors	98
6.2 Water Pump Subsystem Block Diagram	99
6.3 Moisture Sensor Block Diagram	101
6.3.1.1 Voltage Regulation	103
6.3.1.2 555 Timer Pulse Generation	103
6.4 Lighting Subsystem Block Diagram	104
6.5 Shelly RGBW2 Driver Schematics	106
6.6 Overall Schematic Design	109
7.0 Software Design	110
7.1 Embedded Design	110
7.1.1 Technologies	110
7.1.2 Soil Moisture Sensors	111
7.1.3 Main System MCU Design	112
7.2 User Application Design	114
7.2.1 Technologies	114
7.2.2 Mobile UI Design	115
7.3 Hard-Coded Values	120
7.3.1 Constant Values in the Soil Moisture Sensor Readings	120
7.3.2 Constant Values for ESP32 Broadcasting	120
8.0 System Fabrication	121
8.1 PCB Layout	121
8.1.1 3.3V Regulator Board	122
8.1.2 ESP32 Board	123
8.1.3 VEML7700	123
8.1.4 SHT45	124
8.1.5 Capacitive Moisture Sensor	125
8.2 Plant Shelving	126
8.2.1 Shelf Fabrication	126
9.0 System Testing and Evaluation	127
9.1 ESP32 Testing	127
9.2 Moisture Sensor Testing	128
9.3 Humidity/Temp Sensor Testing	128
9.4 Water Pump/Solenoid Valve Testing	129
9.5 Light System Testing	129
9.6 SD2 Testing Plan	130
10.0 Administrative Content	131

10.1 Budget and Financing	131
10.2 BOM	132
10.3 Project Milestones/Schedule	133
10.4 Work Distribution	135
11.0 Conclusion	135

List of Tables

Table 1: Engineering Specifications	11
Table 2: Types of Moisture Sensors	19
Table 3: Lighting Comparison	23
Table 4: Dimming technology	26
Table 5: LED driver selection	27
Table 6: Comparison of different light sensing technologies	30
Table 7: Photodiode part selection	32
Table 8: Soil Type Comparison	34
Table 9: Water Pump Technnology Comparison	37
Table 10: Water Pump Part Comparison	38
Table 11: Solenoid Valve Part Comparison	41
Table 12: MCU Comparison	41
Table 13: AC to DC Converter Technology Comparison	49
Table 14: AC to DC Power Converter Selection	51
Table 15: Regulator Part Selection	55
Table 16: Comparison of Embedded Language Options for the ESP32-WROOM	57
Table 17: Comparison of IDEs	58
Table 18: Basic Comparison of App Development Technologies	59
Table 19: In-Depth Advantages/Disadvantages of Each Technology	60
Table 20: Basic Camparison of App Storage Technologies	63
Table 21: Local Storage Options in Flutter	63
Table 22: Communication Protocol Selection Table	69
Table 23: IPC-2221 Specification Category and Description Table	73
Table 24: Electrolytic and Galvanic Corrosion	83
Table 25: BOM table	132
Table 26: Budget Table	133
Table 27: Senior Design 1 Tentative Schedule	133
Table 28: Senior Design 2 Tentative Schedule	134

List of Figures

Figure 1: House of Quality.....	12
Figure 2: Hardware Block Diagram.....	13
Figure 3: Overall Embedded Software Flowchart.....	14
Figure 4: EMF between two capacitor plates [2].....	18
Figure 6: The Shelly RGBW2.....	25
Figure 5: Diaphragm Pump Model.....	35
Figure 7: MCU and USB-UART Schematic.....	96
Figure 8: Regulator Schematic.....	97
Figure 9: I2C Multiplexer.....	98
Figure 10: Sensor Schematic.....	99
Figure 11: Water Subsystem Block Diagram.....	100
Figure 12: Moisture sensor board schematic.....	102
Figure 13: Lighting Subsystem Block Diagram.....	105
Figure 14: Shelly Driver Schematic 1.....	107
Figure 15: Shelly Driver Schematic 2.....	108
Figure 16: Overall PCB Schematic.....	109
Figure 17: Soil Moisture Embedded Software Flowchart.....	111
Figure 18: Main System MCU Software Flowchart.....	112
Figure 19: Soil Moisture Embedded Software In-Depth Flowchart.....	113
Figure 20: Access Point Process for WiFi Connection.....	114
Figure 21: General App Logic Flowchart.....	115
Figure 22: Home Page with Hot Keys Explained.....	116
Figure 23: General Mobile App Walkthrough.....	117
Figure 24: Activity Diagram for the Mobile Component.....	118
Figure 25: Class Diagram for the Mobile Component.....	119
Figure 26: Use-Case Diagram for the Mobile Component.....	119
Figure 27: 3.3V Regulator PCB Layout.....	122
Figure 28: ESP32 PCB Layout.....	123
Figure 29: VEML 7700 Layout.....	124
Figure 30: SHT 45 Layout.....	124
Figure 31: Capacitive Moisture Sensor Layout.....	125

1.0 Executive Summary

This project centers on the creation of a smart, semi-automated system for plant care, designed to simplify the way users grow their plants. With a focus on accessibility, customization, and reliability, the system allows users to grow a variety of plants in different soil types, while automatically responding to environmental conditions through built-in sensors and programmable components.

At the heart of the project is the ability to monitor soil moisture across multiple zones. Junior was responsible for designing and implementing the wireless moisture sensing system, which uses compact sensor nodes to measure soil hydration levels and relay that information back to a central controller. This ensures that each plant receives the appropriate amount of water based on its specific needs and soil type, helping conserve water and improve plant health.

To physically deliver the water, Noah led the design of the pump and valve system. A 12V pump supplies water through a network of electronically controlled valves, which open only when needed. This method reduces energy usage and ensures efficient irrigation, especially important when managing several plants with differing water requirements.

Lighting, another key factor in plant growth, was handled by James. He developed a system of LED light strips that can be adjusted in brightness and color to mimic optimal sunlight conditions. These lights are controlled electronically, responding both to ambient light levels and to user preferences, creating a dynamic environment for each plant type.

To bring everything together in a user-friendly way, Ashley created a custom app that allows users to tell the system exactly how much water and what kind of lighting their plant should receive. Through a clean interface, users can input details about their plant's needs, and the system will adjust accordingly—eliminating the guesswork from plant care.

All components are supported by custom-designed circuit boards, which were built using collaborative CAD tools to ensure clean integration between hardware and software. The boards are optimized for low power usage, enabling long-lasting operation for the sensor nodes and efficient energy management for the system as a whole.

Altogether, this project demonstrates how thoughtful engineering and collaboration can come together to solve everyday problems in creative, efficient ways. By combining automated sensing, controlled actuation, and user input,

the system delivers a balanced solution for smart plant care that is both technically sound and easy to use.

2.0 Project Description

2.1 Background and motivation

Plants are essential to life, providing the foundation for the food we consume and the air we breathe, all while enhancing the well-being of our environment itself. They contribute to biodiversity, the reduction of carbon dioxide levels, and overall create a healthy outside environment. Therefore, it is no surprise that they have been integrated as an indoor addition, bringing these benefits directly into homes. Indoor plants provide an aesthetic value that improves the overall look of a living space, while also enhancing air quality by filtering toxins and producing oxygen. This synergy of ecological benefits and visual appeal makes plants an ideal choice for home decor. Beyond their ecological value, plants offer numerous benefits to individuals. Gardening and plant care are considered fulfilling and productive activities that contribute positively to mental health. Numerous studies have shown that spending time with plants can lead to improved well-being, including a reduction in stress and anxiety. Nurturing plants can foster a sense of accomplishment, promote relaxation, and even serve as a form of therapeutic remedy, providing both emotional and psychological healing.

Despite the well-documented benefits of plants, many individuals are hesitant to engage in plant care or gardening tasks. Those with busy schedules, limited knowledge of plant care, or lack of confidence in their gardening abilities often find the challenges of keeping plants alive daunting. There is a common misconception that successful gardening requires a “green thumb” – an innate ability to understand and care for plants effortlessly. However, plant care is actually a skill that must be learned and practiced, just like any other aspect of self-care or personal development. For individuals with demanding schedules, keeping up with the watering, lighting, and overall care that plants require can quickly become overwhelming. The pressure to maintain healthy plants often leads people to avoid plant care altogether or to experience repeated setbacks that discourage them from trying again. These setbacks can result in the loss of the intended health benefits plants bring to a home, as well as the failure to achieve the desired aesthetic appeal. As a result, many people turn to artificial alternatives, such as plastic plants, which can mimic the visual aspects but fall short in delivering the many health benefits that live plants provide, such as improved air quality.

Our team, consisting of four college students who balance rigorous engineering coursework with jobs and extracurricular activities, has personally encountered these challenges. While we each appreciate the idea of incorporating real plants into our homes for both their beneficial and aesthetic aspects, our busy schedules have discouraged us from successfully maintaining plants. The time required to properly care for plants, combined with the frustration of watching them struggle or fail to thrive, makes it easy to understand why many people forgo the joys of plant care. Moreover, the added pressure of restarting or compensating for dead or neglected plants only exacerbates the feelings of defeat. However, despite the frustration, we all recognize the value plants can add to a living space and the benefits they bring to mental and physical well-being. This personal experience has inspired our project, which seeks to overcome these barriers by automating the more complex aspects of garden maintenance.

This project aims to address these challenges by creating a smart garden system that automates critical plant care tasks. With our system, users can enjoy the benefits of maintaining live plants without the burden of constant attention. Tasks such as watering, soil moisture monitoring, and ensuring plants receive optimal light exposure will be automated, making plant care more accessible and less time-consuming. Our goal is to bridge the gap between technology and nature, enabling individuals to enjoy the positive effects of indoor plants without the frustration of trying to juggle them with other life responsibilities. By creating a more user-friendly experience, this smart garden project will empower individuals to embrace the rewards of plant care with greater ease, confidence, and satisfaction, helping them to enhance their living spaces with minimal effort.

2.2 Existing Products

There are a few existing products that try to tackle a similar concern as our smart garden, however, these projects tend to have great diversity in implementation, as environments can integrate differing combinations of light exposure, RGB LED spectrum altering, watering systems, multiple/single simultaneous environments, heating elements, etc. However, there are notably quite few environments that offer the ability for the user to change and specify the parameters for plants that have critically different demands in regards to light and soil requirements. This lack of adaptability in existing solutions highlights a gap in the market, particularly for gardeners or users with specific plant needs, such as those requiring a precise combination of soil pH, humidity, or light exposure. As plant care becomes increasingly tailored to the needs of particular

species, a flexible solution allowing for these adjustments is increasingly important.

Marketed products are vast, with a multitude of companies and approaches, offering solutions for varying gardening environments. For example, the 'Click & Grow' company offers multiple similar products along this scope, with products like the "Smart Garden 9" and the "Smart Garden 3." These products provide automatic watering and lighting services for the plants within, aiming to simplify the gardening process. While the "Smart Garden 9" offers increased space to allow different plants to cohabit the same environment, there is no separation or regulation to maintain the growth needs of plants with notably different nutrition, watering, and heat requirements. Each plant in the Smart Garden receives the same type of light exposure and watering, which can be problematic for species with significantly differing needs. The system does feature reusable pots with soil mixtures designed for general use, but it lacks the ability to individually adjust for plant-specific conditions. This is a limitation for anyone growing a diverse array of plants with varying care requirements.

Another similar product, the 'LetPot' "Hydroponics Smart Growing System," offers a closer correlation to the aspects of our project plan. As one of Amazon's top sellers and a top-rated product in this category, it features directed water pumps, customizable light spectrums, and support for multiple plants. The system allows users to adjust the light intensity, change the LED spectrum to cater to specific growth needs (for vegetables, flowers, etc.), and monitor water volume and power consumption. The product also includes a convenient app interface that helps manage these features. However, despite its advantages in user control and ease of use, the 'LetPot' system still does not allow for multi-environment support. All the plants share the same soil and watering system, which presents the same challenge as the "Smart Garden 9" in terms of plant care. Even though 'LetPot' is priced lower and can house the same number of plants as the 'Click & Grow' product, it does not offer the same level of customization that our system intends to provide.

What sets our project apart from both the "Click & Grow" and "LetPot" systems is its inherent flexibility. By allowing users to control and specify environmental conditions for each plant individually, it addresses the issues posed by the more generic solutions in the market. Our system would provide the ability to adjust parameters such as lighting, moisture levels, and even temperature for each plant, supporting a more personalized approach to indoor gardening. Additionally, the integration of multi-environment support would allow users to grow plants with vastly different care requirements in the same system, offering a level of customization that current products lack. With the ability to program

multiple plant zones with varying light and moisture conditions, our system provides an innovative solution to the limitations seen in existing products.

2.3 Goals and Objectives

We expect our design to alleviate the problems mentioned above; specifically, our design will provide a minimal maintenance solution to growing plants indoors, using plant pots or small planters. These plant pots will be stacked vertically on shelves for space saving reasons. Maintenance requirements will be minimized in order to prioritize user's upkeep as well. Therefore, the goals for our project are as follows:

Project Goals

- ❖ **Goal One:** To build an optimized and efficient indoor growing environment for plants, that takes into account a given species' ideal water and light intake
 - Basic Goal: To deliver water to the plants in accordance with the ideal amount of water for that particular species
 - Basic Goal: To provide sufficient spectral power density to the plants to ensure optimal growth
 - Basic Goal: To measure the soil moisture and soil temperature at the plant pot for feedback control
 - Advanced Goal: Use a programmable light that can output at a variety of frequencies for optimum plant growth
 - Advanced Goal: To send the collected data through a wireless bluetooth connection to the MCU
 - Advanced Goal: To utilize duty cycle control and extended on/off periods for the LED lights to simulate day and night cycles
 - Stretch Goal: The moisture/temperature sensor should be such low power that the battery on the PCB only needs to be replaced twice a year, assuming constant use
- ❖ **Goal Two:** To develop a user friendly app that will allow the user to manually input ideal growth conditions for a given plant
 - Basic Goal: Allow the user to input the ideal soil moisture level and lighting values

- Basic Goal: Allow the user to see real time information about the soil moisture and temperature
- Advanced Goal: Create a small database within the app that comes with preset ideal values for the most common potted plants grown in Florida
- Advanced Goal: Notify the user when a soil moisture or temperature critical low/high has been measured
- Stretch Goal: The app will consider what stage of growth a plant is currently in (seedling, budding, flowering), and modify the amount of water delivered for even more optimized growth, depending on the plant species

Project Objectives

❖ Water System Objectives

- To deliver water to the plants via a small water pump and electronically controlled solenoid valves
- To respond to inputs from the MCU to know when to open the solenoid valves, and for how long to do so.

❖ LED Light Fixture Objectives

- To use a surface mount light emitting diode to provide light to the plants
- To maintain relatively low power consumption to add minimal load and cost to the user
- To change the color of the programmable LED light to output the ideal growth frequency

❖ App/UI Objectives

- To create a user friendly interface that allows the user to create plant “profiles”; it will store the user input values for ideal soil moisture level and optimal spectral frequency, and communicate with the MCU to control these values
- To communicate to the MCU via Bluetooth
- To report the listed data values, like current moisture and temperature values, to the user

- To notify the user if any values of the critical conditions fall below or above a certain threshold

❖ **Capacitive Moisture/Temperature Sensor Objectives**

- To accurately measure the moisture and temperature of the soil in which the plant is growing
- To consume very little power in order to remain functional for long periods without recharging, ideally half a year or more
- To communicate via Bluetooth to the MCU the current values

❖ **MCU Objectives**

- To facilitate communication between the various peripheral devices
- To receive real time information from the sensors via Bluetooth
- To open/close the solenoid valves in response to the real time soil moisture level
- To duty cycle control the LED lights in order to provide the ideal power spectral density

2.4 Features and Functionalities

Overall growth conditions vary tremendously for various phyla of plants throughout the world, and therefore our design will be geared towards plants that are able to grow in the outdoors environment found in the earth's tropic and subtropic zones (between 0 and +- 35 degrees of latitude), and which expect to see fresh water, as opposed to brackish or salt water. However, even in this reduced category of greenery, growth can occur under a large range of conditions; however, optimal conditions for different types of plants have been well established in the scientific literature. Our goal is to optimize the plant growth by providing the ideal amount of water, as well as providing exactly the ideal frequencies of light, for the ideal amount of time. Of course, these values will generally vary slightly, or sometimes largely, based on which specific type of plant that is currently being put on the shelf. Therefore, our design will also implement an app that will allow the user to input which type of plant is in which of the three shelf locations. From the app, the user will be able to input the ideal amount of water the plant should receive over the course of a single day, as well as the ideal frequency of light at which photosynthesis is most efficient for that given species.

The functionalities can be listed as follows:

❖ **Overall System Functionality**

- The system will only require the maintenance of refilling the water tank when the tank starts to run low
- An accompanying app will track which type of plant is currently on which of the 3 shelf positions
- After the ideal water and light data has been entered into the app, the system will automatically deliver the correct amount of water and light to that location

❖ **App Functionality**

- The app will allow the user to select one of three plant positions, being top, middle, and bottom, and then specify which type of plant is in that position
- The app will have preset values for some of the most common plants grown in Florida, allowing a large percentage of users to simply use preset values
- The app will allow for users to create their own plant preset, which will allow for ideal water data, as well as the ideal light data, to be added
- The app will also have the capability of sending alerts to the user if any of the three critical conditions reach critical highs or lows; this would be for soil temperature, soil moisture, and light density per square meter
- The app will provide a small step by step checklist for when the user switches plants on the shelf; this will include a reminder to move the moisture/temperature sensor, as well as a reminder to seat the water line correctly

❖ **Soil Moisture/Temperature Sensor Functionality**

- The system will come with three “plug and play” sensors, which simply need to be inserted in the plant pots on the shelf
- The sensors will detect the soil moisture/temperature at a location near the roots, and will send the MCU a value every 10 minutes

- For seamless connectivity, the sensor will send its data via Bluetooth to the MCU

❖ **Water System Functionality**

- The system will come with a 12 Volt self regulating water pump which will keep the pvc lines at a constant pressure
- A built in, inline pvc water regulator will ensure that water pressure stays low enough to meet safety standards
- Electronic PVC solenoid valves will regulate the flow of water into each pot; these will be entirely independent of each other
- When the MCU determines that the soil moisture is ideal for a given plant, the solenoid valve for that plants water line will close immediately

❖ **LED Lighting Functionality**

- The system will come with high power, 120V surface mount, and programmable LED fixtures
- These fixtures will be duty cycle controlled in order to moderate the spectral density incident onto the plant
- These fixtures will also be programmable, and will emit a range of light that is ideal for a given plant
- The lights will also come with a dimming feature, which allows the light to operate at a more comfortable intensity for the human eye; in this mode, the lights will stay on for longer in order to still meet the ideal needs for the plants

❖ **MCU Functionality**

- The MCU will both process data from other systems, as well as act as the input to other systems within the total framework of the design
- The MCU will read data values for soil moisture, soil temperature, and light per square meter incident onto the plant
- The MCU will receive data from the app which informs it of which plant is in which position on the shelf

- The MCU will receive data from the app which informs it of the ideal amount of soil moisture for the plant; the MCU will compare the value read from the sensor and the ideal value, and
- The MCU will receive data from the app about the ideal lighting conditions for the given plant type in one of the three location; based on this information, it will have the LED operate at a specific operating frequency, as well as duty cycle control the light as to give the plant both the ideal frequency and amount of light energy
- The MCU will provide the app with data from the sensors so that the user can be sent critical information about their plants, in the event a critical condition value is outside of the acceptable range

2.5 Specifications

Table 1 below shows the design specifications and engineering requirements for this project. In this table, 3 specifications will be highlighted to be the values tested during the final review in SD2. These highlighted requirements serve as the foundation for ensuring the system operates reliably and efficiently. Each specification is carefully selected to address key functional aspects such as power consumption, response time, durability, and ease of integration with various plant environments. By focusing on these critical parameters, the design can be systematically evaluated to determine whether it meets the intended goals and provides a practical solution for users. Furthermore, these specifications will guide testing and validation procedures throughout development. By establishing clear benchmarks, the system's performance can be measured against expected outcomes, allowing for the identification of necessary modifications early in the design process. This structured approach not only ensures the project's success but also enhances its scalability for future improvements or variations in implementation.

Table 1: Engineering Specifications

Specifications Table	
Parameter	Requirement
App Data Refresh Rate	At least once every hour
Web Server Refresh Rate	At least once every minute
Soil Temperature Polling Rate	At least once every hour
Soil Moisture Polling Rate	At least once per minute
Light Sensor Polling Rate	At least once per minute
Number of Environments Maintained Simultaneously	At least 3
Light Sensor Relative Accuracy	No more than 3 percent difference between values under identical lux
Temp Sensor Accuracy	Within 2 percent between the Ambient Temperature range of 0 degrees Celcius and 35 degrees Celcius
Power Consumption (12V LED lights)	Less than 200 Watts
Electronic Solenoid Valve Response Time	Less than 5 seconds
Water Pump Flow Rate	At least 1 GPM (Gallon per Minute)
Water Pump Pressure	At least 40 PSI
Water Pump Power Consumption	Less than 75 watts
Maximum Shelf Size	4'x 6'x3'
15A Breaker Trip Time (12V)	Less than 1 second

2.6 House of Quality

Symbols	Descriptions								
+	Positive Polarity								
-	Negative Polarity								
⇓	Strong Negative Correlation								
⇑	Strong Positive Correlation								
↓	Negative Correlation								
↑	Positive Correlation								

Engineering Requirements									
		Temperature Measurement Precision	Water Pump Rate	Size	Battery Capacity	Moisture Level Polling ate	Cost	Data Transfer Rate	Solenoid Valve Response Time
Marketing Requirements		+/-	+	+	-	+	+	-	+
Upgradability	+	+	↑		↓	⇑	↑	↓	↑
Battery Life	+				↑			↓	↓
Maintainability	+			↓			⇓		
Portability	+			⇓	⇓		↓		
Water Capacity	+			↓			↑		
Affordability	+		⇑		⇓		↓		↓
Reliability	+		↑	↑	⇑		↑	⇑	⇑
Target for Engineering Requirements		95% Precision	At least 1 GPM	4"6"2"	1000 mWh	Once per minute	\$500 - \$1000	480 megabits per second	< 5 Seconds

Figure 1: House of Quality

2.7 Block Diagrams

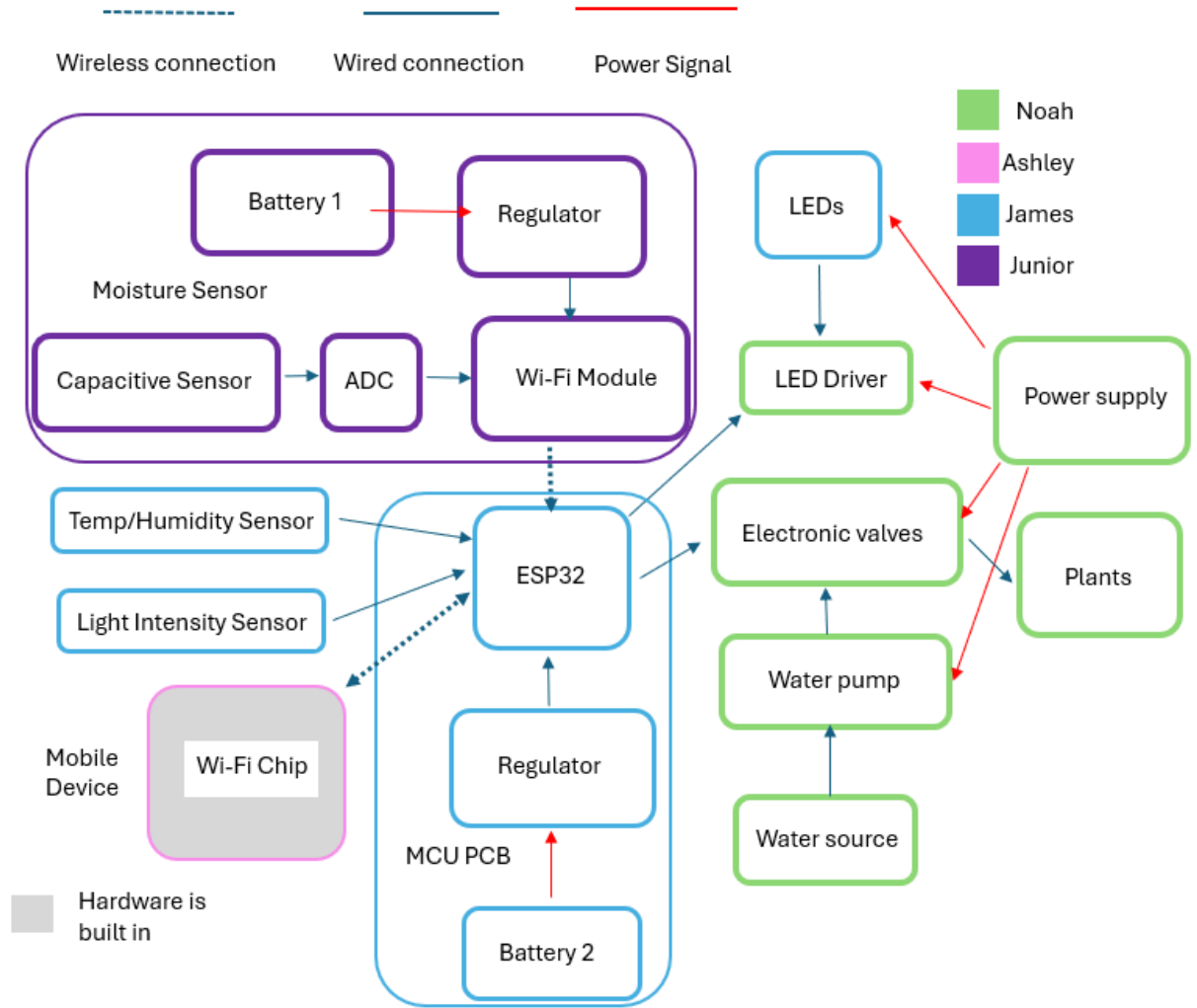


Figure 2: Hardware Block Diagram

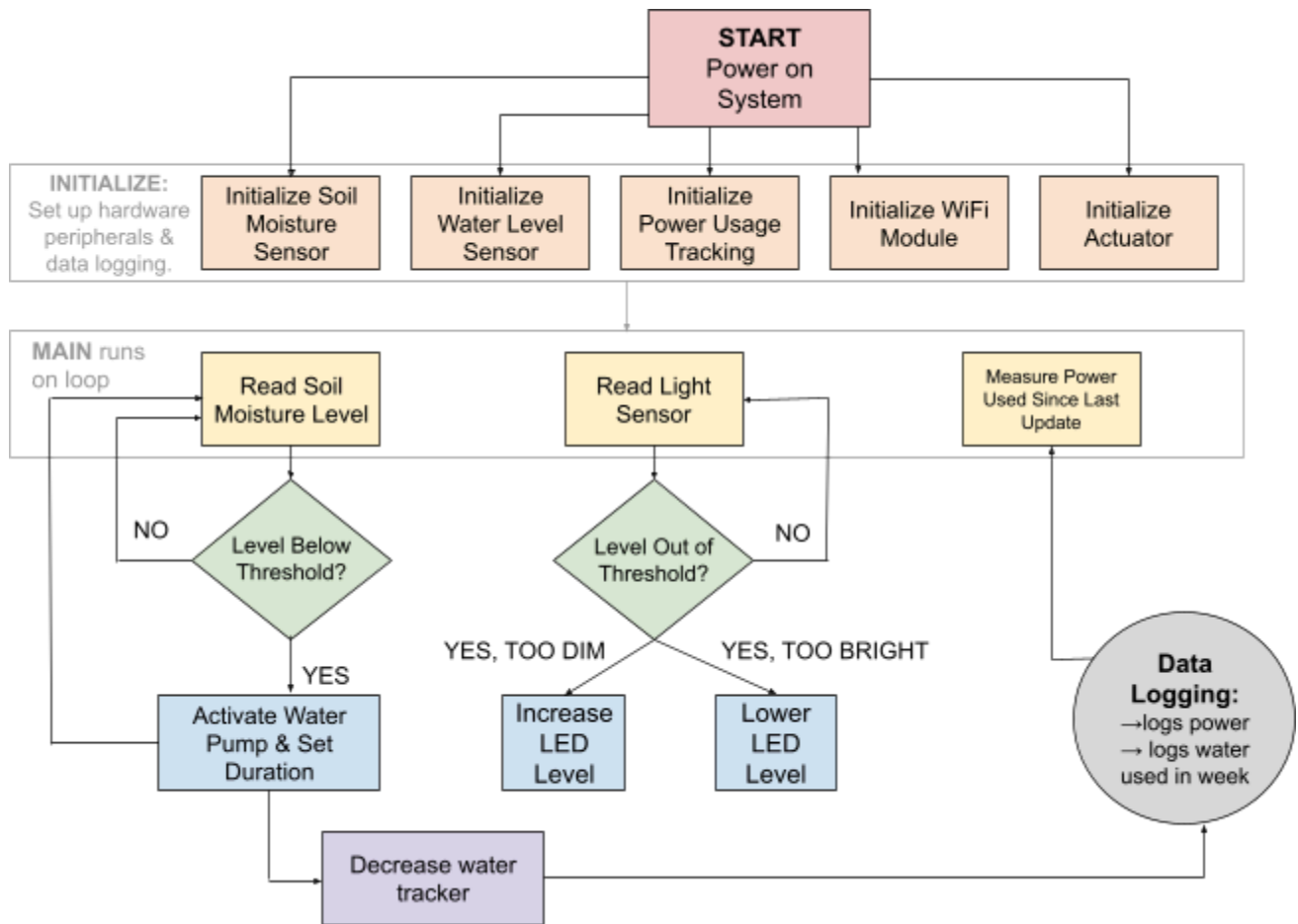


Figure 3: Overall Embedded Functionality Software Flowchart

3.0 Research

We will be discussing and comparing the various technologies and parts that will be implemented for our project. For each item we will be going over why we chose certain technologies and items over others.

3.1 Technology Comparison

In this section we will go over specific technologies, what we have learned about each of them, and why we think certain technologies would be best for our project.

3.1.1 Moisture Sensing Technology

Water is necessary for every plant to survive, and if you have too much or too little water for the plant, it can cause the plant to die. We plan to monitor the water intake of the plants by measuring the moisture of the soil that the plants are in. Some ways that we can measure the moisture of the soil are by using a volume water content (VWC) soil moisture sensor, soil water tension sensor, and single-point sensor. By incorporating these sensors, the system can ensure that plants receive the optimal amount of water needed for healthy growth. This data-driven approach allows for real-time adjustments to the irrigation process, preventing both underwatering and overwatering, which are common issues in plant care. Additionally, by automating moisture monitoring, users can maintain plant health with minimal effort, making the system more convenient and efficient.

3.1.1.1 Volume Water Content Sensor

The VWC sensor measures the percentage of water within a given volume of the soil, this is expressed by the ratio of total water content to total soil content. Electromagnetic VWC sensors operate by measuring the dielectric constant of the soil. As the amount of water changes in the soil so does this dielectric constant or permittivity of the soil. Water has a much higher permittivity than soil and as the soil becomes wetter the permittivity of the soil increases. VWC sensors have capacitors that charge and discharge that generates an electrical signal through the soil. This method can allow for fast and precise response from the sensor, making VWC sensors an effective option to monitor moisture levels in real time.

Accuracy of these sensors is a critical feature, without correct readings we would not be able to supply the correct amount of water that is necessary. By overwatering the plant you can suffocate it and underwatering the plant can cause stress, poor growth, and even death. Among the different types of moisture sensors the most accurate sensors are the volumetric and water tension sensors because of their advanced design.

Due to these sensors being a high-quality product, which has continued to be developed better over the years as well. The volumetric water content sensors have an accuracy range of $\pm 1\%$ to $\pm 5\%$. The accuracy of these sensors can vary depending on the type of soil, soil texture, and compactness of the soil. For the best accuracy of a VWC it is important to know the type of soil that you are using and temperature changes that could affect the dielectric property of the soil. To achieve the best accuracy users should calibrate the sensor for the type

of soil that they are using and take into account any other factors that could affect readings.

3.1.1.2 Soil Water Tension Sensor

A soil water tension sensor measures how difficult it is for a plant to get water from the soil. Soil tension is measured in centibars (cb) or kilopascals (kPa), where 1 kPa is equal to 1 cb. And a soil tension reading of 0 cb means then the soil that the plant is in is fully saturated with water. As the soil dries up then the reading from the water tension sensor also increases, meaning that the plant needs to do more work to absorb the water.

Different types of soil can have different readings for how much water is in the soil. This can occur because of the varying soil compositions, texture, and structure of the soil, which can all influence how the water is being retained within the soil. For example, a sandy soil that does not absorb water well will have a lower reading since the plant does not have to put as much work in to extract the water. Whereas if the plant was in a clay like soil it would need to do more work to absorb the same amount of water. Because of this it is important to calibrate the sensor and log all the data of the soil that you are testing to better understand the sensor readings. This can help the user get a better understanding of the water level that is necessary for the plant.

To better improve the accuracy of a soil water tension sensor you should know the soil composition because this will affect the soil texture as well as how compact the soil is. The placement of soil tension sensors is another important factor because you will need to be consistent with where you place it relative to the plant. Consistency ensures that the readings from the sensor are meaningful and reliable.

Placing the sensor closer to the root of the plant will have different readings than placing it further away from the plant or near the surface. This is because the water pressure near the roots may be different since the roots are able to absorb the water, rather than the water that is on the surface. Because of this it is important to have the tension sensor closer to the root of the plant so the user is able to better understand how much water the plant is absorbing.

3.1.1.3 Single-Point Moisture Sensor

Single-point moisture sensor takes only the reading at a certain location at the depth that it is placed at. Unlike the previous sensors mentioned above, a single-point moisture sensor is only able to take a reading at the specific point rather than the whole area around the sensor. These sensors have two

rods/electrodes that are placed into the soil and function by sending an electrical signal between the two rods. The sensor then measures the resistivity of the electrical signal through the soil.

This type of sensor has poor accuracy which is largely due to the fact that it is a low quality product, which is a reason for only being able to take localized measurements. Because they are only able to take localized measurements they fail to account for any variations in soil moisture of the surrounding area. That means that if there are any dry patches or wet patches that it would go unnoticed. These sensors are best used when placed near the root of the plant where the moisture of the soil is most impactful to the plant.

A major drawback of these sensors is that the electrodes that are in the sensor are able to easily corrode over time which can lead to poor measurements or complete failure. This corrosion can be attributed to the product being cheap and using low quality metals. Due to the layer of rust that forms on top of the rods, the correct amount of electrons may not be delivered or received from one rod to the other. Over time the corrosion on the rods will get worse and worse to the point where the sensor will eventually be completely inaccurate or break.

3.1.1.4 Capacitive Sensing

Capacitance sensors work similarly to a volume water content sensor. This is because the sensor also depends on the dielectric constant of the soil. The PCB traces are used as capacitor plates and the soil is used as the dielectric medium for the electrons to cross. As the capacitance of the circuit changes this corresponds to the moisture content of the soil changing as well. These capacitor plates are used as electrodes to measure the capacitance of the soil. This is illustrated in further detail with the image below from METER Group.

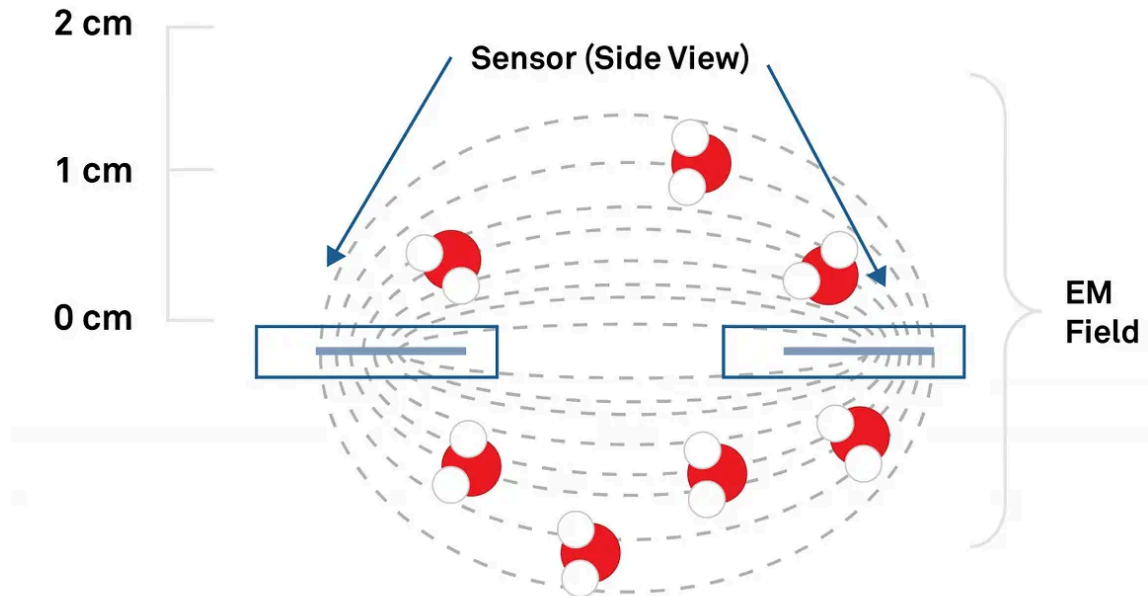


Figure 4: EMF between two capacitor plates [2]

To measure changes of the moisture using a PCB layout, traces are made onto the PCB that cover a certain area, these traces will act as capacitor plates. These traces in the circuit are paired with a timing element that takes these measurements over specific intervals. The signal that is received from one capacitor plate to the other tells the sensor how much moisture that there is in the soil. Using an ADC the data from the sensor will be sent to the WiFi module attached onto the sensor PCB, which this WiFi signal will then be sent to the ESP32. By communicating this data to the ESP32 it will allow for remote monitoring. From this data, the ESP32 is able to control the water content of each of the plants based on the threshold value for the certain plant.

Accuracy for a PCB capacitance sensor typically varies based on the design and calibration of the PCB sensor. Calibrating the sensor will require the user to place the PCB moisture sensor into water, this reading will be where the sensor is saturated. To improve the accuracy you can increase how large the traces that are being placed into the soil. The larger area allows for the capacitor plate traces to be discharged and recharged again. However, if the traces are too large, where the space in between the two traces is minimal, this can cause inaccurate readings because of water that may be connecting the two traces together.

3.1.1.5 Moisture Sensing Technology Selection

The Volumetric Water Content sensor is a good option for this project. This sensor has continuously been developed and improved over time and is a reliable choice. These sensors record their data over a wide area in the soil, helping prevent any areas from going undetected. They are also a good choice for preventing corrosion since most models use stainless steel probes. However, the issue with this sensor is that it is very expensive and we would need at least three of each sensor.

A Soil Water Tension sensor is also another good option for this project. Its reliability and corrosion prevention would allow for great use in this project however these sensors are also very expensive as well.

The single-point moisture sensor measures the resistivity of the moisture in the soil. These sensors are made cheaply out of poor metals and their sensing capabilities cover a small area. Being made out of poor metals allows for corrosion to occur more easily on the sensor causing them to degrade faster. This short lifespan makes single-point moisture sensors an impractical choice for gardeners looking for consistent and long-lasting soil moisture monitoring.

From researching these various types of sensors, it gives us an idea of how we want to approach our design. Going into this project we knew that we wanted to use a PCB moisture sensor, however we did not know much about how this technology works. We wanted to be able to design our own PCB where we will be creating our own capacitance moisture sensor. This capacitive sensor design is based on the idea of the volumetric water content sensor, this specific sensor can have better accuracy with how the PCB traces on the sensor are made and can also be improved with proper calibration.

Table 2: Types of Moisture Sensors

	VWC	Water Tension	Single-Point	Capacitive Sensing
Soil moisture calculation	Uses dielectric permittivity of the soil	Uses tensiometer to measure soil suction	At a specific point takes the conductivity of soil	Uses traces as capacitor plates to measure soil moisture
Sensitivity	Affected by soil type and	Soil texture	Contact to soil and soil	Soil make up and salinity

	salinity			
Output Value	Percentage	kPa	Binary or Resistance Value	Voltage or Digital Signal
Measurement Depth	Multiple depths	Near roots	Fixed at a single point	Few cm depth
Accuracy	High	High	Moderate	Moderate but can improve with calibration
Cost	High	High	Low	Low

3.1.2 Lighting

Photosynthesis is an essential process for every plant, because it allows plants to convert light energy into the nutrients that they need in order to survive. An important part of photosynthesis is lighting, this affects a plant's ability to produce chlorophyll, which is the green pigment responsible for absorbing light and driving energy production. Without enough light exposure, a plant is not able to produce enough chlorophyll which can lead to growth issues, discoloration, and potentially death. A consideration to achieve ideal lighting conditions is the type of bulb that we will use. Different light sources, such as LEDs, fluorescent, incandescent, and infrared bulbs emit varying intensities and wavelengths of light which affect plant growth differently.

The process of photosynthesis itself occurs in two main stages: the light-dependent reactions and the light-independent reactions (also known as the Calvin cycle). In the light-dependent reactions, chlorophyll absorbs photons of light, exciting electrons that are then transferred through a chain of proteins in the plant's chloroplasts, ultimately producing ATP and NADPH. These molecules are crucial for the second stage of photosynthesis. If the plant does not receive enough light, the light-dependent reactions cannot produce sufficient energy for the Calvin cycle, which in turn limits the plant's ability to synthesize glucose from carbon dioxide. This glucose is the energy source that drives cellular functions and growth. Hence, ensuring that a plant receives the appropriate type and intensity of light is crucial for maximizing photosynthetic efficiency and supporting healthy growth.

The wavelength of light also plays a key role in photosynthesis. Plants primarily use red and blue light for photosynthesis, as these wavelengths are most effectively absorbed by chlorophyll. Red light (around 660 nm) promotes flowering and fruit production, while blue light (around 450 nm) is important for vegetative growth, helping to control leaf and stem development. By understanding the biological needs of plants in relation to light wavelengths, we can optimize light sources to better match the needs of the plants we are growing, ensuring they receive the most beneficial light for their specific growth stages. This consideration is especially important for indoor gardening, where natural sunlight is limited or non-existent.

3.1.2.1 LEDs

An LED emits both light and heat that is based on a PN junction. Electrons have to jump the energy gap of the junction, when this occurs the required voltage is met and both light and heat are emitted from the LED. LEDs allow us to easily change how much light each plant is getting and gives us the option to customize the lighting level for each plant. By doing this it would allow for use of different types of plants, such as ones that may need more or less lighting. This would be done by changing the duty cycle of the array of LEDs. Customizability allows for the user to be able to change the lighting spectrum of light. Using more lighting colors can promote growth at various stages of the plant's life.

LEDs are more energy efficient than fluorescent tubes by 18%-44%, as observed by a study from the University of Michigan. LEDs convert almost all of the energy supplied to them as light and not as heat, unlike fluorescent and incandescent bulbs. LEDs are more expensive than other bulbs but this initial cost can be made up for by the longevity and energy efficiency of LEDs.

3.1.2.2 Fluorescent

Fluorescent lighting is another good option since it doesn't give off as much heat as an incandescent bulb, as well as being a decent source of light emission. Fluorescent bulbs give off energy by having a collision of free electrons and mercury vapor that is inside the tube of the bulb. This collision then causes an excitation between electrons causing high energy levels. These electrons will then fall back down to emit two different UV wavelengths, 245 nm and 185 nm.

Since UV light radiation is not on the visible light spectrum, a coating has to be applied to the inside of the bulb to emit the desired wavelength on the light spectrum. The coating allows the manufacturer to select different colors for users to buy, this can range from a bright white to a warm orange. The downside

of fluorescent bulbs is that this is the only form of lighting control for the bulb. To change the color of the bulb you would need to change the bulb itself.

Not all of the UV light is converted into visible light. A portion of the electrical power that is supplied to the fluorescent bulb is converted . Moreover, the more energy that a bulb gives the more heat it has to emit since not all of the energy is converted into light, making fluorescent bulbs less energy efficient than LEDs.

3.1.2.3 Incandescent

Incandescent lighting uses most of the electrical power that is supplied to it as heat rather than light, this is not an efficient choice for our gardening system. Incandescent lights function by sending an electrical current through a wire within the bulb. The more current that is passed through the wire the hotter the wire will get, the hotter the wire gets the brighter the wire will glow. Because of this most of the electricity that is produced is in the form of heat rather than light. Around 10% of the electrical power that is consumed by incandescent bulbs is in the form of visible light [14].

Another downside is that there is hardly any form of lighting control for incandescent bulbs. However, these lighting controls can be found in LEDs where the user is able to increase or decrease the light intensity. As well as being able to change the color of the light, this would allow for optimal growth at various stages of growth for the plant.

3.1.2.4 Infrared

Infrared lighting is only used because of the radiation that would be given off by the bulb, which is only in the form of heat energy. The heat that is given off from infrared can allow for more tropical plants to be used and also growth of plants. Infrared lighting becomes useful when trying to stimulate any growth with plants such as sprouting and flower blooming. Some arrays of LEDs have a wide enough light spectrum where it is able to emit infrared waves. This is a viable option for using infrared since it is already built into the array of LEDs and could help reduce cost if infrared lighting will be used in this project.

Infrared does a good job at keeping plants warm and stimulating growth. However, just like humans, too much infrared lighting on the body is not healthy and has to be closely monitored. Too much heat will cause the plants to be overwhelmed and stressed and in turn the plant will start wilting away. Our gardening system is designed to be indoors in a well regulated tempered climate. From previous products and projects of hydroponic gardens the nominal environmental temperature is between 59-86°F. Because of this, having

infrared lighting may not be necessary, as well as being more of a risk than a benefit because of overexposure.

3.1.2.5 Lighting Selection

Given the importance of lighting, our project focuses on creating the most optimal lighting environment for various types of garden plants to ensure healthy growth and development.

Incandescent lighting is the worst option for our project. From the limited efficiency, high heat output, and lack of control, incandescent lighting is an unsuitable option for optimizing our indoor garden system. Most of the energy that is produced is in the form of heat and would require more power to reach the desired light intensity. Another downside of incandescent lighting is that too much heat can be harmful to the plant.

Infrared lighting is a unique choice as it can promote plant growth but does not emit enough visible light to be the only light source. There are options to include infrared lighting in LED lights, however this will make the overall LEDs more expensive. Another option is to buy a stand alone infrared lighting source but these can also prove to be costly as well. An infrared light source can be hard to manage as well because having too much infrared can be harmful. While infrared lighting can help promote growth, there are too many risks and too costly to consider infrared as an option for our garden.

Fluorescent is a good option as it will consume less power to produce the same amount of power to produce the same amount of light as an incandescent bulb. However, fluorescent bulbs do not convert as much electrical power to light energy as an LED does.

Certain LEDs are able to remotely change the color and light intensity that is emitted. We will be using such LEDs so that the user is able to change the intensity and color of the light as desired. LEDs are also the most efficient power consumption option. While there are certain LEDs that can have infrared lighting built into them, we will not be selecting this option as previously discussed.

For these reasons that have been discussed we will be using an LED light with the user being able to change the light intensity and as well as the possibility of changing the color that the LED emits.

Table 3: Lighting Comparison

	LEDs	Fluorescent	Incandescent	Infrared
--	------	-------------	--------------	----------

Energy Consumption	Low	Moderate	High	Moderate
Lifespan	50,000+ Hours	~15,000 Hours	~1,500 Hours	Varies
Cost	High	Moderate	Low	Varies
Heat Emission	Low	Moderate	High	Only Heat
Lighting control	Full color spectrum, dimmable	Emits cool white color	Emits warm red/yellow	N/A

3.1.3 LED Drivers

This project relies on both the implementation of LED lights to provide light to the plants, but also the ability to dim or entirely shut off these sources based on inputs from light sensors in each of the plant environments. Furthermore, LED lights often require precise voltage and current control. For dimming applications, a PWM signal is often used to regulate the effective voltage at the LED, and yet again, this is also done with drivers. Drivers also allow the programming of light in LEDs that support multiple colors, and can even be used to mix these lights in applications where individual addressable LEDs are used. For these reasons, a driver is absolutely mandatory for this project. However, various technologies are available, with different purposes, and as per usual, different pros and cons.

Courtesy of Shelly Group Corporate

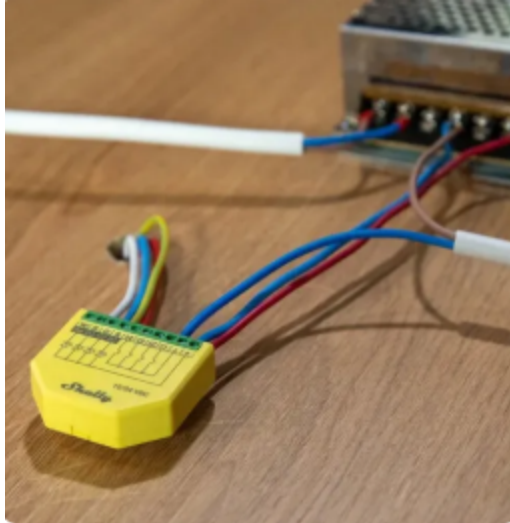


Figure 6: The Shelly RGBW2

3.1.3.1 Dimming Technologies

In LED drivers, there are generally two types of dimming methods, being PWM and analog dimming. In PWM (Pulse Width Modulation) dimming, the LED is cycled on and off rapidly, and the period over which this happens in a given cycle will determine its duty cycle. This will have the effect of lowering the apparent brightness, as well as the light intensity, because the LED is only on for a fraction of the time that it would be on if not dimmed. Analog technology, on the other hand, simply varies the current through the LED in order to get the dimming effect. The analog method requires much simpler circuitry, basically only a variable resistor, but is flawed in a few important ways. One such way is that LED's are often very sensitive to the input voltage and current, and can burn out or turn off completely if a certain ceiling or floor threshold are reached. Additionally, this type of dimming can cause uneven dimming in LED light strips, where multiple lights are in series with each other. Lastly, LED's actually emit a different frequency spectrum when provided with a different current than the standard operating current. This can be very noticeable or not at all obvious based on the specific LED, but this is a problem that simply is not faced by PWM dimming. PWM, on the other hand, is generally more efficient, and avoids some of the problems mentioned above, but almost always requires special hardware to actualise.

These two dimming technologies can be compared using a table of specifications below:

Table 4: Dimming technology

Dimming Technology	PWM Dimming	Analog Dimming
Efficiency	Excellent	Poor
Circuit Simplicity	Poor	Excellent
Lighting Effectiveness	Excellent	Moderate
Technology Compatibility	Excellent	Poor

3.1.3.2 LED Driver Comparison

When it comes to deciding which is the correct LED Driver for this project, a few things must be taken into consider. As discussed in the technology comparison section, there are some clear benefits to going with PWM Dimming, especially given that an LED Driver will prove an immensely useful device for turning off the LED strips individually. Furthermore, it would be excellent if this could be done remotely, as is the theme with the project, and the only way to accomplish such a feat is to use a wireless LED Driver. Therefore, our design constraints mandate the usage of a wireless LED Driver that supports PWM Dimming. There are a few products that match this requirement

3.1.3.3 Philips Hue Lightstrip Plus LED Driver

This particular driver is integrated with the light strip that it is controlling. Generally, it is considered to have an excellent frequency output, with specific color selection that few other models can match. However, using this device requires using its specific app, meaning that this app would need to be integrated into the app being developed for this project for a seamless experience. Specifically, the lighting sensor inputs, as well as the user inputs into the app for the type of plant, would need to somehow be transferred to the proprietary Philip Hue app, or vice versa, which would make implementing this particular device very complicated.

3.1.3.4 WLED for ESP32

Just based on the name, and given we are using the ESP32 as the MCU for this project, this light driver seems like an obvious fit. The code is open source, the lights it controls are highly programmable, and generally this meets the requirements of our design. However, it is not quite that simple, as there are some problems here. One Problem is that for the WLED to be controlled

remotely, another PCB must be designed; this one would also need a dedicated power source on the PCB. Furthermore, the WLED can struggle when tasked with controlling multiple LED strips simultaneously.

3.1.3.5 Shelly RGBW2

The Shelly RGBW2, sometimes sold as the RGBW PM in the United States, has some unique benefits that relate specifically to the project. One particularly useful ability of the shelly RGBW2 is the ability to be controlled both by the shelly app, as well as APIs and even MTTQ. This proves incredibly useful for testing purposes, as the app can be used to test the product initially, while the ESP32 can ultimately control the device. It is therefore a given that the Shelly has wifi capabilities, and also possesses bluetooth capability. Furthermore, the Shelly can drive up to 4 simultaneous strips of lights, and can dim them using PWM, or change their color if the LED light strip is programmable. The shelly is also versatile in it's powering option, as it can both be supplied by a 12V DC or 24V DC source, and can also drive either 12V DC or 24V DC light strips. Additionally, it supports 288W of output power, which is more than enough to meet the design requirement for this particular project. Ultimately, this project utilized the RestAPI of the Shelly, which allows it to receive inputs via HTTP Post requests.

3.1.3.6 LED Driver Conclusion Part Comparison Conclusion

Given the unique specifications that this project mandates, the selection was narrowed down to only these drivers. In the table below, the upsides and downsides of each driver will be derived based on their performance on a number of relevant specifications

Table 5: LED driver selection

LED Driver	Philips Hue Lightstrip Plus	WLED for ESP32	Shelly RGBW2
Accessibility	Terrible	Perfect	Excellent
PWM Dimming	Excellent	Excellent	Excellent
Supports Multiple LED Strips Simultaneously	No	No	Yes
Efficiency	Excellent	Moderate	Excellent

Ease of Powering	Moderate	Poor	Perfect
Quality of Light	Perfect	Moderate	Moderate

Based on these specifications, and considering the overall requirements of the project, the LED drive that is the most compatible is the Shelly RGBW2, with the WLED for ESP32 in close second. Ultimately, it comes down to simultaneous control from the Shelly, although this can be easily remedied by using multiple WLED's in case the Shelly does not perform as expected.

3.1.4 Light Sensing Technology

In our project we will have a light sensor for each of our plants to allow for optimal growth for various plants. This sensor will send a reading to an app and the user can then adjust the intensity of the light as necessary. For choosing a light sensor we will be researching three different types of light sensors, photodiodes, photoresistors, and phototransistors.

3.1.4.1 Photodiode

A photodiode is a basic semiconductor with a P-N junction. When light comes in contact with this P-N junction, electron-hole pairs join together and allow for the flow of electrons to be formed, creating a current that passes through the P-N junction [8]. As the intensity of the light that is applied to this P-N junction increases, the more current that is being passed through the P-N junction. The sensor converts the amount of lux (the intensity of illumination) that is seen by the sensor and converts this into amps. Lux measures that amount of light that falls onto one surface.

$$\text{Lux (lx)} = \text{Lumen (lm)}/\text{m}^2 \text{ [9]}$$

Equation 1: Lux equation

To have a current output from this sensor, the diode will need to be oriented in reverse biased. In reverse bias more electrons in the N layer are pulled towards the positive terminal and more holes in the P layer are pulled to the negative terminal [10]. This is called photoconductive mode, this then causes the depletion region of the P-N junction to increase, from this effect the response time decreases. This photoconductive mode also creates a proportional relationship between the input light intensity and output current from the light sensor. However, because of this noise is created in the sensor called “dark current”. This dark current is leakage current that comes from the P-N junction.

3.1.4.2 Photoresistor

A photoresistor measures light intensity by changing the resistance of the sensor as light hits it. This light intensity input measurement is the same as the photodiode and measures the amount of lux that hits the surface of the sensor. As the light intensity increases the resistivity decreases and similarly, as the light intensity decreases the resistivity increases [8]. Photoresistors are also cheaper to make and sell than other types of light sensing technology.

To measure the light intensity a simple voltage divider can be created by connecting another resistor in series to the photoresistor. As the resistance of the photoresistor increases more voltage will be observed at Vout, and as the resistance decreases Vout will decrease as well.

One of the disadvantages of photoresistors is that for it to be accurate it has to have a slow response time [11]. This is not useful in an application where fast measurements may be necessary. However, this sensor can be used in a setting where light only needs to be detected and not measured.

Another disadvantage of photoresistors is that they are susceptible to an increase in temperature. As the temperature of the sensor increases the sensitivity of the sensor can change and become unreliable. While we are measuring temperature in our project, we are not controlling it. If the temperature changes and causes the sensor to become unreliable then we would need to replace this sensor.

3.1.4.3 Phototransistor

Like the photodiode, the phototransistor is another semiconductor device, these two types of sensors are very similar. In a phototransistor, the sensor receives a current from a photodiode and this current is then amplified by the built-in Beta value of the transistor [8]. A phototransistor able to work around the problem posed when using a photodiode, that the current output can be too small and difficult to read. This is further described in the equation below.

$$I_B = \frac{I_E}{(\beta+1)}$$

Equation 2: Relationship of Base current and Emitter Current

However, a problem occurs because of this amplified current. Like a photodiode, a phototransistor also has a dark current and because a phototransistor is able to amplify the photocurrent the dark current is also amplified [12]. The dark current being applied leads to a larger noise that is

being created. While having the photocurrent being amplified is nice to have, it can cause more noise to be detected in a low light intensity environment.

Another problem is that a phototransistor has a slower response time than a photodiode, this is due to the built in gain of the transistor [13]. The larger the gain there is from the transistor the longer the response time, and similarly the smaller the gain, the shorter the response time. However, reducing the gain of the transistor also decreases the sensitivity of the sensor as well.

Table 6: Comparison of different light sensing technologies

	Photodiodes	Photoresistors	Phototransistors
Response Time	Fast	Slow	Medium
Output Measurement	Amps	Ohms	Amps
Cost	Medium	Low	High
Response Type	Linear	Linear	Linear

3.1.4.4 Light Sensing Selection

After researching and comparing photoresistor, photodiode, and phototransistor light censoring sensing technology, the photoresistor is not an ideal option for our project. These types of sensors do not take accurate measurements and are cheaply made. As stated earlier, these sensors are best used to detect the presence of light, rather than for detecting how much light is currently on the sensor.

The phototransistor is a good option for our project and works in a similar way to photodiodes. However, these sensors not only amplify the signal but amplify the dark current as well. This unwanted dark current is seen as noise and may impact the readings. Another downside of the phototransistor is that they are more expensive than a photodiode.

For our project we will be selecting a photodiode. This is because photodiodes provide a signal with limited noise. However, we may choose to switch to a phototransistor if we are struggling with getting a strong enough signal from the photodiode.

3.1.4.5 Photodiode part selection

Photodiodes, also known as light intensity sensors, will allow the user to be able to monitor the light around each of the plants. These photodiodes can be placed in various places around the area of the plant so that there are multiple readings to observe for each plant.

Photodiodes are able to be purchased individually or on a development board. The benefit of buying these individually is that it will be cheaper for us to buy but would require us to design and solder a new sensor. Designing a new sensor would be a beneficial experience for us but could cause many errors with readings and would require more troubleshooting. Purchasing the development board will allow us to save a lot of time with testing a newly developed sensor. By doing this we would be able to test our code once we get the sensor and would not have to worry as much about any inaccurate readings from the sensor.

For our project we will be using various sensors and have decided that we will only be using I2C communication for all of our sensors. Because of this, we will only be discussing specific photodiodes that use I2C communication.

3.1.4.6 Adafruit BH1750 Ambient Light Sensor

This sensor has a 16-bit resolution with a lux reading of 0 to 65,000+ lux, and from the datasheet can be improved to 100,000 lux with the proper calibration. This sensor has a regulator built into the board already and is able to support both 3.3V and 5V inputs. The sensor uses the standard pin size of 2.54mm. However, the only information that is given about the overall size of the board is that it is “not much bigger than a grain of rice”. This sensor also has no specifics on what the operating temperature is for the sensor. While we will be monitoring the temperature of each plant we do not want the burden of worrying that it will not operate in certain temperatures. Another downside when looking at the datasheet is that there is no specifics on the power drawn for the sensor.

3.1.4.7 Adafruit VEML7700 Ambient Light Sensor

The VEML7700 has a lux range of 0 to 120,000 and also has a 16-bit resolution without the need for calibration, unlike the BH1750. It has an operating voltage of 2.5V to 3.6V. The dimension of the board cannot be found on the datasheet but the sensor itself is 6.8 mm x 2.35 mm x 3 mm. The spectrum of the sensor is between 250 nm and 950 nm. The power draw of this sensor cannot be found within the data sheet

3.1.4.8 TSL2591

The sensor needs a voltage supply of 2.7V to 3.6 V and is able to operate anywhere from -30°C to 70°C. From the spectral responsivity graph, Channel 0 and Channel 1 are able to measure from 450 nm to 1100 nm. In an active state the sensor draws typically 275 uA, with a max rating of 325 uA. In a sleep state this sensor draws about 2.3 uA. The lux range of the TSL2591 is not found in the datasheet which is crucial information for a light intensity sensor.

3.1.4.9 Photodiode Selection

For our sensor selection we will be going with Adafruit VEML7700 Ambient Light Sensor. While the BH1750 is cheaper, the range of the VEML7700 is higher and could prove to be useful in our project. We are not going with the TSL2591 because the lux operating range is not given to us, so we will not know the saturation region of this sensor. The TSL2591 is also the most expensive option. While the operating temperature is not provided for the VEML7700 and the BH1750, our garden system is designed to be used as an indoor unit where temperature should not be an issue.

Table 7: Photodiode part selection

	BH1750	VEML7700	TSL2591
Range (lx)	0-100k	0-120k	0-88k lux
Resolution	1 lux	0.0036 lux	0.000377 lux
Power	~0.36 mW	~0.51 mW	~0.6 mW
Current Draw	0.12 mA	0.17 mA	0.2 mA
Standby Current	0.01 µA	0.8 µA	2 µA
Cost	\$3.29	\$1.25	\$1.67

3.1.5 Soil Types

Before discussing water delivery to the plants, soil types must be discussed. Ideally, the soil type shouldn't be particularly relevant for such a system, but different soil types have different water retention abilities, and this in turn will affect the average draw of the water pump, as well as the average amount of cycles that the solenoid valves will see, which will matter when determining the appropriate AC to DC converter to select for the project. Therefore, we will consider three different types of soil, ranging from severe loss in soil moisture, to moderate loss in soil moisture, to very little loss of soil moisture. Of course,

from the level of the user interface, the user will be able to tell the app which particular type of soil is currently occupying the plant pot; while this will not change the watering schedule per se, as this is determined by the capacitive threshold of the sensor, this will allow app to communicate to the user how often the system is actually running. In terms of how water retention is actually measured, the volumetric water content it taken of the soil after some given volume of water is applied to it, and then some time threshold later, all soil types are measured, with the difference in soil moisture being the basis for saying that one soil type has high retention compared to another. However, this type of measurement itself is outside the scope of the project, and therefore the retention characteristics are taken from standard values assigned to each soil type; even the specific values are not so relevant as compared to the comparative measure between soil types.

3.1.5.1 Potting Mix

Potting Mix is actually not exactly a natural soil, but a conglomerate of different naturally occurring substances that are ground together. This includes vermiculite, limestone, fertilizers (typically potassium or phosphorus nitrates), and wetting agents. Potting mix is one of the most widely available and versatile soil options sold at hardware and garden supply stores. It is typically composed of peat moss, vermiculite or perlite, and composted organic materials. This type of soil is engineered to retain moisture efficiently while also allowing adequate aeration for plant roots, meaning that it has a high retention value, which in turn will mean the pump will run the least when a pot is filled with this soil type.

3.1.5.2 Garden Soil

Garden soil is a heavier, more compact soil that contains natural mineral content such as clay, sand, and silt, often mixed with organic compost. It is generally denser than potting mix and is intended for in-ground use or raised garden beds, although it certainly can be used in indoor applications, such as within a plant pot. Although it is not quite as ideal for retaining moisture as Potting Mix, it will still rank in the moderate retention category, as the clay in particular serves to retain the moisture well. Therefore, the pump and valves will run a bit more in this case than in the case where a pot is filled with potting mix.

3.1.5.3 Succulent Soil

Cactus soil, commonly marketed for use with succulents and desert plants, consists mainly of sand, pumice, and small quantities of organic material. It is specifically designed to drain quickly and avoid water retention, which is ideal for preventing root rot in drought-tolerant plant species. The sand and pumice are rather porous, which causes rapid drainage, and therefore water must be applied to the plant rather frequently to meet the moisture threshold required. Of

course, this type of soil will therefore cause the pump to run the most often. As is standard engineering practice, the worst case scenario in terms of power draw will be assumed, and this will affect the sizing of the AC to DC converting unit chosen.

An additional idea for a future improvement would be to allow the user to input the soil type within the app, and to allow this to influence water delivery. For this project, the user will enter in the plant type, and based on ideal moisture values the pump will deliver some amount, but in some cases using the soil type could also be useful. The following soil types are summarized in the table below, although no particular selection is made, as the user will determine the soil used. This table serves only to function as a way to summarize the water retention characteristics of each soil type:

Table 8: Soil Type Comparison

Soil Types	Succulent Soil	Garden Soil	Potting Mix
Water Tension Characteristics	Drains very quickly	Drains at a moderate speed	Retains water very well

3.1.6 Water Delivery to Plants: Pumps

A critical aspect of this design is the delivery of water to the individual plants. For this particular application, given that a control feedback loop is required, the only viable option is to use a powered pump to draw from a water storage tank and pump it to the plants. As described above, this control will be achieved by actuating electronic solenoid valves that will control the amount of water flow (GPM) to each plant individually. Therefore, the optimal tank, pump, power supply, and solenoid valves must be analysed with these particular constraints in mind.

3.1.6.1 Types of Water Pump Technology

In general, the type of pump needed for this design allows for a wide variety of pump types. The most commonly seen pump types in other irrigation projects are Diaphragm pumps, centrifugal pumps, and Peristaltic pumps; each of these types has certain benefits and downsides associated with them. Because the solenoid valves will actually control the amount of water flow to the plants, the pump need not be particularly precise in its water flow, nor does it need continuous flow. Additionally, due to the short piping distances in this project, significant pressure is not needed from the pump either.

3.1.6.2 Diaphragm Pumps

Diaphragm pumps are the most common, and work via a moving diaphragm that pulls a suction on the fluid, creating pressure. A check valve is needed in such designs to prevent backflow; this is always implemented in the main housing of the pump, and is never external to it. These types of pumps are very common in demand pump applications, meaning that the pump keeps a water system at some threshold pressure, and is only activated when the pressure drop occurs in the system. Their self priming nature also lends well to its usage in drawing from water reservoirs that sit at a lower relative height than the pump. A diagram is pictured below, courtesy of Tacmina, that shows a basic representation of the water flow, the diagram, and the internal check valves in a simple diaphragm pump:

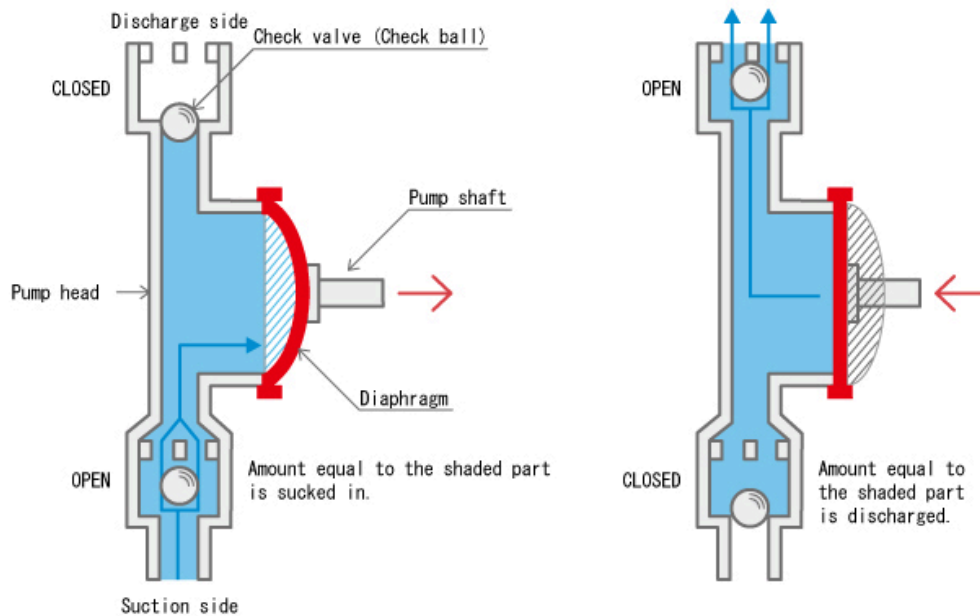


Figure 5: Diaphragm Pump Model

3.1.6.3 Peristaltic Pumps

Peristaltic pumps, on the other hand, use rollers to compress the tubing that the liquid is stored in, which creates a pressure differential on one side of a given roller. Because the rollers contact only the tubing, the fluid never directly contacts the pumping mechanism. These types of pumps are generally used in applications where the pump itself must control precise water flow to a certain destination. Additionally, in applications where the fluid being pumped must not

be contaminated to any environment outside of the designated tubing, these pumps would be a requirement. Furthermore, these pumps are generally very low pressure and work at very short distances.

3.1.6.4 Centrifugal Pumps

Centrifugal pumps use a spinning impeller to impart force on fluid within the pump, which pressurises it in a certain direction relative to the impeller. This method of pumping is commonly used in water pumps for boat motors, and in general, are found in applications where a large amount of water must be delivered. Furthermore, these pumps tend to be poor at regulating pressure, because if the reservoir from which the water is being pulled is suddenly at too low of a level, the impeller will continue spinning, meaning pressure drops off rapidly. Additionally, they are not self priming, and therefore requires bleeding air from the entire system before use

3.1.6.5 Continuous Flow Pumps

Continuous Flow pumps are common in applications where constant water flow is needed, such as fountains that constantly stream out water. Generally, they are quite simplistic in their design. Additionally, they do not require any kind of pressure sensing mechanism, as they will continue to pump when applied voltage. They are most commonly found as centrifugal pumps, rarely as diaphragm pumps, and never as Peristaltic pumps. This pump type has the downside of being less power efficient than demand pumps

3.1.6.6 Demand Pumps

Demand Pumps, on the other hand, use a pressure sensor within the pump housing to determine what the pressure is on the output nozzle of the pump. When this pressure is below a certain preset (and often adjustable) threshold, the pump will turn on, provided it has power. They are common in settings where constant flow is not needed, as it is often more difficult to switch the incoming power on and off than to simply develop a constant pressure across a water system. These types of pumps are most commonly seen as diaphragm pumps in this type of application, although any type of pump mechanism has a pressure sensing and control unit, meaning any type of pump can be found as a demand pump. Additionally, this pump type has the benefit of inherently being more power efficient than continuous flow pumps.

3.1.6.7 Water Pump Technology Comparison

Below is a table that summarizes the water pump technologies that were considered for the project, showing the upsides and downsides for both relative

to our project demands. Keep in mind we can choose between demand and continuous for each of the three pump types, meaning we will make two selections:

Table 9: Water Pump Technology Comparison

Technology	Demand	Continuous	Peristaltic	Diaphragm	Centrifugal
Flow Rate	Varies	Very High	Very Low	Moderate	Very High
Power Consumption	Minimal	Very High	Very Low	Moderate	Varies with specific pump

3.1.6.8 Water Pump Part Comparison

Below is a basic description of the design constraints, the types of pumps that meet the constraints, and then a comparison of these suitable water pumps to decide which pump will ultimately be used for the project.

3.1.6.9 Water Pump Constraints

As mentioned previously, the water pump for our system needs to be able to maintain a moderate pressure, somewhere between 50 and 100 PSI in order to maintain strong enough water pressure to effectively deliver water. An inline water pressure regulator will serve to make fine tune adjustments for the specific plant needs. It is clear that the ideal pump based on the types listed above would be a diaphragm demand pump. For this application, a continuous pump would not only be wasteful, but would be detrimental to the desire to limit water flow to a specific amount. Furthermore, the centrifugal and peristaltic pump types present too many other problems, such as too low of a flow rate in the case of the peristaltic, or too high of GPM in the centrifugal case. Therefore, three diaphragm, demand pumps are compared in the table below:

Table 10: Water Pump part comparison

Model Number	R3526144D	Delavan PowerFlo 7822FS-201-SBIP	Everflo EF2200
Cost	\$49.99	\$81.37	\$77.47
GPM	3.0	2.2	2.3
Maintained PSI	50	60	70
Supply voltage	12V DC	12V DC	12V DC

Given that all of these pumps meet the minimum product requirements needed for the design, the R3526144D has been selected as the optimal pump for this project, given its lower price compared to the other devices, and its higher flow rate.

3.1.7 Water Delivery to Plants: Solenoid Valves

In the water system design, the solenoid valves will control how much water flows to the plants. These valves will be controlled from an external input, from the microcontroller, based on moisture value readings from the moisture sensors. The solenoid valves will require fast actuation, as regardless of the polling rate of the sensors, if the valves do not open/close fast enough, the incorrect amount of water will be delivered. The power source of the valves must be also be paid close attention, as this will determine how exactly the MCU will control the water flow

3.1.6.1 Solenoid Valve Types

Electronic solenoid valves are essential components of this automated fluid control system. Fundamentally, they actuate by using an electromagnetic coil to open or close a valve, allowing or restricting fluid flow. This is done by providing a current through the coil, which will be controlled from the MCU, as discussed above. There are two primary types of solenoid valves used in modern irrigation project applications: direct-wired solenoid valves and bluetooth-controlled solenoid valves. Of course, each type has its benefits and downsides, and the constraints must be considered when ultimately choosing which type to go with.

3.1.7.2 Direct Wired Solenoid Valves

Direct-wired solenoid valves are the most common selection for fluid control applications in general. These valves are hardwired to a power source, which in this case, would be the MCU, which would directly control it via an output signal. When an electric current flows through the coil, it generates a magnetic field that shifts the valve's internal mechanism, either opening or closing the fluid passage; in most cases, these valves are normally closed and do not hold the open position unless a constant current is provided. One key advantage of the direct-wired solenoid valves is their reliability and instantaneous response. Since they are physically connected to a control circuit, there is minimal risk of communication interference or delays. Additionally, these valves can operate in environments where wireless signals may be weak or unreliable, such as industrial settings with significant electromagnetic interference. Furthermore, they are a bit easier to control simultaneously, as the output voltage can simply be provided for each valve. The one major downside here is the additional wiring complexity, as physical wires must run from the MCU to the valve.

3.1.7.3 Wireless Controlled Solenoid Valves

Wirelessly-controlled solenoid valves provide wireless operation, allowing them to be triggered remotely via a Bluetooth/Wifi-enabled microcontroller, such as the ESP32. These valves contain a built-in Bluetooth/Wifi receiver that processes the control signal and activates the solenoid. Being a wireless connection, they have the obvious benefit of limiting wiring complexities, and additionally, allow their placement far from the MCU due to wire length not being a concern. Due to the presence of other remote processes in this project, they may appear to be an obvious choice, but there are some important downsides to consider

However, they rely on wireless communication, which can be affected by connectivity issues, particularly when multiple valves are involved. Additionally, Bluetooth/Wifi solenoid valves typically consume more power than their wired counterparts due to the need to maintain an active wireless connection. Additionally, unless they are battery powered, they will still require connection to a power supply unit, meaning the reduction in wiring is not as impactful as it may seem, giving it must still be in close proximity to the power supply.

3.1.7.4 Solenoid Valve Actuation using a wireless LED driver

Another option for controlling the solenoid valves is to wire them directly to the output of the Shelly RGBW2, which is a multi-purpose driver that will be elaborated on later in the report. For now, it suffices to know that the RGBW2 is a wirelessly controlled device that can function as an on and off switch. It can

receive wireless signals and process them either by MTTQ programming or via an app created by the company Shelly. This potential solution allows for the option to use directly wired solenoid valves while leaving the problem of allowing the ESP32 to control its on and off cycle, as the RGBW2 will provide the valves with power (the Shelly itself is connected to the output of the AC to DC converter that will be discussed later), as well as turn them on and off via a wifi signal from the main MCU. There are other products that can also perform this function, which will be elaborated on further in the LED driver section, but for reasons that will be listed there as well, the Shelly is a naturally excellent option.

3.1.7.5 Solenoid Valve Part Comparison

Below is a description of the design constraints, the types of valves that meet the constraints, and then a comparison of these suitable electronic solenoid valves to decide which water valve will ultimately be used for the project.

3.1.7.6 Solenoid Valve Constraints

As mentioned previously, the solenoid valves for this application need to be somewhat fast in their actuation speed. Additionally, they should operate at the same voltage as the water pump for simplification in wiring. Furthermore, another important aspect is whether or not the valve is normally closed or not, as having to supply power in the most common operating condition will significantly lower the efficiency. Additionally, there is the distinction between the direct controlled and the wireless type. Practically speaking, although a bluetooth connection is useful in some ways, the need for the valves to be wired to power anyways takes away the greatest benefit of the bluetooth valves. Therefore, the ideal valve will be a 12V DC operated valve that is directly wired to the MCU. Additionally, because a ¼" schedule 40 pipe will be used, a valve with that size would be ideal, although converters can be used in case of a different valve size.

Another key factor to consider when selecting a solenoid valve is its pressure rating and flow coefficient (Cv value). The valve must be able to handle the expected water pressure in the system without leakage or failure. If the pressure exceeds the valve's rated capacity, it may not seal properly or could sustain damage over time. Additionally, the flow coefficient determines how much water can pass through the valve at a given pressure drop, impacting the overall efficiency of the irrigation system. A valve with too low of a Cv value may overly restrict water flow, reducing the effectiveness of the watering process. On the other hand, a valve with too high of a Cv value may result in excessive water usage or pressure loss elsewhere in the system. Therefore, selecting a valve that balances actuation speed, voltage compatibility, pipe size, and pressure/flow characteristics is essential for ensuring smooth and efficient operation. Three

product examples have been listed in the table below, where important specifications are shown to differentiate between them:

Table 11: Solenoid Valve part comparison

Model Number	DIGITEN DC 12V	VITON U.S. Solid 1/2" NPT Brass Electric Solenoid Valve	U.S. Solid Motorized Ball Valve- 1/2 inches Stainless
Cost	\$7.69	\$81.37	\$35.85
Normally Closed	Yes	Yes	No
Port Size	1/4"	1/2"	1/2"
Supply voltage	12V DC	12V DC	9V - 24V AC or DC

Based on the needed specifications, it is clear the DIGITEN 12V DC solenoid valve is the best fit. It is normally closed, has the cheapest cost, has the correct port size, and supports the supply voltage that will already be used for the water pump, making it the ideal fit for this project.

3.1.8 MCU Parts Comparison

When selecting the ideal microcontroller, we consider key factors such as Wi-Fi connectivity, I/O capabilities, and processing power. The system needs to wirelessly communicate with sensors, control actuators like water pumps and lighting, and efficiently process data while minimizing power consumption. We looked at microcontrollers that offer Wi-Fi capabilities, such as the ESP32-WROOM-32, ATmega328P, MSP430, and RP2040, but each comes with trade-offs in terms of GPIO availability, communication protocols, and energy efficiency. By comparing each microcontroller, we can determine the best fit for developing a personalized CPU that meets the specific requirements of our project.

Table 12: MCU Comparison

<i>Features</i>	<i>ESP32-Wroom-32</i>	<i>ATmega328P</i>	<i>MSP430</i>	<i>RP2040</i>
<i>CPU Cores</i>	Dual-Core Xtensa LX6	Single Core	Single Core	Dual ARM Cortex-M0+
<i>I/O Pins</i>	38 GPIO	23 PIO	24 PIO	30 GPIO
<i>Analog Inputs</i>	18 channels (12-bit ADC)	8 channel (10-bit ADC)	8 Channel (10-bit)	3 Channel (12-bit ADC)
<i>PWM</i>	16 Channels	6 Channels	8 Channels	16 Channels
<i>Communication Protocols</i>	SPI, UART, I2C, I2S,	UART, I2C, SPI	UART, I2C, SPI	UART, SPI, I2C
<i>Wi-fi capability</i>	802.11 b/g/n	N/A	N/A	N/A
<i>Bluetooth Capability</i>	BR/EDR and Bluetooth LE	N/A	N/A	N/A
<i>Power Consumption</i>	10µA Deep Sleep	0.2mA Active Mode	230µA Active Mode	30mA Active
<i>Memory</i>	4M Byte	32Kbyte	16K Byte	2M Byte
<i>RAM</i>	4M Byte	2KByte	512 Byte	264k Byte
<i>Operating Voltage</i>	3.0V-3.6V	1.8V-5.5V	1.8V-3.6V	3.3V

3.1.8.1 ESP32-Wroom-32

When we discuss the ESP32 we know that it is a powerful microcontroller that also is versatile making it one of the best choices for projects that need to be connected through WiFi and or bluetooth. Having a dual core processor allows the microcontroller to be able to handle real-time information at high processing power. It also has a large amount of GPIO and ADC pins that will allow us to use different components for the project. Although it has its complications for example how the ESP32 can be very complicated and complex to program and to debug.

3.1.8.2 ATmega328P

The ATmega328P is a microcontroller that is simple for electronic projects. Although we can see that it is less powerful than the ESP32, it is able to handle the basic needs for applications that do not need high power or real-time processing. The best thing about the ATmega328P is how it is easy to use and can quickly be programmed. But it has its limits, for example it doesn't have the Wi-Fi and bluetooth capabilities which hinders the user that needs multiple sensors that have to be connected. Knowing these things the ATmega328P would not be the best choice for this project.

3.1.8.3 MSP430

The MSP430 microcontroller is used for its low-power consumption which is good for applications that are battery powered. Having low powered systems allows the microcontroller to not use more energy than needed while still having efficient processing capabilities. Things that need to be monitored for a long period of time benefit the most for the MSP430. It is also versatile for interface because of the multiple analog and digital peripherals but it still is not as powerful as the ESP32. While the MSP430 is a great microcontroller it would not be the best choice for our project due to the fact that it does not have the Wi-Fi and bluetooth capabilities that we need for the project.

3.1.8.4 RP2040

RP2040 is a microcontroller that gives both good performance and uses power efficiently. It also allows the user to be able to do more complex things with the multiple sensor input and computation abilities because of the dual-core ARM Cortex-M0+ while having more GPIO pins. One thing that stood out about the RP2040 is the I/O pins that makes it easy for the user to implement different protocols for the sensor and the display itself.

However the problem with the RP2040 that makes it unfit for this project is the lack of Wi-Fi and bluetooth capabilities. Without those things we cannot have the ideal project where monitoring is necessary.

In conclusion, when looking for a microcontroller that is great for our project we needed to have features that gave us the ability to have Wi-Fi and bluetooth connections, high processing power, and low-power consumption. Out of the four microcontrollers that has been discussed we ended up choosing the ESP32-Wroom-32 due to the built-in Wi-Fi and bluetooth capabilities and its ability to monitor and control remotely.

3.1.9 Total System Energy Sources

The system will require a variety of energy sources; specifically, the moisture sensing PCB's will require battery power, the MCU will also require battery power, the lighting system and driver will need an energy source, as will the water pump and solenoid valves.

3.1.9.1 Lead-Acid Battery

Another approach to powering both the pumps and the valves would be a separate lead-acid battery. Lead-acid batteries store energy within the positive and negative plates internal to the battery; a chemical reaction occurs when

connected to a device, allowing electron flow (current) to travel towards the device, thereby powering it. In this project, one clear advantage of this power source is the portability, as this allows the pump and valves to function without the need for a nearby wall outlet to supply power. This is particularly useful in environments where electrical outlets are not readily available or in situations where the system needs to be mobile, such as in outdoor garden setups or temporary installations. Additionally, the battery provides an independent and uninterrupted power supply, ensuring the system can continue to function even during power outages.

However, this leads to some other problems; for example, the battery will need to be charged after enough usage, and improper maintenance could lead to reduced battery life or even failure. Furthermore, the battery itself has no built-in overcurrent protection, meaning that circuit breakers would be necessary to ensure the safety of the components downstream of the battery. Without proper protection, there is a risk of damaging sensitive components or causing fire hazards due to excessive current. Additionally, the limited capacity of a lead-acid battery may require more frequent charging, especially in systems that demand high power for long durations. This could lead to periods of downtime or the need for larger, more expensive batteries, further increasing the complexity and cost of the system. Furthermore, lead-acid batteries can be relatively bulky and heavy, which might make them less suitable for projects that prioritize compactness or portability. Regular maintenance and proper storage conditions, such as avoiding deep discharges or extreme temperatures, are necessary to maximize the lifespan and efficiency of the battery, adding another layer of complexity to the system's operation.

3.1.9.2 Lithium-Ion Battery

An alternative to the lead-acid battery for powering both the valves and the pump is a lithium-ion battery. Lithium-ion batteries store energy through the movement of lithium ions between the anode and cathode during charge and discharge cycles. When connected to a load, such as the valves or pump in this system, these ions create an electric current that can power devices effectively and efficiently. In the context of this project, one key advantage of lithium-ion batteries is their energy density—these batteries provide significantly more power for a given size and weight compared to lead-acid batteries. This makes them ideal for compact and portable systems, particularly in outdoor or space-constrained environments where reducing bulk is essential.

Another major advantage is their cycle life and charge efficiency. Lithium-ion batteries typically offer more charge cycles and faster charging times, allowing the system to be operational more frequently with less downtime. This can be

particularly beneficial in a smart irrigation system that is expected to maintain consistent functionality over extended periods. However, lithium-ion batteries come with their own set of considerations. They are more sensitive to overcharging and deep discharging, which necessitates the inclusion of a Battery Management System (BMS) to monitor temperature, voltage, and current in real-time. This increases the overall complexity and cost of the power system but is necessary for safe operation. Additionally, lithium-ion batteries are more expensive upfront than lead-acid alternatives, which may be a limiting factor depending on budget constraints. Nevertheless, the long-term benefits of reliability, compactness, and performance make lithium-ion batteries a strong contender for projects like this, especially if the goal is a more permanent, low-maintenance setup with minimal recharging interruptions.

3.1.9.3 CR2032 Coin Cell Battery

The CR2032 is a widely used coin cell battery known for its compact size, long shelf life, and steady voltage output. It is a 3V lithium-based battery, typically used in low-power electronics such as remote controls, key fobs, small sensors, and embedded systems. One of its primary advantages lies in its ability to supply a consistent voltage over time, even with minimal current draw, making it ideal for long-duration, low-power applications. Its lightweight form factor and small footprint make it particularly attractive for systems with limited space or those that require lightweight components. These batteries are also inexpensive and easy to replace, making them a practical choice for prototyping and small-scale projects.

CR2032 batteries do have limitations that must be acknowledged. For one, they offer a relatively low current output—typically maxing out at a few milliamps—which makes them unsuitable for devices with higher power demands such as motors, pumps, or bright LEDs. Additionally, while their capacity (typically around 220 mAh) is suitable for intermittent or standby operations, they are not intended for continuous or high-frequency use. Attempting to use a CR2032 to power components that exceed its current rating could result in voltage drops, rapid depletion, or even leakage. Therefore, they are best reserved for use in systems where current consumption is tightly controlled and minimized through efficient design.

In the context of this project, the CR2032 battery is an ideal power source for the soil moisture sensor PCBs. These boards are designed to draw very low amounts of current, especially when placed in sleep mode between readings, aligning perfectly with the CR2032's strengths. Since each sensor unit operates independently and requires wireless communication only occasionally, the battery's modest capacity is sufficient to power the board for an extended

period without needing frequent replacement. Furthermore, using individual CR2032 batteries for each sensor node eliminates the need for complex wiring or centralized power distribution, allowing for easier deployment and maintenance across various planting zones. The coin cell battery's small form factor also ensures that it won't interfere with the physical layout or mounting of the PCBs, keeping the overall design clean and unobtrusive.

3.1.9.4 PC IFR18650-1500 Battery

The PC IFR18650-1500 is a high-quality lithium iron phosphate (LiFePO₄) rechargeable battery that will be utilized to power the moisture sensing PCBs in this project. These batteries are a popular choice for embedded systems due to their compact size, high energy density, and long cycle life. The 18650 form factor is known for its versatility and is commonly used in various applications requiring a small, efficient power source. The PC IFR18650-1500 specifically provides a nominal voltage of 3.2V and a capacity of 1500mAh, making it suitable for low-power devices such as the moisture sensors.

One of the main advantages of using the PC IFR18650-1500 battery is its long cycle life. LiFePO₄ batteries generally provide over 2,000 charge cycles, ensuring a longer operational lifespan for the moisture sensing PCBs. This makes it ideal for our application, where long-term reliability is crucial. Additionally, the battery's high energy density allows it to provide consistent and reliable power, even for extended periods, without the need for frequent recharging. Its relatively low self-discharge rate ensures that the battery will retain charge over time, making it suitable for devices that may not be in constant use but need to remain operational when required.

However, this battery type is not without its limitations. Unlike a 12V power supply or wall-powered systems, the PC IFR18650-1500 has a lower voltage output of 3.2V, which requires the design to include a step-up converter to regulate the voltage to a level compatible with the PCBs. Despite this, the size and capacity of the PC IFR18650-1500 are well-suited for this project's needs, as it will provide adequate power to the moisture sensors while maintaining a compact form factor. Furthermore, the relatively low maintenance requirements of LiFePO₄ batteries and their safety advantages, such as thermal stability and resistance to overcharging, make them a safe and reliable choice for this embedded sensing application.

3.1.9.5 Line Power

Line Power, more colloquially known as "wall power," is the standard 120V AC provided by the local electric utility in the United States. This power source is

ubiquitous in most households and provides a reliable means of powering various electrical appliances. As both the pump and valves can be found in 120V AC varieties, it is a viable and convenient powering option for many applications, including this project. One significant benefit of using line power is the elimination of the need for additional components such as batteries or AC to DC converters, making it a more straightforward solution. Additionally, the output frequency and phase of wall power are incredibly stable, which ensures consistent performance of connected devices. In terms of cost, wall power is also highly economical, with low rates for energy consumption (per kilowatt-hour), making it an attractive option for long-term operation. This stability and affordability contribute to its widespread use in both residential and industrial settings.

However, despite these benefits, there are some notable downsides to using line power in this specific application. Devices operating at higher voltages, such as those powered by 120V AC, generally pose greater safety risks. In particular, any device powered by line voltage can lead to electrical hazards if not properly managed, which requires added caution in installation and usage. If multiple devices are plugged into the same circuit via an extension cord or power strip, this can increase the load on the circuit, potentially leading to overheating or even fire hazards. This is particularly true if the circuit is not protected by a dedicated circuit breaker, which in many cases is not under the direct control of the user. In the context of this project, another issue arises when considering the power requirements of solenoid valves and pumps that run on 120V AC. The higher voltage may actually create challenges, such as causing too much pressure for the solenoid valve to operate safely, potentially damaging the system or causing leaks. The higher energy output of 120V AC can also make precise control of the system more difficult, especially if fine-tuning the operation of the valves and pumps is necessary for optimal performance.

3.1.9.6 Power Selection Table

Power Sources Used	Series Lithium Ion 3.7V Batteries	12V DC Power (Rectified and Converted from line power)
Used For	I2C Sensors, Moisture Sensors, main board	Solenoid Valves, LED lights, each Shelly RGBW2, Water Pump
Associated Hardware Component	3.3V Regulator (See Chapter 6)	120AC to 12DC converter

3.1.9.7 Power Distribution Table

Hardware Components	Temp/Humidity Sensor (x3)	Light Sensor (x3)	Moisture Sensor (x3)	LED Light Strip (x3)	Solenoid Valve (x3)	Water Pump	Shelly RGBW2	Total Power Consumed
Power Consumption	2.5mW	1.5mW	10mW	20W (x3 = 60W)	5W (x3 = 15W)	65W Peak	500mW	90.514 W
Supply Voltage	3.3V	3.3V	3.3V	12V	12V	12V	12V	N/A

3.1.10 Water Delivery to Plants: Pump and Valves Power Source

For both the pump and the solenoid valves, power needs to be delivered to both sets of devices in order for them to work. Common pump voltages for these applications are 12V DC, 24V DC, 110V AC, and 120V AC. Generally speaking, the smaller pumps will be controlled via a DC power supply, whereas systems that must deliver high flow rates or sustained flow will be powered using an AC source. For the solenoid valves, it is very similar; those with very high switching speeds, or those expected to handle very high pressure ratings, will typically be driven with a higher voltage AC source.

3.1.10.1 AC to DC Converters

When driving a pump for the system, one approach is to use an AC to DC converter that plugs into the wall outlets; this will first transform the voltage down to the desired level, typically 12V or 24V, and then rectify it in order to produce DC. This can be done in many fashions depending on the application; for example, both single phase and three phase power can be converted, and the type of rectification can even be varied; both half wave and full wave rectifiers are commercially available. For this particular application, the only viable option would be a full wave rectified, single phase converter.

Based on the lighting constraints, a 12V AC to DC converter would need to be used. Compared to the 24V converters, the 12V variety is typically smaller, quieter, and will be compatible with far more devices, as 12V is one of the most common voltage values chosen for devices in such an application. However, one downside is that the output current in the 12V system will be proportionally

higher, and generally speaking, the 12V converters have a lower power rating internally.

Another important consideration when using an AC to DC converter is efficiency and heat dissipation. Converting 120V AC to 12V DC involves power losses due to heat generation within the transformer and rectifier components. This heat must be managed effectively to prevent overheating, which could impact the longevity and reliability of the converter. Many modern converters incorporate switching power supply designs, which offer improved efficiency over traditional linear power supplies by minimizing heat generation and maximizing power conversion efficiency. Additionally, proper ventilation and heat sinking may be required depending on the power rating of the converter to ensure stable operation. In the context of this project, selecting a high-efficiency AC to DC converter with adequate thermal management is crucial to maintaining reliable performance, especially in a system where continuous operation is required for both the pump and lighting components. The images below show first a common wiring schematic for such converters; the second shows the physical appearance of the device to gain a scale of size and weight for the product. This will be expected to rest within the shelving system.

3.1.10.2 AC to DC Technology Comparisons

As discussed above, various technologies exist for actually converting AC power to DC power. For example, this can be done by using half wave rectifiers with or without capacitor smoothing, or can be done with full wave rectifiers that can also use or not use capacitor output smoothing, and which specific technology to use depends heavily on the specific application. In this particular application, in order to double the amount of time that the light strips are on for (which will increase the light intensity/meter squared over some given time interval), the lights will need to be powered by a device that will have it at full output voltage more often than every other cycle, which rules out the use of half wave rectifiers. Furthermore, the project demands high efficiency and low leakage current for the AC to DC converter, as current draw will already be quite high, and leakage cannot be afforded without significantly upsizing the converter itself. This is important because heat dissipation is a natural concern, given the converter will be inside of a compartment near other electronics. Based on heating specifications of wired water systems, which will be discussed in Chapter 4, the device should ideally not be any warmer than 170F at absolute maximum power draw, which will be determined by summing the current draw of every device and sizing the device appropriately. We can summarize the technology positives and negatives in the table below:

Table 13: AC to DC converter technology comparison

	Suitable for Project design	Power Drawn	"Smoothness" of output signal
Half wave rectified	No	Less than Full wave rectified	N/A
Full wave rectified	Yes	More than Half wave rectified	N/A
Output Smoothing Capacitors	Yes	More than without smoothing capacitors	Smoother, better for sensitive equipment
Without output smoothing capacitors	Yes	Less than with smoothing capacitors	Less smooth, only suitable for unsensitive equipment

3.1.10.3 AC to DC Converter Part Comparison and Selection

Now that the most ideal technology has been selected, it is appropriate to determine which specific converters meet these requirements. However, a critical constraint must first be discussed, which is the overall power drawn by the system. As is standard engineering practice, the worst case draw of the system is to be considered when determining the expected load from the system. In this case, the following draws need to be considered: The current draw from the water pump, the current draw from three separate electronic solenoid valves, the current draw from one ESP32 MCU that is using Wifi constantly to communicate, the current draw for the LED light driver, the current draw for the Shelly driving the solenoid valves, and finally, the current draw from each individual LED light strip. Current draw is being listed here because every device will use an input voltage of 12V, meaning the currents can simply be summed and multiplied by 12 to find the total power consumption. The pump will draw a maximum of 7 amps, the valves will draw a maximum of 0.4 amps each, meaning that all 3 will draw 1.2 amps at most, the two Shellys used (one for driving the lights and one for driving the solenoid valves) will use 80 milliamps each, each WS2811 LED light strip will use 7.5 amps (meaning 22.5 amps for all three light strips, and the ESP32 used as the main MCU cannot ever draw more than about one amp without causing it damage, even in applications more power consumptive than this particular project. Therefore, the total current that can be drawn will be 30.9 amps, and therefore a $30.9 \times 12 = 369$ Watt device will be needed at minimum. However, because thermal properties of

these converters are not linear, it is never safe to be even within ten percent of the maximum power rating; the maximum rating is meant to describe what power it can supply before internal components are damaged simply by the current, not from the ensuing heating. With the 10% constraint, the specific Ac to Dc converter must be able to supply a maximum of 407 watts. With this information, in combination with the requirements discussed in the technologies section, the following five converters will be considered:

Table 14: AC to DC power converter selection

Specific AC to DC converter	XINCOL AC to DC Converter 12V 500W	inShareplus 12V 500W Power Supply	Makelele MS-500-1 2 Power Supply	MEAN WELL SE-600-12 Power Supply	ALITOVE 12V 50A 600W Power Supply
Meets Power Requirement?	Yes	Yes	Yes	Yes	Yes
Heat Dissipation Characteristics	Poor	Excellent	Excellent	Excellent	Moderate
Footprint	11.43" (L) × 6.18" (W) × 2.57" (H)	10.2" (L) × 3.39" (W) × 2.16" (H)	8.61" (L) × 7.43" (W) × 4.88" (H)	9.72" (L) × 5.00" (W) × 2.50" (H)	9.41" (L) × 6.05" (W) × 2.93" (H)
Supports 120V AC at input?	Yes	Yes	Yes	Yes	Yes

With these parameters in mind, the ALITOVE converter has been selected due to it being well above the maximum power threshold, as well as having a more moderate price than the others, as well as having a sufficiently small footprint.

3.1.10.4 DC to DC Converters

DC to DC converters are essential components in modern electronic designs, particularly in applications where different parts of a system require different voltage levels for optimal operation. These converters allow for efficient conversion of a higher or lower DC voltage to another DC voltage, making them crucial in regulating power within a circuit. In this project, DC to DC converters will be employed on the moisture sensing PCBs to step up or step down the voltage from the primary power supply (provided by batteries such as the PC IFR18650-1500 or an external source) to a stable level suitable for the sensors and other integrated components.

The primary function of a DC to DC converter is to regulate the voltage supplied to the various components of the circuit, ensuring that each part of the system receives the appropriate voltage required for optimal performance. For instance, many sensors and microcontrollers operate at lower voltage levels than the power source, which necessitates the use of a voltage regulator to step down the power. Conversely, when higher voltage is required by certain parts of the circuit, a DC to DC converter can efficiently step up the voltage to meet those needs. This ensures that the entire system operates safely, with all components receiving the proper power supply to avoid damage or inefficient performance.

One of the key advantages of using DC to DC converters in a PCB design is their efficiency. These converters can operate with high efficiency (up to 90% or more) by reducing power loss during the voltage conversion process, which is particularly important in battery-powered systems like the moisture sensors in this project. This efficiency leads to longer battery life, making the system more sustainable and reducing the need for frequent recharging. Furthermore, DC to DC converters are typically compact, lightweight, and available in a wide range of output voltages, making them ideal for embedded systems. Their integration into the moisture sensing PCBs will ensure that the components are powered consistently and reliably, even in battery-powered configurations.

Additionally, modern DC to DC converters are equipped with various protective features such as overcurrent protection, thermal shutdown, and short circuit protection, all of which contribute to the overall safety and reliability of the system. By including a DC to DC converter in the design, we are able to safeguard the system from power surges or spikes, while maintaining stable operation throughout the device's lifespan. These converters are also typically designed to minimize noise, which is important when dealing with sensitive components like sensors that require precise readings.

3.1.10.5 Linear Regulators

Linear regulators are a type of DC to DC converter commonly used in electronic designs to provide a stable output voltage from a varying input voltage. These regulators function by dissipating excess power as heat, and they maintain the output voltage at a consistent level by adjusting the flow of current through a pass element (typically a transistor). While linear regulators are simple in design and provide a clean, low-noise output, they are generally less efficient compared to switching regulators (such as buck or boost converters), especially in applications where there is a significant difference between the input and output voltage.

The primary advantage of linear regulators is their simplicity and the quality of the output voltage. They can provide a very stable and low-noise voltage, which makes them ideal for applications where power purity is crucial, such as in analog circuits or audio equipment. However, this benefit comes at a cost. Linear regulators dissipate the difference between the input and output voltage as heat, meaning they are inefficient when there is a significant difference in voltage levels. For this project, linear regulators are not an ideal choice due to the relatively high power requirements of the system, especially considering that the moisture sensing PCBs will be powered by small batteries (such as the PC IFR18650-1500). In battery-powered designs, efficiency is paramount in order to prolong battery life and minimize the need for frequent recharging. Using a linear regulator in such a system would lead to significant energy losses, as much of the power would be wasted as heat. This will reduce the operational time of the device, making it impractical for this real-world application where battery longevity is important.

Additionally, linear regulators tend to generate more heat compared to switching regulators, which could be problematic for compact systems like the moisture sensing PCBs. Excess heat could affect the performance of nearby components and even lead to thermal damage if not managed properly. The inefficiency of linear regulators, particularly when there is a significant difference between input and output voltages, makes them a less suitable option for the power conversion requirements of our project.

For these reasons, we have chosen not to use linear regulators in our design. Instead, we are opting for more efficient DC to DC converters, such as switching regulators, which provide much higher efficiency by converting excess voltage into usable power, rather than dissipating it as heat. Switching regulators are more appropriate for battery-operated systems, as they can step up or step down the voltage while minimizing energy loss, ensuring longer battery life and reliable operation.

3.1.10.6 Switching Regulators

A switching regulator is a type of DC-DC converter that uses high-frequency switching to convert input voltage to a desired output voltage. Unlike linear regulators, which dissipate excess voltage as heat, switching regulators are highly efficient because they transfer energy through inductive or capacitive components rather than resistive ones. This efficiency is typically in the range of 80-90%, or even higher, depending on the design. Switching regulators work by rapidly switching the input voltage on and off using a transistor (such as a MOSFET), and using inductors and capacitors to store and release energy to regulate the output voltage.

One of the primary advantages of switching regulators is their efficiency. Since they don't dissipate energy as heat, they can handle larger voltage drops without requiring significant heat dissipation, which makes them ideal for applications where power efficiency is crucial, such as battery-powered systems, portable electronics, or high-power devices. Additionally, switching regulators can step up (boost), step down (buck), or invert the voltage depending on the configuration, making them highly versatile in different power supply applications.

3.1.10.7 DC to DC converter Part Comparison

3.1.10.7.1 TPS613221A

The TPS613221ADBVT is a boost converter switching regulator produced by Texas Instruments. It is specifically designed to step up the input voltage to a higher output voltage with high efficiency. This device is ideal for low-voltage power supply applications, such as battery-powered systems. It operates at an input voltage range of 0.9V to 5.5V and can boost this to an output voltage of up to 5.5V, making it suitable for powering devices that require a stable voltage from a lower supply voltage, such as sensors or microcontrollers in embedded systems.

The TPS613221ADBVT has a high efficiency, typically reaching 90% or higher, depending on the input-output voltage difference with a current output limit of 1.2A. This high efficiency is essential for battery-powered applications, as it ensures minimal energy loss and helps extend battery life. The TPS613221ADBVT also integrates several features that make it easier to design with, such as automatic switching between pulse-width modulation (PWM) mode and pulse-frequency modulation (PFM) mode for optimal efficiency under varying load conditions. It's also capable of maintaining a stable output even

under dynamic load changes, ensuring that sensitive components powered by the regulator receive consistent voltage.

However, despite its benefits, switching regulators like the TPS613221ADBVT can introduce noise into the power supply due to the switching frequency, which can be problematic in highly sensitive applications. Proper filtering and layout design are required to minimize this noise. Still, for most embedded and IoT applications, the TPS613221ADBVT provides an ideal balance of efficiency, performance, and compactness, making it well-suited for the moisture-sensing PCBs in our project where battery life and efficiency are important considerations.

3.1.10.7.2 LM2576

Another potential option for a switching regulator is the LM2576 buck converter produced by Texas Instruments. The LM2576 is designed to convert a higher input voltage to a lower output voltage, and is often used in applications requiring efficient voltage conversion from a larger voltage supply to a smaller output voltage. The LM2576 operates with a wide input voltage range from 7-60V and can deliver output voltages of 3.3V, 5V, 12V, 15V which makes it a viable candidate for powering sensors or other low-voltage devices. It features a 52-kHz switching frequency, allowing for smaller external components like inductors and capacitors, and is capable of providing a maximum output current of up to 3A.

While the LM2576 is efficient for many high-power applications, it has a few limitations that make it ideal for our specific moisture sensing PCBs. The LM2576 has a high maximum output current capability compared to the TPS613221ADBVT, which could be a limiting factor if the moisture sensor boards require higher current for reliable operation, especially when multiple sensors are connected. Additionally, while the LM2576 is compact and efficient, it tends to generate more noise than the TPS613221ADBVT due to its higher switching frequency, which could potentially interfere with the analog signals from the moisture sensors, compromising the accuracy of readings.

3.1.10.7.3 LTC3703

Table 15: Regulator Part Selection

Regulator Selection	TPS613221A	LM2576	LTC3703
Supply	0.9V-5.5V	7V-60V	4V-20V

Voltage Range			
Output Voltage Range	1.8V-5.5V	3.3V-15V	0.8V-24V
Max Current Output	1.6A	3.0A	5A
Efficiency	High but steep dropoff when current output is greater than 0.1A	High	High
Cost	Low	Medium	High

3.1.11 Software Development Technology

In the sense of our project, the driving software logic for basic functionality revolves around the embedded software which will be driving the entire main plant environment and running the soil moisture sensors. Therefore, after comparing the chips and selecting the ESP32-WROOM chip for our project, it became clear that the next step would involve mapping out the software development needs and approaches. For this decision, it would involve comparison and selection of both the language and the IDE.

Furthermore, to ensure our project's full functionality, a comprehensive user interface is essential for seamless data entry and system monitoring. Real-time communication is crucial, enabling users to specify the environmental requirements for each individual plant environment. Additionally, optimal implementation requires users to monitor system performance to confirm proper environmental settings, track power consumption for financial awareness, and maintain overall interest in their plants' well-being.

For convenient and fast access, we have determined that a mobile application is the most practical solution. Since this project is designed for flexible placement in various locations, providing portable access through a mobile platform aligns best with user needs. We believe the demand for desktop access will be minimal, making a mobile-focused solution the most efficient and user-friendly choice.

3.1.11.1 Embedded Software Technology Comparisons

Before deciding the IDE, which actually holds quite a small bearing on the project's success, we elected to first decide on the language to be implemented in the main MCU and the soil moisture sensors. Since these components will rely on interfacing with each other in order to run the entire ecosystem, there was no question as to whether or not they would be developed in the same language.

Table 16: Comparison of Embedded Language Options for the ESP32-WROOM

	Pros	Cons
C++	High efficiency and performance; object-oriented design support (features support module design); supported by multiple frameworks like Arduino and ESP-IDF; extensive libraries and community support and resources	Manual memory management is required; overall, this language is determined to have a higher complexity than others (i.e. Python)
C	Deemed to be a lightweight language and offers direct hardware access; pretty standard for embedded systems	Lacks a lot of modern programming paradigms; as projects grow larger, the language requirements result in increased complexity
MicroPython	Syntax is simplified, which offers easier development; interactive Read-Eval-Print-Loop (REPL) is supported for rapid prototyping	Execution speed is slower than C++; hardware control is severely limited; library availability is minimal
Espruino (JavaScript)	Can show suitable for event-driven applications; high familiarity for web	Higher runtime overhead; less widespread adoption for incorporation into the

	developers expanding to embedded systems design	ESP32
--	---	-------

Due to its efficiency, compatibility with multiple frameworks and the availability of extensive embedded libraries, our group decided on C++ as our language choice. Given that three quarters of our group members have familiarity with C++ in embedded development, with one member strongly advanced in Arduino Uno development, it seemed that this choice would offer better implementation understanding on behalf of the group. In addition, one of the most driving forces behind this decision was the community support and resources, as C++ in embedded programming is widespread and quite common. From example availability to increased chances of debugging support, this was an extremely attractive option bringing it to be our first choice.

Once the language was selected, we moved on to determining how we would put the language to use for our ESP32. Therefore, the next option was choosing our Integrated Development Environment (IDE). To reiterate, this decision is less crucial and, in fact, if each member of the group wanted to develop in a different environment, given that the language rules never change across platforms, it would not make an impact on the operation of the system. However, our group decided it would be best to choose an IDE and all uniformly operate on the same one since it would ensure access to all the same features, making development across individual computers easier to maintain with debugging assistance from each other, which would likely require knowledge of how the IDE works.

Table 17: Comparison of IDEs

	Pros	Cons
Arduino	Requires minimal setup; extensive library; increased community support; ideal for rapid prototyping	Limited debugging tools; less hardware control compared to others
ESP-IDF	Official framework for ESP32; provide low-level hardware control; supports advanced JTAG debugging	Higher learning curve; complex setup is required

VSCode (via the PlatformIO extension)	Simple and beginner-friendly interface; interactive REPL for debugging	More complex requirements for the initial configuration
--	--	---

The Arduino IDE seemed to be most suitable for general development in our case due to its ease of use, vast community support, and quick prototyping capabilities. Especially because of the time restrictions enforced by this project, our group decided to avoid the extremely advanced and complex IDEs that would create difficulty in the software development requirements of the project.

3.1.11.2 App Development Technology Comparisons

In order to decide the best way to create the mobile element, we researched the possible development platforms and technologies available on several different criterion:

Table 18: Basic Comparison of App Development Technologies

	Supported Mobile Operating Systems	Coding Language(s) Required	Implementation Difficulty
React Native	iOS and Android	JavaScript	Moderate
Flutter	iOS and Android	Dart	Moderate
Swift	iOS only	Swift	High
Kotlin	Android only	Kotlin	High
Ionic	iOS and Android	HTML, CSS, and JavaScript	Low to Moderate
Cordova	iOS and Android	HTML, CSS, and JavaScript	Low to Moderate
Progressive Web Apps (WPAs)	Any OS with a modern browser technology	HTML, CSS, and JavaScript	Low

Table 19: In-Depth Advantages/Disadvantages of Each Technology

	Advantages	Disadvantages	ESP32 Integration
React Native	Strong community support; single code base for multiple platforms; reusable components; hot reload for instant UI updates	Performance lags compared to native apps; reliance on third-party plugins; development cost may increase based on project needs	Possible via third-party libraries or native modules
Flutter	High performance; single code base for multiple platforms; rich set of customizable widgets; hot reload for real-time code changes	Limited access to third-party libraries; larger app size; development cost may increase based on project scope	Available via plugins or native modules
Swift	High performance; full access to iOS features, seamless integration with iOS APIs	Limited only to the iOS environment; requires a separate codebase for Android	Direct integration with ESP32 via BLE or Wi-Fi Connection
Kotlin	High performance; full access to Android features; seamless integration with Android APIs	Limited to the Android environment; requires a separate codebase for iOS	Direct integration with ESP32 via BLE or Wi-Fi Connection

Ionic	Faster development; shared codebase for multiple platforms; utilizes web technologies; extensive library of UI components	Limitations on performance; relies on plugins for native features	Integration through plugins; may require native code
Cordova	Access to native device features via plugins; single codebase for multiple platforms; wraps web apps in native containers	Limitations on performance; relies on plugins for native features	Integration through plugins; may require native code
Progressive Web Apps (WPAs)	Single codebase; immediate updates; provide an app-like experience without the need for app store deployment	Extremely limited access to device features; dependent on browser capabilities and sensitive to previously set privacy modes	Indirect integration via web protocols such as HTTP, WebSockets, etc.

To comply with our project's needs, seamless integration with the ESP32 microcontroller for real-time user interaction was non-negotiable in determining which application development technology to use. In this capacity, Flutter and React Native stood out as strong candidates. Since both frameworks allow for cross-platform development within a single codebase, they also represent reduced cost through development time. In addition, these hot reload capabilities enable real-time code changes to cater to the immediate feedback and updates we hoped to embed within our overall system. In both Swift and Kotlin, the direct integration with the ESP32 was an especially attractive feature for the requirements of our project. However, WPAs were essentially eliminated from our purview due to their inability to incorporate direct or plugin-based connection to the MCU.

Despite Swift and Kotlin's appeal based on the direct ESP32 integration, their restricted mobile OS environment capabilities were unattractive to our group, thus removing them as contenders for our mobile development environment. This is because the project's need for real-time responsiveness to the user's input and efficient ESP32 integration may be extremely inhibited by a codebase lacking unification. In addition, lack of familiarity and resources regarding both Ionic and Cordova removed them from the list of desired technologies.

Between Flutter and React Native, both seemed to be great options for development of the mobile app. However, Flutter wound up being our ultimate choice as the technical needs meshed better with our hopes for the smart garden project. In addition to the unified codebase for both OS environments, the Hot Reload, high performance, Flutter's flexible UI Design and through its widget system ensures ease of implementation. Furthermore, the plugin ecosystem support for BLE, Wi-Fi, and Firebase all are ideal supporting elements for the real-time communication required for our project with the ESP32. Despite requiring native code writing to integrate complex hardware features, Flutter's plugin ecosystem is able to simplify this process significantly when linking the software and hardware communication.

Flutter's architecture helps F.L.O.W.E.R.S. meet its full potential through its architecture, performance characteristics and development tools. Dart compilation within Flutter is placed directly into native ARM machine code via Ahead-of-Time (AOT) compilation. This process is able to bypass JavaScript bridges, unlike React Native, which results in faster startup times and better runtime performance; with a system like ours, requiring frequent updates from the ESP32 module, these are crucial. Since no JavaScript bridge is needed to interact with native APIs, the latency is significantly reduced in order to enhance performance during continuous sensor updates or other control unit actions. State management libraries such as Riverpod, Bloc, or Provider can simplify the data handling from real-time data streams received from the sensors which would ensure an immediate update for the UI. Riverpod may be especially helpful as the app would be able to establish a continuous Wi-Fi connection to the ESP-32. Riverpod can also assist in storing the incoming data from soil moisture sensors into the state objects. In addition, the Skia UI element implementation for graphics is ideal for managing complex animations and efficiently displaying UI changes which meets the direct needs of displaying dynamic data for the garden such as soil moisture levels, water levels, light status, or power consumption in real time. Plugin ecosystems available for flutter are robust and versatile including the `flutter_blue` plugin for Bluetooth Low Energy (BLE) communication and `wifi_iot` for Wi-Fi control and direct ESP32 communication. In the case that these plugins are unable to perform the desired communication tasks between Flutter and the ESP32 module, a failsafe exists

by adding platform-native code directly with Java or Kotlin for Android and Swift or Objective-C for iOS. In this case, the direct communication with the ESP32 hardware would be established either through BLE, Wi-Fi, or HTTP depending on the software implementation.

Flutter's UI toolkit will be especially helpful in conveying the real-time data to users given its flexible UI toolkit. From interactive graphs representing soil moisture trends or water requirements of the system to custom alert widgets for low levels as well as sliders or toggles for lighting or watering adjustments, the user interface capabilities flutter is able to offer bring it to be our champion in terms of mobile app development.

3.1.11.3 Storage in the Mobile App

Within this project, there will be a few critical points at which data within the app must be saved from one session to the next. For achieving this, we compared the possible options.

Table 20: Basic Comparison of App Storage Technologies

	Local Storage	Cloud Sync
Data Availability	Offline or manually initiated sync cycles	Online and automatic sync across devices
Access Speed	Instant - from device memory	Can be slower - dependent on internet connection
Offline Support	Full offline support	Offline caching only
Data Sharing	Difficult to be shared between devices	Instant sharing

Table 21: Local Storage Options in Flutter

	Hive	Drift	SQLite
Storage Type	Key-Value Store	Relational, SQL	Relational, SQL
Best Used For	Lightweight	Complex	Standard SQL

	structured data requiring fast operations	queries and normalized data formats	needs, manual query handling
Setup Difficulty	Very Easy	Moderate	Moderate
Performance	Very fast; integrated within Dart	Fast with smart caching enabled	Fast
Querying	Standard CRUD operations with key/object pairs in Dart	SQL similarities crossed over with Dart syntax	Pure SQL strings

3.1.12 PCB Design Software

PCB design is a crucial step for developing our project. Which is why choosing the best development software for designing our PCB is important. Some of the options that we looked at when making this decision were Fusion360, KiCad, and Altium as possible choices. These options offer a wide range of features and benefits, but the decision largely depends on factors such as usability, accessibility, learning curve, and support resources. After evaluating the key features of each software, we were able to weigh the pros and cons to find the most suitable option that meets our team's needs while considering time constraints, technical capabilities, and future project requirements. The goal is to select a tool that will allow us to design, test, and manufacture a high-quality PCB in an efficient and cost-effective manner.

3.1.12.1 Fusion360

Fusion360 was selected as the preferred software due to our prior experience using it in Junior Design. UCF helps provide us access to the educational version at no cost, enabling us to incorporate more complex design features as needed. Using this rather than using the free version. Fusion360 also offers cloud-based collaboration, allowing multiple team members to work on the design from their own computer without having to send over files to each other. This cloud-based feature is especially valuable in collaborative environments, as it ensures that everyone is working on the most up-to-date version of the design in real time. Additionally, Fusion360 offers integrated CAD tools that allow for the

seamless design of both the PCB and mechanical components in a unified workspace, which further enhances our workflow and reduces potential compatibility issues during the design phase.

3.1.12.2 KiCad

KiCad is an open-source PCB design software that is available for free to all users. It offers extensive online resources and learning materials. Although KiCad has a steeper learning curve when compared to Fusion360, these resources can provide support with any learning that may be needed. However, since our team has no prior experience with KiCad, additional time would be needed to learn this tool. Despite the learning curve, KiCad is often praised for its flexibility and feature set, offering capabilities that rival some commercial software. For teams willing to invest time into learning the program, it can be an incredibly powerful tool with no cost involved, making it especially attractive for long-term projects or when a budget is a primary consideration. The open-source nature of KiCad also means that it is continually updated by a community of developers, providing frequent improvements and bug fixes.

3.1.12.3 Altium

Altium is a very advanced PCB design software widely used in professional industries. While UCF does not provide access to this software, students can use their six-month free trial. Due to its advanced capabilities, Altium may require a high-performance computer to handle any complex designs. Although our team is not familiar with Altium, there is a wide amount of online learning material that is available if we choose to invest time into learning this software. Altium's advanced tools, such as signal integrity analysis, advanced routing algorithms, and component management, make it an excellent choice for high-performance and professional-grade PCB designs. However, the combination of the cost, steep learning curve, and the need for high-specification hardware makes it less ideal for a smaller team with time constraints and a need for rapid development.

3.1.12.4 PCB Software Selection

Fusion360 is a program that all of us are familiar with and comfortable using. While Altium is a valuable tool that we can use, the learning curve needed and unfamiliarity with the program steers us away from this choice. For KiCad, it is a free resource for us to use and can be applicable for our project, but our group is not experienced enough with the program and would need to dedicate time to learn. Our choice is Fusion360 because we have the most experience with this software and do not need time to learn all the necessary features. Given that our

primary goal is to design a functional and efficient PCB within a reasonable timeframe, using Fusion360 minimizes the potential risks of delays or errors that could arise from working with unfamiliar software. Additionally, Fusion360's integrated collaboration features provide a practical advantage in managing our design process as a team, ensuring that we can work seamlessly together and make necessary adjustments on the fly. The following table shows the software comparison between the platforms.

3.1.13 Temperature/ Humidity Sensing Technology

In our project we will have a Temperature/Humidity sensor for each of the plants to allow the plants to grow within their respective environment. These sensors will also send a reading to an app and the user can then also adjust the threshold as necessary. For choosing a temperature/humidity sensor we looked into thermistor and semiconductor-based.

3.1.13.1 Thermistor

A thermistor is a type of temperature sensor made from semiconductor materials whose resistance varies significantly with temperature. There are two main types: NTC (Negative Temperature Coefficient) thermistors, where resistance decreases as temperature increases, and PTC (Positive Temperature Coefficient) thermistors, where resistance increases with temperature. Due to their high sensitivity to small temperature changes, NTC thermistors are commonly used in temperature sensing, control systems, and electronic protection circuits. We can use the equation below to see the relationship between resistance and temperature where T is temperature, R is resistance, ABC are Calibration constants:

$$1/T = A + B \ln(R) + C(\ln(R))^3$$

Equation 3: Relationship between Resistance and Temperature

Thermistors are highly sensitive to small temperature changes, making them ideal for precise measurements in narrow temperature ranges. They are compact, inexpensive, and offer fast response times. However, their nonlinear behavior requires calibration and limits their accuracy over wide temperature ranges. Additionally, thermistors can be affected by self-heating and are not suitable for extreme temperature environments.

3.1.13.2 Semiconductor-based

Semiconductor-based temperature and humidity IC sensors use integrated circuits to measure environmental conditions with high accuracy and digital output. These sensors

often rely on a combination of bandgap references for temperature and capacitive sensing elements for humidity, all built on a silicon substrate. They are compact, require minimal external components, and typically communicate via digital protocols like I²C or SPI, making them ideal for embedded systems and IoT applications.

However, they typically have a limited operating temperature and humidity range compared to more industrial-grade sensors. Additionally, their accuracy may degrade in harsh or high-moisture environments, and they can be more expensive than simpler analog sensors.

3.1.13.3 AHT 20

The AHT20 is a modern, digital temperature and humidity sensor developed by ASAIR, designed for precision environmental monitoring in compact and cost-sensitive applications. It uses a capacitive humidity sensor and a bandgap temperature sensor, both integrated on a single CMOS chip, with built-in signal processing and calibration. The sensor communicates via the I²C protocol, making it easy to interface with most microcontrollers and development boards. It features a typical humidity accuracy of $\pm 2\%$ RH and temperature accuracy of $\pm 0.3^{\circ}\text{C}$, with a response time of under 8 seconds. The AHT20 operates within a humidity range of 10–90% RH and a temperature range of -40°C to 85°C , making it suitable for a variety of indoor and moderate outdoor applications. It is factory calibrated, which eliminates the need for manual calibration and ensures consistent performance across units. The compact size and low power consumption also make it a strong candidate for battery-powered and space-limited devices like smart thermostats, weather stations, and smart gardening systems.

3.1.13.4 DHT 11

The DHT11 is a low-cost, digital temperature and humidity sensor commonly used in beginner electronics projects and basic environmental monitoring applications. It uses a capacitive humidity sensor and a thermistor to measure humidity and temperature respectively, with the data converted and transmitted using a single-wire digital protocol. The DHT11 has a humidity range of 20–80% RH with an accuracy of $\pm 5\%$ RH, and a temperature range of 0 – 50°C with an accuracy of $\pm 2^{\circ}\text{C}$. Although less accurate and slower than more advanced sensors, it is extremely simple to use and widely supported by libraries across platforms like Arduino and Raspberry Pi. The sensor has a sampling rate of about 1 reading per second and is factory calibrated, which means users do not need to handle analog-to-digital conversion or sensor calibration manually. Due to its low cost and ease of use, the DHT11 is ideal for educational purposes and basic indoor climate sensing where precision is not critical.

3.1.13.5 SHT 45

The SHT45 is a high-precision, digital temperature and humidity sensor developed by Sensirion, part of their 4th-generation humidity sensor series. It features a capacitive humidity sensor and a bandgap temperature sensor, integrated into a compact, factory-calibrated, and fully digital module. The SHT45 offers exceptional accuracy, with a typical humidity accuracy of $\pm 1.0\%$ RH and a temperature accuracy of $\pm 0.1^\circ\text{C}$, making it suitable for demanding applications such as medical devices, smart home systems, data centers, and precision agriculture. It communicates via the I²C interface, supports multiple I²C addresses, and includes configurable measurement modes (e.g., single-shot or periodic), allowing designers to balance power and performance based on their needs. The sensor has a wide operating range: 0–100% RH for humidity and -40°C to $+125^\circ\text{C}$ for temperature. Additionally, the SHT45 is highly resistant to long-term drift, chemical exposure, and contamination, thanks to Sensirion's proprietary sensor protection and advanced on-chip signal processing.

3.1.14 Communication Protocol

For embedded systems communication protocol is an important tool. Different types of communication protocol can be used in different scenarios and each have their own advantages. The types of communication protocol that we will be looking into are I2C, SPI, and UART.

3.1.14.1 I2C

I2C is a form of communication that works with two wires, SDA and SCL, which are used to send and receive data. I2C allows for multiple different devices to be hooked up to the same clock (SCL) line, while the data (SDA) line sends the unique information from each sensor. Being able to use multiple sensors is important for our project and is best suited for our project. However, since we are multiple sensors that are exactly the same means that we have sensors with the same addresses. To avoid this problem an I2C multiplexer is needed so that each sensor is communicating individually.

3.1.14.2 SPI

SPI is a form of communication protocol that uses MISO and MOSI to send and receive data. SPI works in a master-slave configuration where the master device controls the clock of all devices and initiates communication between all devices. SPI uses full-duplex communication which is able to send and receive

data at the same time, this allows for faster communication. SPI uses a daisy chain like configuration where the master communicates with one device at a time before it moves to the next, this allows for SPI devices to not need a unique address for each device. However unless a multiplexer is used, each device would need to have its own select line on the ESP32.

3.1.14.3 UART

UART is another form of communication protocol, it uses an RX and TX line to receive and transmit data. UART is great for a device that has to communicate with another device that is far away since there is no clock that is controlling the device, only receiving and transmitting data. The devices on each line must have the same baud rate and start/stop bits to have proper communication between devices. Each device has to have the previous configuration correct for each device or there will be no communication.

Table 22: Communication Protocol Selection Table

	I2C	SPI	UART
Synchronized Clock?	Yes	Yes	No
Amount of Devices?	Multiple	Multiple	One
Full/Half Duplex	Half Duplex	Full	Full
Signal Speed	Slow	Fast	Medium

4.0 Standards and Design Constraints

4.1 Standards

4.1.1 Health and Safety

This project will not deal with particularly hazardous materials processes, equipment, or techniques. However, one area of clear safety significance is wiring between components; this includes the connection between the AC to

DC converter, the connection between the AC to DC converter and the water pump, the connection between the AC to DC converter and the LED light strips, and any other connection that is passing current between wired components.

Proper insulation and secure connections are crucial to preventing electrical hazards such as short circuits, overheating, or accidental exposure to live wires. Additionally, strain relief mechanisms should be implemented to ensure that wires are not unintentionally pulled or disconnected during regular operation. To further enhance safety, fuses or circuit breakers can be incorporated to protect components from potential overcurrent conditions. While the system operates at relatively low DC voltages after conversion, the initial AC input still poses a potential risk, making careful handling and proper enclosure of electrical components essential for safe operation.

4.1.1.1 Health and Safety Standards

For this project, various different safety standards have been observed and followed. In particular, for the wiring of the water pump, *IEC 60335-2-51: stationary circulation pumps for heating and service water installations*, as well as *IEC 60335-2-55: electrical appliances for use with aquariums and garden ponds*, will be followed closely, particularly in regards to the wiring bending restrictions near the wall outlet. Additionally, all plug connections will be in accordance with the National Electrical Manufacturers Associations (NEMA) plug standards, in particular NEMA 5-15, in order to ensure the appropriately sized plug is used for any given current carrying application. Additionally, regarding the piping and tubing pressure ratings, the American Society for Testing and Materials (ASTM) standards listed in ASTM D1785 are closely followed for the ¼" schedule 40 PVC pipe in use for the water system.

4.1.1.2 Electrical & Safety Standards

In the development of the F.L.O.W.E.R.S. system, electrical and safety standards are fundamental to ensure reliable performance, user protection, and long-term system durability. One key standard is IEC 60529, which defines Ingress Protection (IP) ratings. This ensures that outdoor electronic components—such as sensors, controllers, and lighting fixtures—are resistant to water, dust, and environmental factors like insects or plant debris, all of which are common in garden environments. For safe and compliant electrical wiring, the NFPA 70 (National Electrical Code) provides detailed guidelines that help reduce risks of electrical shock, fire hazards, and other malfunctions. Since the system relies on battery-powered and possibly solar-powered components, ISO 50001 plays a pivotal role in managing and optimizing energy consumption. This standard

supports the design of an efficient energy management system that minimizes waste and promotes sustainability while keeping performance at its peak.

4.1.1.3 PCB Design Standards

Printed Circuit Board (PCB) design standards define the criteria necessary to ensure that a PCB can be successfully manufactured and operated as intended. These standards are often specified by the manufacturer. As mentioned above, for this project, the selected manufacturer is JLCPCB, and their specific requirements will guide the design process, as well as the specific standard used in regards to the PCB design; this will be unique to each company.

For example:

- Drilled holes in the PCB must range between 0.3 mm and 6.3 mm, with via holes having a minimum size of 0.3 mm and a 0.5 mm pad diameter.
- Vias must be positioned at least 0.254 mm from other vias and traces, regardless of net.
- The smallest allowable trace width is 0.127 mm, and traces must maintain 0.127 mm separation from one another.
- A minimum 0.3 mm clearance is required between any trace and the edge of the board.
- The outer copper layer thickness options for 2-layer boards are limited to either 1 oz or 2 oz
- Minimum spacing requirements include 0.5 mm between holes and pads of differing nets.
- Pads should have at least 0.127 mm clearance from pads on different nets, and 0.2 mm from traces.

4.1.1.3.1 – Overview of IPC-2221 Design Standards

The IPC-2221 standard serves as a foundational guideline for designing printed circuit boards that are durable, functional, and suitable for manufacturing. It covers a broad range of topics aimed at optimizing both the performance and reliability of PCBs. The following key areas are highlighted in the standard:

1. Thermal Management:

Effective heat dissipation is essential to prevent component failure and ensure long-term reliability. IPC-2221 provides recommendations on using thermal vias, selecting appropriate board materials, and placing heat-generating components to manage temperature distribution efficiently.

Design for Manufacturability:

IPC-2221 emphasizes simplifying the production process through design choices that enhance fabrication, assembly, and testing. Following DFM practices helps minimize errors, reduce costs, and improve yield during manufacturing.

2. **Trace Routing:**

The standard includes detailed guidance for trace layout, focusing on signal integrity, impedance control, and minimizing electromagnetic interference (EMI). It offers strategies for routing high-speed signals to reduce crosstalk and signal degradation.

Component Placement:

Proper component arrangement enhances electrical performance and thermal balance while simplifying assembly. The standard outlines how to avoid interference, improve access for testing, and enhance system reliability through strategic layout.

Material Selection:

IPC-2221 provides criteria for selecting board materials based on thermal conductivity, dielectric constant, and mechanical robustness. Material choice directly affects signal transmission quality and resistance to environmental stress.

3. **Design Guidelines:**

The standard defines essential parameters such as minimum trace width, clearance, and layer stack-up. These ensure that the board can handle power requirements, avoid shorts, and maintain overall functionality.

4. **Quality and Reliability:**

To ensure consistent performance, IPC-2221 includes recommendations for inspection techniques, testing protocols, and quality assurance practices that help detect defects and validate compliance with industry benchmarks.

5. **Environmental Considerations:**

IPC-2221 also encourages design strategies that account for environmental impact and operational context. This includes ensuring material compatibility with harsh conditions such as humidity, temperature extremes, and mechanical vibration. Designing for environmental resilience contributes to longer product life cycles and reduced failure rates in the field.

6. **Electromechanical Integration:**

The standard supports integration with mechanical components and housings by outlining mechanical constraints, such as mounting hole placement, board outline tolerances, and connector alignment. Proper

electromechanical design ensures the PCB fits seamlessly within enclosures and interacts correctly with other system elements, improving overall robustness and assembly compatibility.

By incorporating these principles, IPC-2221 serves as a robust framework for creating high-performance PCBs that meet the demanding requirements of modern electronic systems. This standard, combined with manufacturer-specific constraints, will guide our design to ensure quality, efficiency, and manufacturability. The following table will summarize these considerations within the standard:

Table 23: IPC-2221 Specification Category and Description Table

Standard Specification Categories	Description
Thermal Management	Recommends strategies for heat dissipation such as thermal vias and layout techniques to prevent overheating and ensure long-term reliability.
Design for Manufacturability	Promotes layout practices that simplify fabrication, assembly, and testing—helping to reduce errors and production costs
Trace Routing	Provides rules for signal integrity, EMI reduction, and impedance control—especially for high-speed signal paths.
Component Placement	Offers guidance for optimal part placement to improve signal flow, thermal performance, and accessibility for testing and assembly.
Material Selection	Advice on choosing materials based on electrical, thermal, and mechanical properties to match performance and environmental needs.
Design Guidelines	Defines core PCB parameters such as trace width, spacing, and layer setup to support power and signal requirements.

Quality and Reliability	Outlines inspection and testing practices to ensure that manufactured boards meet durability and performance expectations.
Environmental	Encourages designs resilient to factors like moisture, temperature swings, and mechanical stress for enhanced product longevity.
Electromechanical integration	Provides layout recommendations for board fit, mounting holes, and interface alignment to ensure seamless system-level integration.

4.1.1.4 Wireless Communication & Connectivity Standards

Effective operation of the F.L.O.W.E.R.S. system depends on seamless communication between components such as soil moisture sensors, microcontroller units (MCUs), actuators, and the user interface on a mobile application. Wireless communication standards ensure that this interaction is reliable and responsive. For short-range, energy-efficient communication—particularly between sensors and the MCU—Bluetooth Low Energy (BLE) is utilized. For high-speed, stable communication to and from the cloud or mobile app, IEEE 802.11 (Wi-Fi) is implemented, enabling real-time data monitoring and system control. In addition, the MQTT protocol, standardized under ISO/IEC 20922, enables lightweight and efficient data transmission between devices and the cloud, essential for IoT-based gardening systems. These standards ensure a communication framework that is not only fast and scalable but also robust enough to handle various smart gardening functions such as logging, automation, and real-time notifications.

4.1.1.4.1 Wireless Communication via the IEEE 802.11 Standard

For this project, wireless communication is a critical feature, as it allows seamless data transmission between the system and external devices. The IEEE 802.11 standard is fundamental in enabling this communication, providing a set of protocols that ensure interoperability between a variety of wireless devices. This standard lays the groundwork for devices to exchange data in a unified manner by defining essential aspects such as frequency bands, modulation techniques, and data transmission protocols.

One of the key elements of IEEE 802.11 is its specification of frequency bands used for communication. Two of the most commonly used bands are the

2.4GHz and 5GHz bands. The 2.4GHz band is particularly advantageous because it offers greater range and better penetration through obstacles like walls, making it ideal for home and indoor environments where devices may need to communicate over longer distances. On the other hand, the 5GHz band supports faster data transfer speeds, but its range is more limited and it struggles with obstacles more than the 2.4GHz band. The standardization of these frequency bands ensures that devices can easily detect and tune into the appropriate frequency, minimizing interference from other devices operating on different frequencies.

The IEEE 802.11 standard also incorporates advanced data transmission techniques, such as Orthogonal Frequency Division Multiplexing (OFDM), which divides the data into several smaller streams to maximize transfer speed. Additionally, Direct Sequence Spread Spectrum (DSSS) is employed to expand bandwidth and reduce interference. These methods improve the robustness and efficiency of wireless communication, which is essential for reliable data transmission in dynamic environments.

Within the IEEE 802.11 family, several versions provide different advantages depending on the needs of the application. The ESP32, which is the selected microcontroller for this project, supports 802.11b, 802.11g, and 802.11n. The 802.11b version operates on the 2.4GHz band and offers speeds up to 11 megabits per second. While sufficient for basic data transfer like video streaming, it is slower compared to other versions. The 802.11g standard also works on the 2.4GHz band but includes OFDM, achieving data rates up to 54 megabits per second. This improvement allows for better video quality and faster data transmission, offering a good balance of range and speed.

For higher-speed applications, the 802.11n version operates across both the 2.4GHz and 5GHz bands, providing flexibility in terms of coverage and speed. In this project, the ESP32 utilizes the 2.4GHz band of the 802.11n standard, capable of reaching speeds up to 150 megabits per second. This is more than sufficient for the needs of this project, as it ensures efficient communication between the moisture sensor boards, the main processing unit, and the connected mobile app.

Incorporating IEEE 802.11 ensures compatibility with a broad range of devices, offering seamless integration into existing networks. This adherence to a well-established standard guarantees efficient and reliable communication, making it an ideal choice for this project. Furthermore, it simplifies the task of ensuring that both data sent from the sensor devices and data received by the system are both understandable and compatible, facilitating smooth user interaction with the final product.

4.1.1.4.2 Wireless Communication via ESP-NOW

ESP-NOW is a proprietary wireless communication protocol developed by Espressif Systems, the company behind the popular ESP32 and ESP8266 microcontroller families. Designed to provide an efficient, low-power, peer-to-peer communication mechanism, ESP-NOW enables direct device-to-device data transmission without the need for a traditional Wi-Fi network or router. This protocol has become an integral part of Espressif's wireless stack, particularly for scenarios where low-latency communication and minimal infrastructure are required, and the plan for this project is to utilize this communication protocol for sending the moisture sensing data from a data collecting ESP32 to the central ESP32.

At its core, ESP-NOW operates over the 2.4 GHz ISM band, the same unlicensed frequency range used by Wi-Fi and Bluetooth technologies. However, unlike Wi-Fi, ESP-NOW does not rely on the IEEE 802.11 association process. Instead, it uses a simplified version of 802.11 frame transmission to exchange data between devices, allowing fast and lightweight communication. Devices using ESP-NOW can send and receive messages almost instantaneously without the overhead typically associated with Wi-Fi authentication and IP-based protocols. Specifically, ESP-NOW reduces the 5 layers of the OSI layer model to just the data link and physical layers, which makes it specifically faster than other transmission methods.

ESP-NOW is distinct in that it uses a broadcast and peer-to-peer communication model. Each device, referred to as a peer, is identified by its MAC address. Communication can occur between up to 20 peers on ESP32 devices. These peers do not need to be connected to a Wi-Fi network, although ESP-NOW can coexist with Wi-Fi on the same chip, allowing for hybrid use cases. The protocol is well-suited for applications that require multiple embedded systems to synchronize or exchange data without requiring internet access or a centralized server.

From a technical standpoint, ESP-NOW transmits data in encrypted or unencrypted forms, using the 802.11 Action frame subtype. The payload size is limited to 250 bytes per message, making it suitable for short bursts of data rather than high-bandwidth communication. Of course, for this project, moisture sensing data can easily be encapsulated in just a single byte, so 250 is more than enough to send the data, and size is not at all a concern in this case. For applications requiring low-latency and deterministic behavior such as sensor networks, device orchestration, and simple automation systems this packet format provides a practical trade-off between complexity and performance.

While ESP-NOW shares the same frequency band as Wi-Fi and Bluetooth, it differs significantly in its protocol stack and operational model. Compared to Bluetooth, ESP-NOW offers a longer range and faster transmission rates without requiring pairing or discovery procedures. This is particularly important given that bluetooth pairing takes time for the devices to handshake, which is minimized greatly by ESP-NOW. Unlike Zigbee, ESP-NOW does not rely on a mesh network topology and does not require a coordinator or gateway. These differences highlight its niche role as a lightweight, rapid-deployment protocol best suited for environments where point-to-point or small-group communication is needed without extensive network overhead.

A critical point of distinction for ESP-NOW is its proprietary status. It is not an open standard governed by a formal standards body such as the IEEE or Bluetooth SIG. Rather, it is fully developed, maintained, and supported by Espressif Systems. While this allows for deep integration with Espressif's own hardware platforms, it also means that ESP-NOW is not universally compatible with microcontrollers or chips from other manufacturers. The proprietary nature of ESP-NOW places it outside the realm of formal industry standards, even though it plays a significant role within the Espressif development ecosystem.

Despite its proprietary limitations, ESP-NOW complies with general regulatory requirements for devices operating in the 2.4 GHz ISM band. It operates within the constraints defined by regional authorities such as the FCC in the United States and ETSI in Europe. This ensures that devices using ESP-NOW can be deployed in most regions without specialized licensing, similar to Wi-Fi and Bluetooth devices. Nonetheless, its use is inherently tied to Espressif hardware, restricting its applicability in broader multi-vendor environments.

In conclusion, ESP-NOW represents a highly effective but proprietary wireless communication protocol designed to meet specific performance and deployment requirements. It fills a unique role in the wireless landscape by offering ultra-lightweight peer-to-peer communication with minimal latency and low power consumption. While not a formal standard, it reflects Espressif's strategic vision to provide robust communication tools optimized for embedded applications within its hardware platform. As such, ESP-NOW is best understood as a purpose-built communication protocol rather than a general-purpose industry standard.

Ultimately, this project did not end up using ESP-NOW. Initially, we believed that an ESP32 would be on each individual moisture sensor, and these values would be sent via ESP-NOW to the main board's ESP. However, due to technical limitations, this idea was not implemented, and instead, values were sent

directly to the main board using its ADC pins. Therefore, ESP-NOW was no longer needed.

4.1.1.4.3 Wireless Communication via HTTP format

HTTP (Hypertext Transfer Protocol) is a widely adopted application-layer protocol designed for distributed, collaborative, and hypermedia information systems. Originally developed as the foundation of data communication on the World Wide Web, HTTP has evolved into a general-purpose protocol capable of transmitting structured messages between devices across TCP/IP networks. It operates using a request-response model, where clients initiate communication by sending requests to servers, which then return the appropriate responses. In embedded systems and IoT applications, HTTP is commonly used to facilitate communication between microcontrollers and web-enabled devices, often over local Wi-Fi networks. In the context of this project, HTTP was utilized to send on/off control instructions from an ESP32 microcontroller to a Shelly RGBW2 device. The Shelly RGBW2, which is configured to operate in “lights” mode, accepts commands via HTTP GET requests sent directly to its IP address over the local network. For instance, to activate a lighting channel, the ESP32 sends a formatted request such as `http://<device_ip>/color/0?turn=on`, where `<device_ip>` corresponds to the assigned IP address of the Shelly unit. This lightweight use of HTTP allows for seamless integration of commercially available smart devices into custom automation systems. Unlike lower-level protocols such as ESP-NOW, HTTP relies on the full TCP/IP stack and requires the microcontroller to be connected to a Wi-Fi network. Although this introduces some overhead and latency, it provides the advantage of compatibility with a broad range of networked devices and services. HTTP’s human-readable format and support for RESTful communication make it especially suitable for debugging and integration with mobile applications or web dashboards. In this project, the use of HTTP simplified the process of remotely switching LED channels on and off, enabling reliable wireless control of the lighting system without the need for complex protocol handling or third-party cloud integration.

4.1.1.5 Data Security & Privacy Standards

As the smart gardening systems collect, transmit, and store sensitive data—such as environmental conditions, watering schedules, and user preferences—data security and privacy standards are critical. ISO/IEC 27001 provides a framework for managing information security, ensuring that collected data is encrypted, securely stored, and protected against unauthorized access. This is especially important when the system connects to mobile applications or cloud platforms. Additionally, the NIST Cybersecurity Framework for IoT offers practical guidance to enhance device-level security, covering aspects like

secure firmware updates, authentication protocols, and network protections. By implementing these standards, the F.L.O.W.E.R.S. system not only protects user data but also builds trust and resilience into the smart gardening experience.

4.1.1.6 Smart Lighting Component Standards

Smart lighting plays a key role in supporting plant growth and maintaining environmental balance, particularly in shaded areas or indoor gardens. To ensure both efficiency and safety, the system follows IEC 60598, which outlines the requirements for safe operation of outdoor and weather-resistant lighting fixtures. This ensures the lights can endure moisture, UV exposure, and varying temperatures. For optimized energy usage and real-time responsiveness, the system uses light sensors and controllers designed according to ISO 9241-303, which guides the performance of visual displays and ensures accurate brightness adjustment based on ambient conditions. Following these lighting standards promotes healthy plant development, reduces unnecessary energy use, and ensures that the system can operate safely and reliably over the long term.

4.1.1.7 Sustainability & Energy Efficiency Standards

Sustainability is at the heart of the F.L.O.W.E.R.S. system, and energy efficiency standards guide its eco-friendly operation. ISO 50001 supports the system's ability to monitor and reduce energy consumption across all major components—pumps, sensors, and lighting systems—ensuring that power is used efficiently. By integrating solar power or energy-efficient hardware, the system reduces dependency on external energy sources. Furthermore, irrigation efficiency is improved by aligning with EPA WaterSense guidelines, which encourage the use of water-saving technologies to minimize waste while maintaining healthy plant growth. These standards together promote a gardening solution that is both environmentally conscious and cost-effective, reducing resource consumption without compromising performance.

4.1.1.8 Sensor & Environmental Standards

Precision and accuracy in environmental monitoring are critical to the success of any smart gardening system. The F.L.O.W.E.R.S. system follows sensor and environmental standards to ensure reliable data collection from soil, water, air, and light conditions. ISO 18400 and ISO 11074 establish methodologies for soil quality testing and sampling, which guide the placement and calibration of moisture, pH, and nutrient sensors. For monitoring water usage, ISO 772 ensures consistent measurement of water flow rates and efficiency, supporting smart irrigation scheduling. Meanwhile, air quality components—such as

temperature and CO₂ sensors—help maintain a stable microclimate for optimal plant growth. Adhering to these standards allows the F.L.O.W.E.R.S. system to deliver accurate environmental data, which in turn powers smart automation and fosters a healthier garden ecosystem.

4.2 Constraints

4.2.1 Sustainability Constraints

4.2.1.1 Battery Life

Sustainability of powered devices is one of the most significant constraints within this project. As the goal of this project is hands off automation as soon as the user is given the project, it is very undesirable to need to replace power supplies like batteries in the design. For this reason, whenever feasible and practical in the design, the goal was to use 120V AC power from the wall to avoid using batteries. One case in which this came up was the water pump. For example, for any given 12V water pump, the device could easily be powered by wiring it to a 12V lead acid battery. Although this is a feasible method, the battery will eventually overtime be depleted of charge, and then there are multiple constraints that become a problem. First and foremost, the pump battery dying with no way to recharge inherently makes the project unsustainable. Further, a safety concern is then generated, because the user may try to connect a battery charger to the device, and if this connection is not fuse or breaker protected, could result in short circuit conditions, which almost always lead to property damage or worse. For these reasons, if at all possible, devices were chosen to be powered directly by 120V AC, or the AC to DC converter, which itself is powered by 120V AC from the wall

However, there are some hardware components for which batteries are not practical to avoid. One case is the mobile device by which the app is run; however, this is not a real concern, as mobile charging is standardized and incredibly safe in all regards. The most concerning batteries are on the Moisture Sensor PCB and the Main MCU. These batteries are more challenging to get to, and in order to make the design optimal for sustainability reasons, the devices are made to be as low power as possible to ensure a long system life. B cell "coin" batteries, which will be used for the moisture sensor PCBs follow IEC 61960 REV, which is a standard describing the battery life expected from such batteries given certain operating conditions. During testing for the design, the power consumption will be tracked as will the charge of the battery; this will allow for the testing of the battery to ensure it meets this standard.

4.2.2 Maximizing Battery Life

For the purposes of this project, the batteries involved will not see significantly high temperatures, but the coin batteries attached to the moisture sensor PCB will be subject to seeing water, as it will be sticking in the soil that will then have water sprayed on it. However, it was for this particular reason that this battery type was chosen. Coin batteries with a model name starting with "CR" use Lithium chemistry to develop their voltage, which is significantly better suited for wet conditions than Alkaline based batteries. By choosing Lithium based coin batteries, they will maintain a slick profile with minimal surface exposed to water, while still supplying sufficient power to run the moisture sensors

4.2.1 Maintenance and Reliability Constraint

4.2.1.1 Corrosion

When researching moisture sensors, we came across the problem that a lot of customers have complained about different types of sensors corroding such as direct contact electrode sensors. Most of the sensors that are built this way seem to be only used for a one time use or over a short period. Because of this we want to avoid corrosion as much as possible when dealing with the moisture sensor that we are going to use for our plants.

The two most common types of corrosion are electrolytic and galvanic. These two types of corrosion are very similar and for the purpose of this project it is important to understand both. Galvanic corrosion happens when two different metals are in contact with each other and an electrical current is supplied between the two. On a PCB this can happen between the metal of the component and the metal from the trace that comes in contact with each other [7]. Once moisture comes in contact with this area on the PCB the electrical current will cause the weaker/more resistive metal to corrode faster. This causes improper voltage readings between components as well as the components to degrade.

Some common reasons for corrosion are electrolysis and soil chemistry [1]. Electrolysis is due to two different types of metal coming into contact with water. The two different metals allow for electrolytes to connect to each metal and allow for a current to flow between the two metals. This allows for oxidation to occur in the metal which will then in turn cause the metal to slowly corrode away. Soil chemistry is another important part of corrosion because of how acidic the water is. If water in the soil is more acidic, then there are more H⁺ proton ions in it. This will cause more electrons to want to pair with the lone

hydrogen ions. The more electrons that are leaving from the sensor in the soil the faster the corrosion will occur.

4.2.1.1.1 Electrolytic Corrosion

Electrolytic corrosion happens when a conductive liquid, known as an electrolyte, comes into contact with an anode and a cathode. This allows for slivers of metal to cross over through the electrolyte and make contact with the other metal. Over a long period of time, if this is not treated properly, the metal will start to degrade.

In a PCB, if there is water that is present between two traces then a metal oxide will start to form between the two traces when there is an electrical current present. This can cause a short to occur in the PCB and over time this short can become permanent if the metal oxide layer forms. This short that occurs in the circuit can lead to undesirable results as well as total circuit failure. This is not desirable for any circuit and we will look to minimize this phenomenon as much as possible.

4.2.1.1.2 Galvanic Corrosion

Galvanic corrosion occurs when two unlike metals are in contact with each other while an electrical signal is being sent through them in the presence of an electrolyte, an electrolyte being water. During this process one metal is stronger, the cathode, and the other is weaker, the anode. Since the weaker metal is more resistive there is more current that is traveling through that metal. This effect will cause the anode to corrode faster and the cathode will not be affected at all.

On a PCB, galvanic corrosion can occur when water comes into contact between the trace and the soldering connection. This will create a conductive path across the two metals and the anode will consume the electrical current. This will cause the anode on the PCB to corrode much faster than the cathode. This happens because solder often contains tin, while the trace on the PCB is made out of copper. Because of these two dissimilar metals galvanic corrosion can occur. Over time these connections on the PCB, between the component and trace, will degrade enough where too weak of a signal is passed or no signal is passed at all and leads to component failure.

This type of corrosion is most commonly found in single-point resistive moisture sensors. This is because these sensors have been cheaply produced and use lower quality metals. Corrosion is reported with these sensors on many reviews and the deterioration leads to sensor failure.

Table 24: Electrolytic and Galvanic Corrosion

	Electrolytic Corrosion	Galvanic Corrosion
Cause	An electrical current passing through metal in an electrolyte	Contact between two different metals in an electrolyte
Environment	Stray electrical currents	Electrolyte is present in a potential difference between two metals
Inducing Factors	Conductive environment, strength of current, and type of metal	Type of metal and conductivity
Effect	Fast metal deterioration	Weaker metal will start to corrode
Prevention	Reduce stray current, coating, increase area between traces	Sacrificial anodes, protective coating, similar metals

4.2.1.1.3 Corrosion Prevention

There are many different types of solutions when trying to resolve corrosion. First is to use capacitive sensors rather than using resistive sensors. Capacitive sensors are sometimes referred to as corrosion resistant sensors, this is because the sensor doesn't require direct contact with the electrodes in the sensor and the water in the soil. By using a PCB soil moisture design we can choose the area of the PCB that we want to put into the soil to measure the change in the capacitance. How quickly or slowly the capacitor takes to fully charge again tells us how much water there is in the soil.

While a PCB is mostly made out of copper, a coating can be applied over the traces such as a gold or gold-nickel alloy to help prevent corrosion, however, this can prove to be costly. Epoxy, aerosol spray, and solder masks can also prevent corroding onto the PCB as well. Another way that we are able to work around any corrosion on the PCB sensor is to add a casing around the exposed components. This would prevent any stray water from getting onto the electrical components. Keeping all of the parts dry allows for no standing water to sit on any metal pieces causing them to corrode.

Keeping our parts dry can also be done by designing a 3D model of the casing for the PCB sensor and using a 3D printer to make it. By making a casing around the components we would be able to prevent water from getting directly onto the components of the sensor. This would greatly help us by increasing the longevity of the PCB.

4.2.1.2 Salinity

One of the major factors that can affect the readings of a volumetric sensor is salinity of the soil. In the context of our project, salinity refers to how much salt there is present in the soil. A study done by the National Library of Medicine researched how salinity affects output readings of different sensors [3]. Their research showed most of the moisture sensors that they observed overestimated the actual moisture level in the plant when there was too much salt in the soil.

This occurs because of the relationship between soil salinity and the electrical conductivity in the soil. From the study mentioned before by the National Library of Medicine, as the salinity of the soil increases so does the conductivity of the soil. The increase in salinity will allow for electrons to flow in a different behavior than what is expected. This effect mostly concerns sensors that are using capacitance based measurements. Under normal conditions these capacitance moisture sensors base their measurements on the dielectric constant of the soil.

With capacitor based moisture sensors the capacitor is being discharged and waiting to be charged up again. With an increase in salinity, the time that the capacitor is waiting to be charged again is now altered because the dielectric material that the voltage is traveling through is now a more conductive medium. This is resulting in improper readings because the capacitor is now charging faster. Calibrating the sensor at a high salinity is a work around for this issue, however you would need to keep the soil salinity level high which can be harsh for many plants.

To ensure that we have accurate measurements it is important for us to take into account the salinity level of the soil. If we are struggling to have accurate measurements in our project this can be one of the areas that we can troubleshoot. In order to do this we can change out the soil that the plant is in.

4.2.3 Environmental Constraints

One aspect of this project that could be easy to overlook is the environmental constraints. It may seem that the least concern of a project about growing plants indoors is an environmental concern, but this is not necessarily the case. For example, invasive plant species have historically caused billions of dollars of

damage, with the US Forest Service estimating that the annual cost of invasive species control in the United States was Five-hundred million dollars as early as 2000, and this cost has gone up each and every year. With this being said, the design is intended for the user to only cultivate and grow the plants native to the region in which they live, or non-native plants that do not pose a risk to native plant populations. However, even though the user is in control of this risk, it is still an important constraint to consider when building a device that could make this mal-practice easier. It is encouraged that all users follow the local wildlife and forestry guidelines in regards to which species they choose to grow in this automated indoor garden.

4.2.4 Security Constraints

Another important constraint to consider is that of security and vulnerability, especially as regards to the wireless communication between devices in this project; specifically the communication between both the phone and the main ESP32, but even the communication between the ESP32s on the moisture boards and the main ESP32. In any application where data is transferred and can be snooped upon, it is incumbent upon the design party to clearly lay out the vulnerabilities to the user, and attempt to remedy them as much as practically possible. For this project, the information being communicated is benign in every sense of the word; It is either moisture sensor values, or light sensor values, to information between the main MCU and the app, where certain values are returned for the user to view. Because there is no clear breach of data here that could cause significant property damage or information loss, no specific remedies have been put into effect. Of course, the Wifi communication will follow the IEEE 802.11 standard, which has built in safety mechanisms, such as Wifi protected access 3, which encrypts the sent data. However, despite the relatively benign nature of the data being transmitted, ensuring security remains important to safeguard against potential vulnerabilities in the system. The primary concern is the possibility of unauthorized access to the wireless network, which could allow malicious actors to disrupt communication between the devices or intercept sensitive data. One way to mitigate this is by implementing encryption protocols like WPA3 (Wi-Fi Protected Access 3) on all wireless communications. WPA3 ensures that any data transmitted over the Wi-Fi network is encrypted, making it much harder for attackers to decipher or hijack the communication. Additionally, using Secure Sockets Layer (SSL) or Transport Layer Security (TLS) for communication between the mobile app and the ESP32 can further protect the data being transferred by providing end-to-end encryption, ensuring that the information is encrypted before it leaves the phone and remains secure until it reaches the ESP32.

Another layer of protection comes from device authentication, ensuring that only authorized devices can communicate with the ESP32. In this case, implementing a device pairing process, where both the mobile app and the ESP32 exchange authentication tokens or keys before any data transfer begins, would add an additional level of security. The ESP32 itself should also be programmed to reject unauthorized access attempts and should employ security measures like firewall rules to prevent exposure to unnecessary open ports. Finally, regular software updates should be scheduled for the ESP32 and the mobile app to patch any discovered vulnerabilities. By adhering to these security standards and best practices, such as using WPA3, SSL/TLS, and device authentication, the project can reduce the risk of security breaches, even though the data being transmitted is not highly sensitive. This approach will ensure that the system remains secure and reliable for users.

5.0 Comparison of ChatGPT with Other AI

5.1 General Generative AI Assistance

Generative AI technology has been regarded as one of the most powerful tools in academic assistance lately, especially as the power it holds in its ability to contribute to collegiate education needs in real time is unparalleled through any other means of human assistance. Generative AI is remapping the way that students learn and problem solve and even potentially depleting students from the necessary skills needed to isolate complex problems and solve them independently. However, this all depends on the way in which AI is implemented in academic assistance.

In many cases, generative AI can show an extremely useful tool for creative needs where students may struggle (See 5.4: AI Assistance for F.L.O.W.E.R.S.).

However, in general, AI has proved to show extremely low yields in terms of useful assistance with hardware-dependent projects. While it is indisputable that AI may prove incredibly helpful with software development and debugging in general, it seems that, overall, the contributions which AI can contribute to a high level project such as full-stack projects or hardware and software connected projects is limited. The limitations of generative AI in hardware-dependent projects stem from the complexity and specificity required in such areas. Hardware projects often involve intricate design decisions, detailed material properties, and real-world constraints that generative AI is not well-equipped to address. For example, AI tools can assist with software coding by providing suggestions, debugging errors, and even generating code based on user input. However, the same level of support is not available when it comes

to designing circuits, selecting appropriate components, or understanding the physical properties that affect performance, such as resistance, capacitance, or thermal properties. In these cases, human expertise is still essential for making the nuanced decisions that ensure the success of a hardware project. Generative AI simply lacks the hands-on understanding needed to navigate the vast array of physical variables involved in hardware design.

5.2 Project Specific AI Generated Assistance

AI Generative assistance was almost exclusively used in the research phase of this design project. This included comparing hardware components and listing their positive and negative attributes, as well as listing general system constraints to plan the project around. The following cases are given that show both the prompt into ChatGPT, as well as the answers. This will be followed by an analysis of the answer in regards to its usefulness for the project

5.2.1 Example Prompt 1

The following prompt was used to consider the pros and cons between the different water pump technologies as relates to this particular application. The prompt read as follows: "Between Peristaltic, Diaphragm, and Centrifugal Pumps, what are the pros and cons for a water system that will be driven by a 12V water pump, and will not require precise water flow control?" ChatGPT responded with the following:

"For a 12V water pump application that does not require precise water flow control, each pump type—Peristaltic, Diaphragm, and Centrifugal—offers distinct advantages and tradeoffs. Here's a breakdown of the pros and cons of each type in the context of your water system:

1. Peristaltic Pump

Pros:

- *Self-priming: Can draw fluid without manual priming.*
- *No risk of backflow: Tubing collapses after each compression, acting like a built-in check valve.*
- *Gentle on fluid: Good if your system ever needs to handle sensitive fluids (like nutrient mixes).*
- *Dry-run capable: Can run dry without damage, unlike centrifugal pumps.*
- *Tubing isolates liquid: No contact between pump internals and fluid—less risk of contamination.*

Cons:

- *Lower flow rate: Typically less efficient than centrifugal or diaphragm pumps for high-volume movement.*
- *Tubing wear: Tubing wears out over time due to constant compression and needs periodic replacement.*
- *Pulsating flow: Output is not smooth—may not be ideal if a consistent flow rate is desired (even if not precise).*

Best for: Clean, low-volume applications where reliability and self-priming are more important than efficiency.

2. Diaphragm Pump

Pros:

- *Self-priming: Like peristaltic pumps, diaphragm pumps don't need manual priming.*
- *Handles higher pressure: Can create more pressure than centrifugal pumps—helpful for pushing water through filters or longer piping.*
- *Can handle particulates: Generally good at handling slightly dirty water, depending on the valve design.*
- *Good suction: Can lift water from a source that's lower than the pump itself.*

Cons:

- *Noisy: These pumps tend to be louder due to their mechanical operation.*
- *Pulsating flow: Like peristaltic pumps, flow is not smooth without a reservoir or accumulator.*
- *More complex design: Valves and diaphragms can wear and require occasional maintenance.*

Best for: Low to medium flow applications where pressure matters more than smoothness or silence.

3. Centrifugal Pump

Pros:

- *Smooth, continuous flow: Ideal if steady water delivery is important.*
- *High flow capacity: Good for moving a large volume of water efficiently.*
- *Quieter operation: Typically quieter than diaphragm pumps.*
- *Simple design: Fewer moving parts, generally low maintenance.*

Cons:

- *Not self-priming: Must be manually primed or placed below the water source.*
- *Can't run dry: Running without water can quickly damage the impeller.*
- *Lower pressure: Less capable of lifting water vertically or pushing through long piping.*

Best for: Simple, high-volume water transfer where the pump can be mounted below the water source.

Recommendation for Your Case-

Since your system:

- *Uses 12V (so energy efficiency matters),*
- *Doesn't require precise flow control,*
- *Likely benefits from reliability and simple operation,*

A diaphragm pump is likely the best balance:

- *It's self-priming,*
- *Handles a range of flows and moderate pressure,*
- *And works well in a 12V system with intermittent demand.*

However, if:

- *You want very low maintenance and continuous operation from a reservoir (and you can mount the pump below the water level), a centrifugal pump could also work."*

This answer provided a good range of considerations given the particular project details, but one potential downside of such an answer is that it does not consider that other types of pumps exist and may be better suited for the project; the generated answer only compares the technology types inquired about. Furthermore, it does not provide any sources by which it generated a response, and therefore its accuracy cannot be relied upon.

5.2.2 Example Prompt 2

The following prompt was used to consider the pros and cons between the different LED drivers that could have been selected for the project. The prompt

read as follows: "Our project requires the use of an LED driver to both supply power to, and change the brightness/color of 3 LED light strips. Could you compare the pros and cons between the Shelly RGBW2 and WLED drivers, given that we need to be able to program them ourselves (we do not want to use an already developed app)."

The reply from ChatGPT was as follows:

Shelly RGBW2

Overview:

The Shelly RGBW2 is a commercial LED controller capable of managing RGB or RGBW strips with a built-in WiFi module. It is designed for smart home applications and supports control via a proprietary app, cloud platform, or HTTP/MQTT protocols for integration.

Pros:

- **Compact and Ready-to-Use:** *The unit is enclosed and ready for deployment, with power terminals and LED strip connectors built in.*
- **MQTT and REST API Support:** *Shelly RGBW2 can be controlled through MQTT or HTTP-based APIs, allowing integration into custom backend systems or microcontroller logic without relying on Shelly's cloud.*
- **Power Monitoring:** *Built-in monitoring of output current and voltage can be useful for diagnostics.*
- **Wide Voltage Support:** *Compatible with 12V and 24V LED strips.*

Cons:

- **Proprietary Firmware:** *Although the device supports local control, it ships with firmware geared toward Shelly's app ecosystem. Custom firmware flashing is possible but voids warranty and may require advanced handling.*
- **Limited Flexibility in Animations:** *While you can control brightness and color, more complex lighting animations or effects are difficult to implement compared to platforms built for customization.*

- **API Latency:** HTTP control introduces some latency, making rapid or fine-grained brightness transitions less smooth.

WLED (on ESP8266/ESP32)

Overview:

WLED is an open-source firmware designed specifically for controlling addressable LED strips (e.g., WS2812B, SK6812) or RGB(W) analog strips via PWM. It runs on ESP8266 or ESP32 microcontrollers and supports a wide variety of custom lighting effects and integrations.

Pros:

- **Highly Customizable:** WLED is open-source and supports integration with REST API, UDP, MQTT, and custom serial protocols, making it ideal for complete in-house software control.
- **Preloaded Effects and Custom Effects:** Includes many built-in animations and the ability to add custom effects or integrate with custom logic via serial commands or web API.
- **Firmware Modifiable:** As a fully open-source solution, users can modify the firmware to fit their needs, including stripping down unused features for faster performance or more memory-efficient builds.
- **Community Support:** Extensive documentation and a strong user community support rapid development and troubleshooting.
- **Cost Effective:** ESP boards are inexpensive and widely available, making this a scalable solution for multiple light strips.

Cons:

- **Requires Assembly and Setup:** Unlike the plug-and-play Shelly module, WLED requires flashing to an ESP8266/ESP32 board, wiring of transistors or MOSFETs (for analog strips), and additional hardware configuration.
- **Hardware Add-ons May Be Required:** If using non-addressable RGB(W) strips, you'll need external MOSFET drivers to handle higher currents, as

the ESP outputs are logic-level only.

Recommendation for This Project

*Given the project requirements — **custom programmability, integration with other microcontroller-controlled components, and no reliance on third-party mobile apps** — the **WLED platform running on ESP32** is better suited for this application.*

The WLED platform provides full control over color, brightness, and effects through programmable interfaces, and allows integration with other ESP32 systems already being used in the project. While it requires more setup compared to the Shelly RGBW2, it offers unmatched flexibility and adaptability for embedded systems that need precise, programmable LED control.

If absolute ease of wiring and installation is more critical and limited dynamic control is acceptable, the Shelly RGBW2 could still be a fallback option. However, the customization limitations and reliance on prebuilt firmware make it less ideal for a fully custom hardware/software development project.

We decided not to go with this particular advice from ChatGPT, because we did not consider the limitations of the RGBW2 to be severe enough to outweigh its upsides. Specifically, ChatGPT seemed to consider the limitations of needing to use the Shelly firmware too strongly, but research showed that it was easily integratable with control from the ESP32 via MTTQ programming and therefore, ChatGPT's assessment was not enough to consider using a different project. However, it was still useful in generating some risk scenarios and made the team research further into the ideal driver for the project, meaning the AI input was still valuable for the project.

5.3 AI Platform Comparisons

For the following prompts, three different AI platforms have been used to generate responses for a few different questions related to our project. The following sections will serve to expand upon the differences in the responses between the platforms in regards to these questions.

5.3.1 AI Comparison Example 2

To have an attempt for AI assist in our project we will be comparing the outputs of AI generative services such as ChatGPT, DeepSeek, and Claude. This section will discuss the AI output of the previously discussed AI services in response to a prompt which is asking the AI to provide solutions for attempting to prevent corrosion onto our moisture sensor PCB. These AI services were fed the following prompt, “What is the best way to prevent corrosion for a PCB moisture sensor, which is placed into the plants soil, that is used in a diy garden project? On the PCB, the traces are used as a capacitive sensor to detect the change in moisture level within the soil”

5.5.3.1 ChatGPT

When asked about the initial prompt, ChatGPT had the following recommendations.

- Use a protective coating to apply over the copper traces such as a conformal coating.
- Use different materials such as gold plated pads and tin and nickel plating.
- Applying a solder mask to all traces and to seal off any exposed copper with solder.
- Elevate the PCB sensor so that it is not in direct contact with wet soil constantly.
- Offered an alternative of using stainless steel probes rather than a PCB based sensor
- Clean and recoat traces periodically

The first few responses were good recommendations from ChatGPT such as applying a coating, using corrosive resistant material such as gold, and applying a solder mask that could be used in this project to prevent corrosion. We most likely would not use gold since it can get expensive. ChatGPT also had responses that were not great too, such as not having direct contact with the wet soil and soldering any exposed copper. These responses can be misleading and could lead to poor design.

5.5.3.2 DeepSeek

When asked about the initial prompt, DeepSeek had the following recommendations.

- Applying a conformal coating.

- Use a solder mask and recommend an ENIG (Electroless Nickel Immersion Gold) finish instead of HASL (Hot Air Solder Leveling) for better corrosion resistance.
- Put epoxy around everything but the sensing area, said that this could affect sensitive and recommended testing before finalizing the PCB.
- Physical shielding (3D printed enclosure)
- Galvanic corrosion prevention, ensure the probes are made of the same metal.
- Use gold plate probes/pins instead of copper traces, explained to connect pins to PCB via wires to keep away from direct soil contact.
- Regular maintenance
- Avoid DC Biasing
- Recommended using a silicone conformal coating, ENIG finish, and shield the PCB. Also to test sensitive after each step has been implemented

From this experience using DeepSeek gave more and better responses than ChatGPT. It gave us some basic recommendations such as avoiding DC biasing and a conformal coating. But also went further by looking specifically at galvanic corrosion and the best way to avoid it. Something that DeepSeek did unlike ChatGPT was to give their recommendation of what to do for a DIY set up and in our opinion it had done a good job.

5.5.3.3 Claude

When asked about the initial prompt, Claude had the following recommendations.

- Conformal coating
- PCB material selection, FR-4 because it has a higher glass transition temperature rating which offers better moisture resistance.
- Solder mask to non sensing areas
- Using ENIG for traces that are on exposed to moisture sensing
- Encapsulate PCB while leaving moisture sensing traces exposed
- Design considerations: use wider traces, increase space between traces, design traces in an integrated pattern for better readings.
- Periodic cleaning
- Design alternative, cover traces with a thin layer of non-conductive material.

I would say that when given this prompt Claude performed better than ChatGPT by giving specific examples of what to use such as an ENIG finish. However, I would not say it performed better than DeepSeek because Claude had gone off

topic with how to improve readings, while DeepSeek dove further into different types of corrosion such as galvanic and how to specifically prevent it.

5.5.4 AI comparison Conclusion

In summary, while it's clear that each AI has its own set of positive and negative attributes, for the purposes of this project, ChatGPT will likely remain the most used. ChatGPT served particularly useful in generating responses in the research phase, especially in regards to system constraints and particular hardware pros and cons, which helped significantly in the design process

6.0 Hardware Design

6.1 MCU PCB Schematic

6.1.1 ESP32

The ESP32 will be powered by a 3.3V regulator. A HiLetgo CP2102 USB to TTL UART dev board will be used to receive USB communication. The USB to UART converter will be connected to VCC, TX, RX, and GND pins on the ESP32. The RX and TX pins on the esp32 are 34 and 35. In our ESP32 we will be using I2C protocol with the SCL and SDA lines being pins 14 and 16 respectively. When designing the board layout test points will be placed in various places to verify and ensure that there is proper connection in the board.

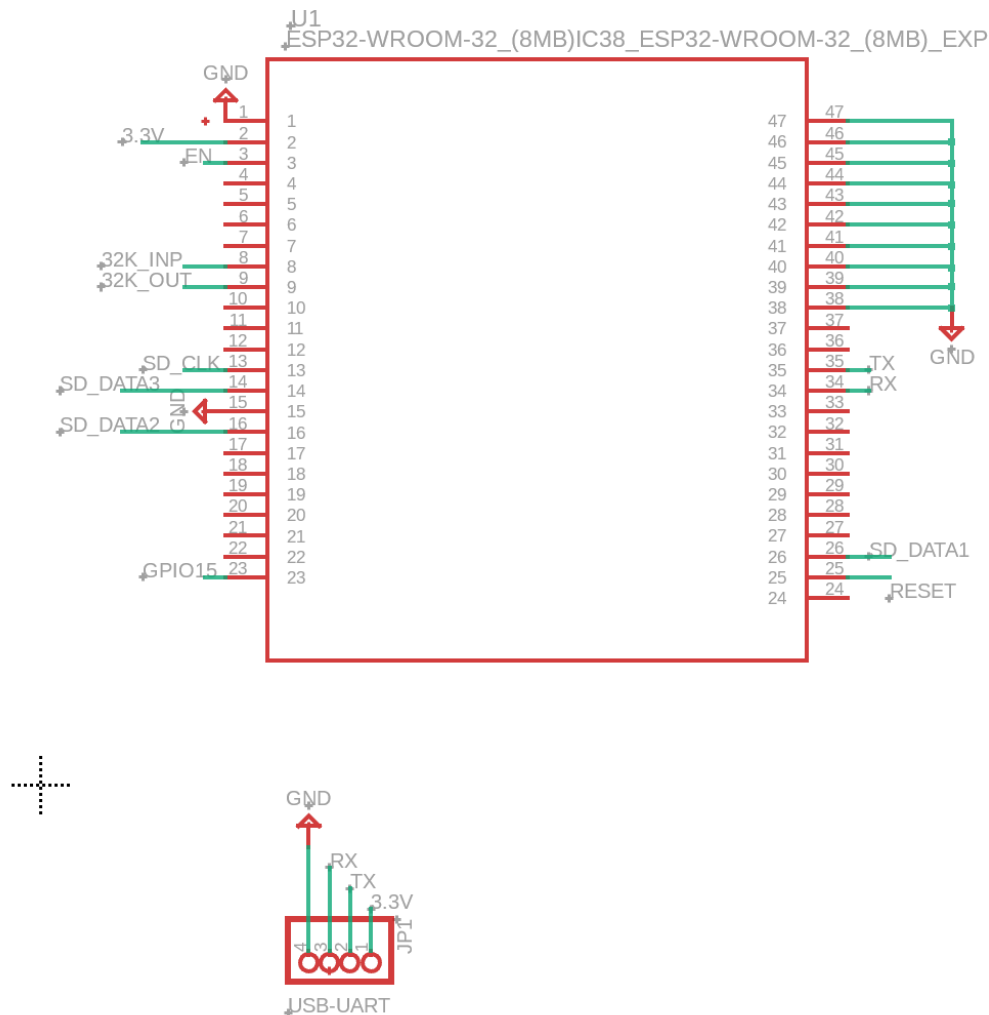


Figure 7: MCU and USB-UART Schematic

6.1.2 Regulators

On our board we will have two different regulators, 3.3V and 5V regulators. To power our regulators we will be using a PCIFR18650-1500 battery which has a nominal voltage supply of 3.2V with a capacity of 1.5Ah. The 3.3V regulator that we will be using is a TPS613221ADBVT. This 3.3V regulator operates with an input voltage from 0.9V to 5.5V and has a typical peak switch current limit of 1.2A. As seen in the datasheet, the recommended operating range for the inductor of the circuit is from 0.7 μ H to 13 μ H. From the schematic below, an inductance of 4.7 μ H has been selected. When selecting a capacitor it was recommended in the datasheet to use an output capacitor within the range of 4.7 μ F to 100 μ F. We chose a 22 μ F output capacitor as well as a bypass capacitor

from the battery. The 5V regulator that we have selected is a TPS613222ADBVT. The 5V will have a lot of the same operating features as the 3.3V regulator such as the 18650 battery, inductor values and capacitor values. Included in our regulator design we added a switch so that we can power on and off our board as we need. Our 3.3V regulator will be used to power our esp32 and other sensors and components on our board. The 5V regulator was being used in a previous version of our design but is not currently being used in the most recent version but will still be included in the schematic if a 5V supply is needed.

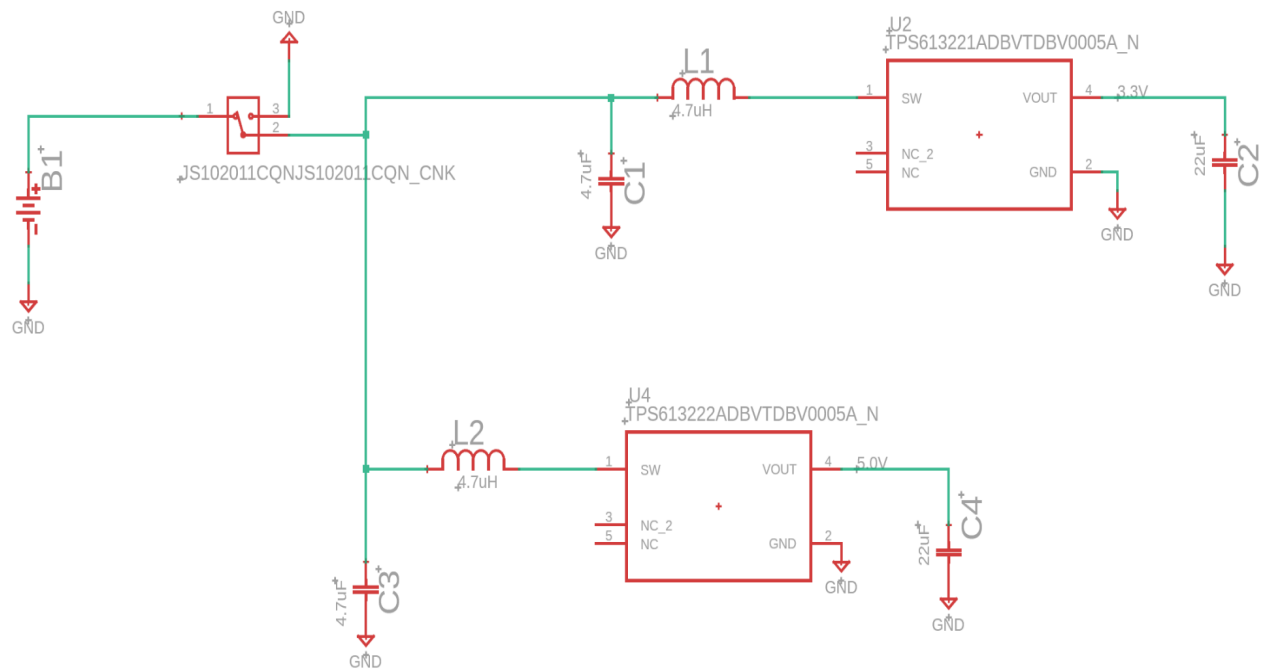


Figure 8: Regulator Schematic

6.1.3 I2C Multiplexer

An I2C multiplexer will be used to communicate with the sensors in our project. We will be implementing three AHT20 humidity/temperature sensors and three VEML7700 light intensity sensors. When using I2C communication protocol we need to be able to communicate with each sensor. This is done by the address that is configured on the pullup resistor of the sensor. However, since we are using multiple of the same sensor that have the same address the I2C protocol will be confused and will not work properly. There are different solutions to this problem and for our project we will be using an I2C multiplexer which will allow communication with each sensor individually without interference.

The SDA and SCL line for the multiplexer will be connected to the SDA and SCL lines on the MCU which are pins 13 and 14 respectively. Connected in parallel to these lines are 1k Ohm resistors per the datasheet design which is connected to 3.3V on the other end of the resistor. Each of the sensors are then connected to the SDA and SCL output pins of the multiplexer.

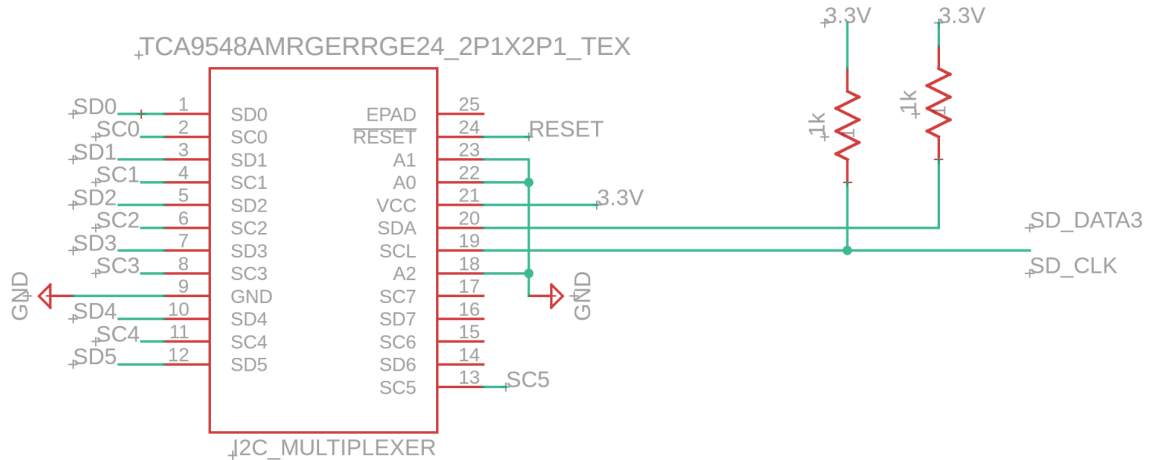


Figure 9: I2C Multiplexer

6.1.4 Sensors

In our project we are using two different types of sensors. One is a humidity/temperature sensor and another is a light intensity sensor. Both of these will take readings in each of the plants respective areas and will be sent to a mobile device for the user to monitor. There will be three AHT20 (humidity/temperature) sensors and three VEML7700 sensors. Each of these sensors use I2C communication protocol in order to send data back to the ESP32. These sensors will be supplied with a VCC from the 3.3V regulator. Each SDA and SCL line for the sensor will be wired to the output SDA and SCL line of the I2C multiplexer. The SDA and SCL lines of the sensor have a 1k resistor in parallel which is connected to 3.3V on the other side of the resistor, this design feature follows the recommendation of the datasheet for the I2C multiplexer.

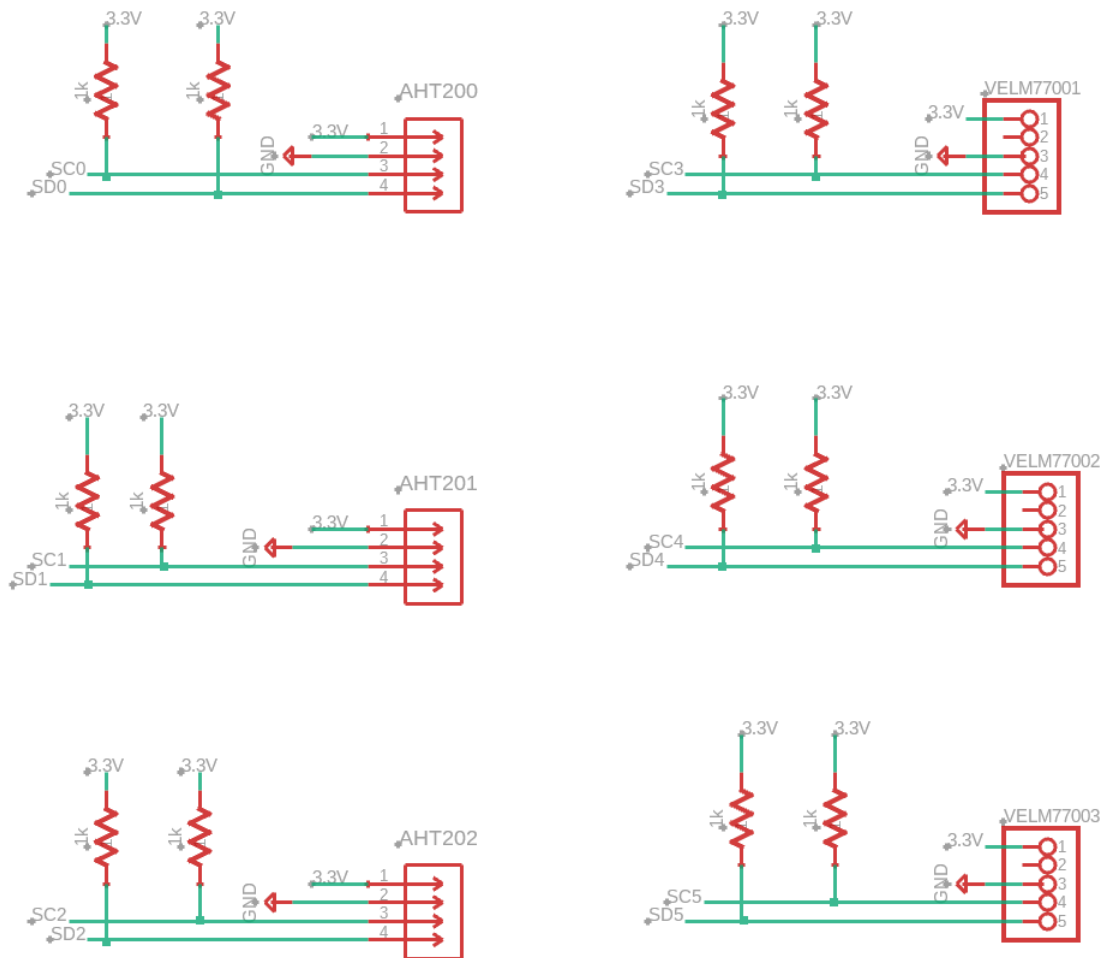


Figure 10: Sensor Schematic

6.2 Water Pump Subsystem Block Diagram

The following block diagram shows the water delivery system in its individual components. This includes both the connections between every component and its power supply, and additionally shows these connections for all 3 plants environments being maintained.

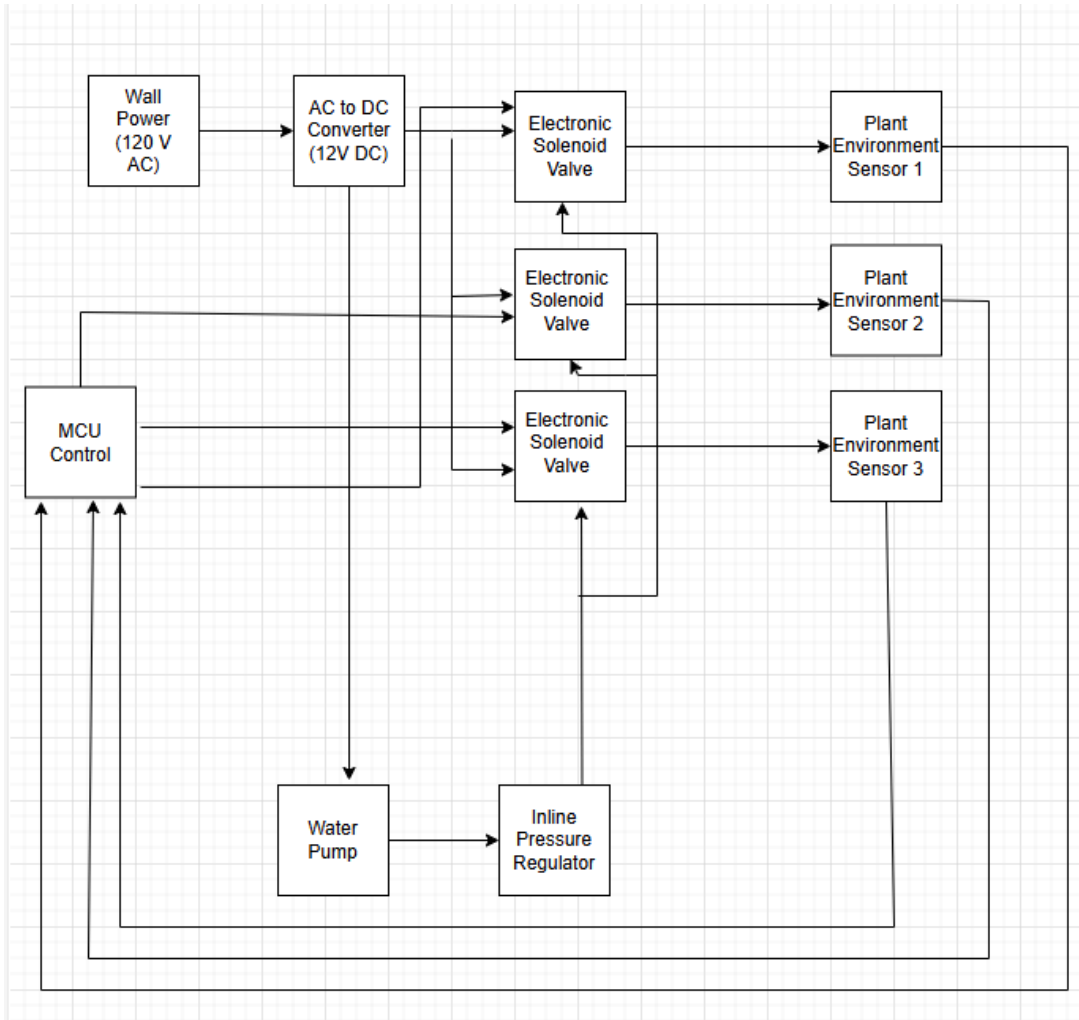


Figure 11: Water Subsystem Block Diagram

With the specific parts selected, the system can be explained component by component. Let's first look at the power supply for every component. The AC to DC converter supplies both the water pump and the solenoid valves for their power needs, as both the three solenoid valves and the pump operate on a 12V supply. First, the water pump will drive the entire PVC pipe system up to 50 PSI, at which point the pressure switch, which is internal to the pump assembly, will turn the water pump off. This water will be drawn from the plastic storage tank located on the shelf that holds the plant pots. From this point forward, no water will flow from the system, as the electronic solenoid valves operate in the normally closed configuration. However, eventually, the MCU will make a reading from the moisture sensor, and when that reading comes back such that the MCU determines that water is needed at a given plant pot, it will output a voltage on one of its pins that will actuate the associated solenoid valve. From

here, water will flow through the valve, and reaches the drip line that runs into the plant pot. The water pump will continue to pump water until the MCU gets a reading that the soil moisture has surpassed a given threshold, at which point it will stop outputting on the GPIO pin that was powering the solenoid valve. Because the solenoid valves are normally closed, this will result in the valves closing, which will cut the water flow to the plants. For an additional few seconds, the pump will actually continue to run, as the running condition for the pump is unrelated to whether or not a valve is open at a given time; it is only dependent on the pressure switch in the pump assembly, which will take some time to bring the system back up to 50 PSI.

Something worth noting about this hardware design, both for this subsystem and for the lighting subsystem, is the heavy reliance on the AC to DC converter. At any given moment, this power supply may be needed to supply the power for the pump, the solenoid valves, and the LED light strips, meaning that the converter must be appropriately sized for this application. As is standard practice, it is assumed that the power supply must be able to handle the "worst case" condition by at least a 10% margin; this means adding the maximum power consumption of all three valves, the water pump, and the led light strips, and the Shelly RGBW2 light driver, and then adding an additional 10 percent to this number. The converter chosen does meet this value, meaning it is appropriately sized for the application.

The water system as a subsystem will not include any specific PCB design, as the moisture sensors will serve as their own subsystems entirely, and therefore, the design will be included in the section below about the moisture sensor subsystem. This also applies to the MCU, which will serve as its own hardware subsystem design, and therefore, the PCB design for the main MCU will be included in that section. The following image shows a photo of the 12V direct wired solenoid valve

6.3 Moisture Sensor

At the core of the system's monitoring mechanism are three soil moisture sensors, which are directly connected to the ESP32. These sensors continuously measure the soil's moisture content, providing essential data for automated irrigation. However, since soil moisture sensors naturally generate an analog signal, and the ESP32 primarily processes digital data, an Analog-to-Digital Converter (ADC) is required. The ADC allows the ESP32 to interpret the raw sensor readings by converting the varying voltage levels into a digital value that can be processed. Once the data is received and analyzed, the

ESP32 determines whether the soil is too dry and, if necessary, activates the water pump and solenoid valve to initiate watering. As soon as the desired moisture level is achieved, the system automatically shuts off the water supply to prevent overwatering.

Once activated, the water pump draws water from a reservoir and delivers it to the plants. Simultaneously, the electronic solenoid valve regulates the flow, ensuring the appropriate amount of water is distributed. The integration of an LCD display enhances the system by providing real-time feedback on soil moisture levels, system status, and water usage, allowing users to monitor and track the irrigation process efficiently.

This automated system optimizes water conservation and plant health, making it an efficient and intelligent solution for smart gardening. Additionally, wireless communication capabilities allow users to remotely monitor and control the system. The ESP32's built-in Wi-Fi module enables real-time data transmission to a mobile app or web interface, where users can view sensor readings, adjust watering schedules, and receive alerts. This connectivity ensures that even users with busy schedules can manage their plants effectively. By combining automated sensing, real-time adjustments, and remote access, this smart garden system provides a comprehensive, low-maintenance solution for plant care.

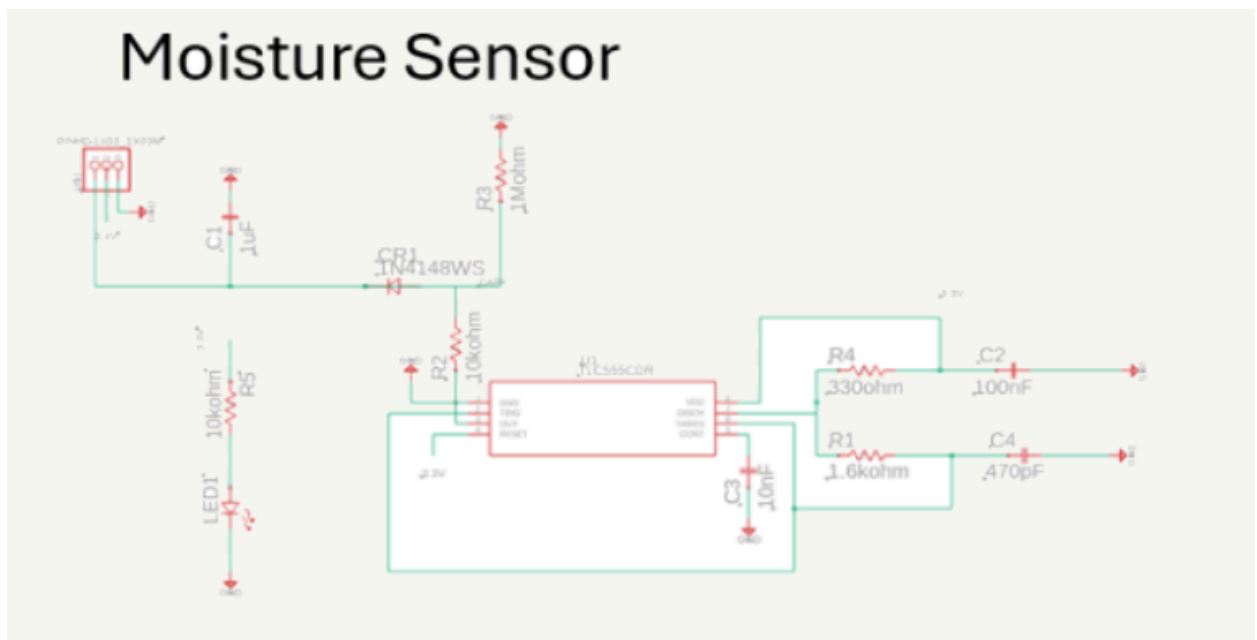


Figure 12: Moisture sensor board schematic

The above schematic is not our design but is used as a reference from an external source. Although not our design we can still analyze the components and their connections from its functionality and working aspects presented to us. In the schematic, it shows that there is a 555 timer pulse generation system used for the pulse-width modulation (PWM).

6.3.1.1 Voltage Regulation

On the right side of the schematic we can see that it demonstrates that it is a voltage regulator that will give us a constant voltage of 3V. This is super important because the timing part of the circuit, and anything else plugged in, really relies on having a dependable source of power to work properly. Imagine trying to keep time with a clock that speeds up and slows down – that's what could happen without this stable voltage!

The real workhorse here is this little chip labeled IC1, which is a fancy type of voltage regulator called an LDO. These LDOs are great because they're efficient, especially when you're dealing with lower power stuff. They're good at taking a slightly higher voltage – maybe 5 volts or even more – and calmly bringing it down to that exact 3.0 volts we need, without getting too hot and bothered.

Then you've got these little guys, the capacitors (C1, C2, C6, and C8). Think of them as tiny sponges that soak up any electrical noise or little bumps in the voltage. They help smooth things out even further, so the circuit gets really clean power. And of course, everything needs a solid 'home base,' which is what the ground (GND) connections provide. It's the reference point that everything else works from.

So, by taking that potentially jumpy input voltage and turning it into a reliable 3.0V stream, this part of the circuit is basically guaranteeing that the 555 timer and everything else connected gets exactly what it needs. This stable power is key for avoiding any weird behavior or inconsistencies, especially when you need things to be precisely timed and controlled."

6.3.1.2 555 Timer Pulse Generation

On the left side of the circuit diagram where creating pulses happens, thanks to a really handy chip called the 555 timer. Think of this chip as a tiny electronic heart that can beat out precise rhythms or generate single 'blips' of electricity whenever needed. It's a super versatile component used for all sorts of timing tricks in electronics.

At the heart of this system is the TLC555CDR (IC2), a low-power CMOS variant of the traditional 555 timer. This component generates precise timing signals based on the surrounding resistors and capacitors. Key elements influencing its operation include R3 (330Ω), R4 (1.6KΩ), and capacitors C4 (100nF), C5 (470pF), and C7 (10nF), which collectively determine the pulse duration and frequency. Additionally, diode D1 (1N4148WS) plays a crucial role in discharge control, ensuring predictable capacitor discharge cycles and maintaining accurate pulse timing.

This balanced combination of power, stability, and signal precision makes the circuit particularly valuable for applications requiring reliability in variable conditions—from automated irrigation systems to sensor-based environmental monitoring platforms. The design serves as both a practical solution for immediate implementation and an educational reference for understanding fundamental electronic subsystems.

With minor modifications to component values or integration with additional sensors and output stages, this versatile circuit can be adapted to address specific requirements across the spectrum of electronic control and automation challenges

6.4 Lighting Subsystem Block Diagram

The following block diagram shows the lighting system in its individual components. This includes both the connections between every component and its power supply, and additionally shows these connections for all 3 plants environments being maintained:

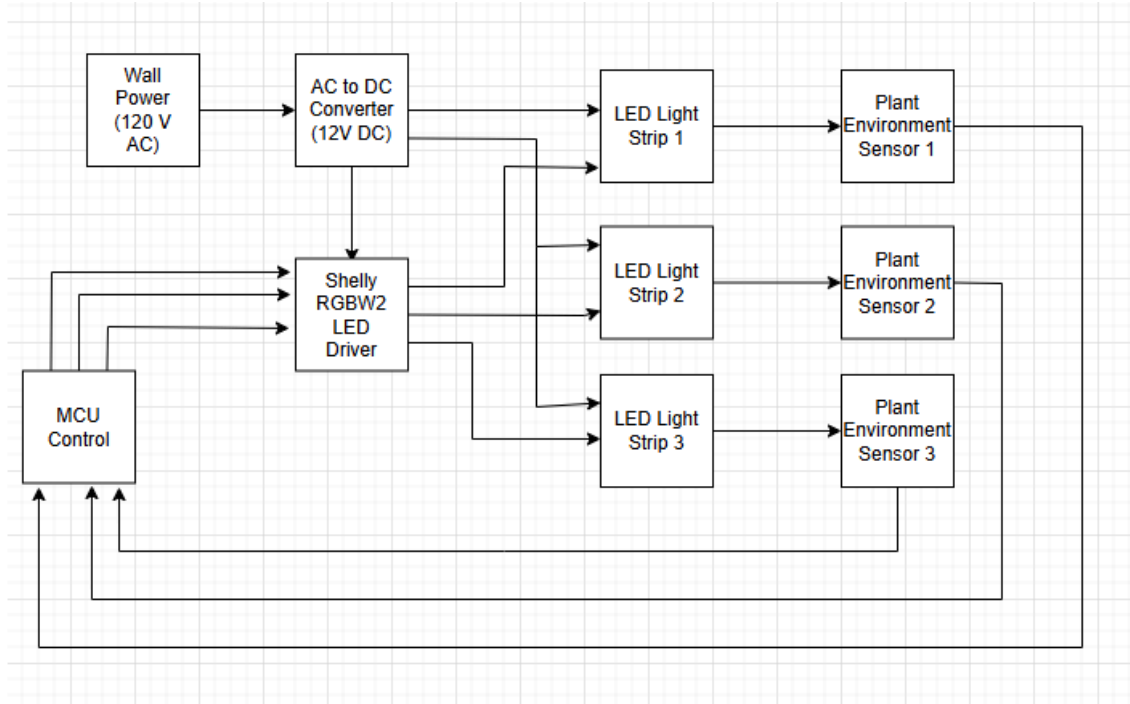


Figure 13: Lighting Subsystem Block Diagram

With the specific parts selected, the system can be explained component by component. Let's first look at the power supply for every component. As with the water delivery system, the AC to DC converter will supply all of the powered units in this module, with the exception of the MCU, which is powered on the board on which it lives. As mentioned above, the Shelly RGBW2 is the magic that allows this system to work correctly. As mentioned in the project description, a fundamental aspect of this design involves both dimming and controlling the frequency of the light present in the plant environment based on the user input from the app's user interface. The Shelly gets powered via the AC to DC converter; in fact, it can run on both a 12V and 24V supply, but in this case, the application calls for the usage of the 12V system. The Shelly will connect via wifi with the ESP32, and will dim/fade each of the lighting circuits based on the control signal from the MCU. For light strips with red, blue, and green colors, this can also be controlled remotely using the Shelly. From here, the light intensity will be measured via the light sensors present in the environment. These sensors will then report a value back to the ESP32, and based on the plant present in the environment, the ESP32 will either brighten or dim the light value to best meet that individual species need.

As mentioned in the water delivery section, special attention must be paid to the power requirements of the AC to DC converter, as it will be driving many devices simultaneously. Additionally, as with the water subsystem, the lighting

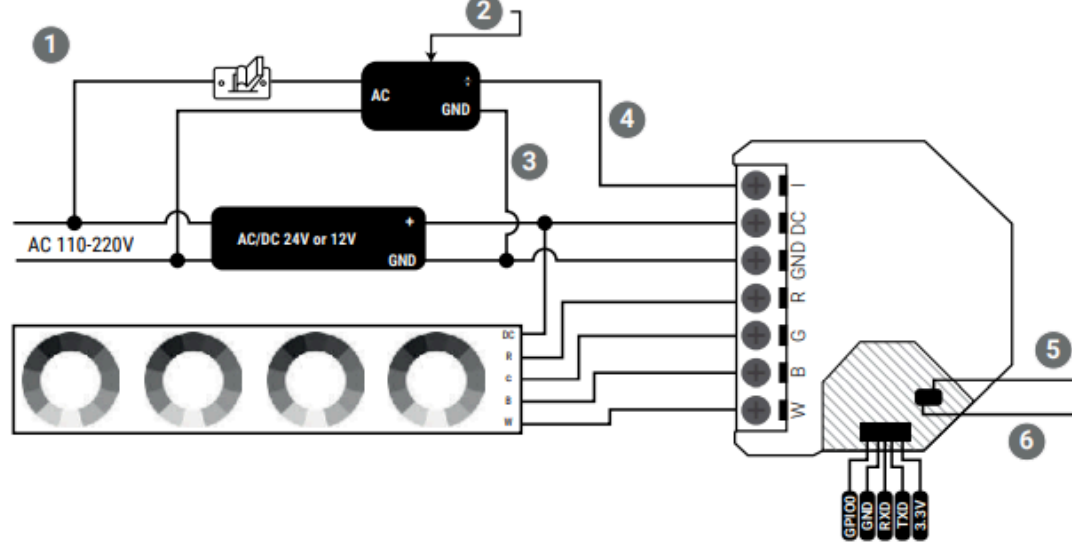
subsystem does not include its own specific PCB designs; the MCU PCB design will be included in its own subsystem section, and the light sensors themselves are purchased as a package and directly wired to the MCU, thereby avoiding the necessity of a designated PCB.

6.5 Shelly RGBW2 Driver Schematics

As has been discussed at length, the Shelly RGBW2 is a critical component for this project design, both for its functionality as a wireless on/off switch for the solenoid valves, as well as power the solenoid valves, as well as for its ability to wirelessly turn on and off the LED light strips while also providing the ability to dim the lights and change the colors to output the ideal growing frequency of a given plant. This wiring does not require a PCB in this project, although Shelly provides a wiring schematic via its website to both of these wiring schemes; one in which RGB LED lights are being controlled, and one in which the Shelly is simply acting as a power source that can be controlled wirelessly and simply functions as an on/off toggle. The second image shows lights being used, but given that this is just a toggle, we can imagine that instead, the 12V DC solenoid valves will occupy these positions. (there will only be three valves and not four, but this is not relevant in terms of how the actual wiring is done; there will simply be one less device controlled by the driver than the schematic shows) Those two schematics are shown below, respectively:

- EN RGBW LED strip scheme
 DE RGBW-LED-Streifenschema
 IT Schema di strisce LED RGBW

- ES Diagrama de la tira LED RGBW
 PT Esquema da tira de LEDs RGBW
 FR Schéma d'une bande de LED RGBW



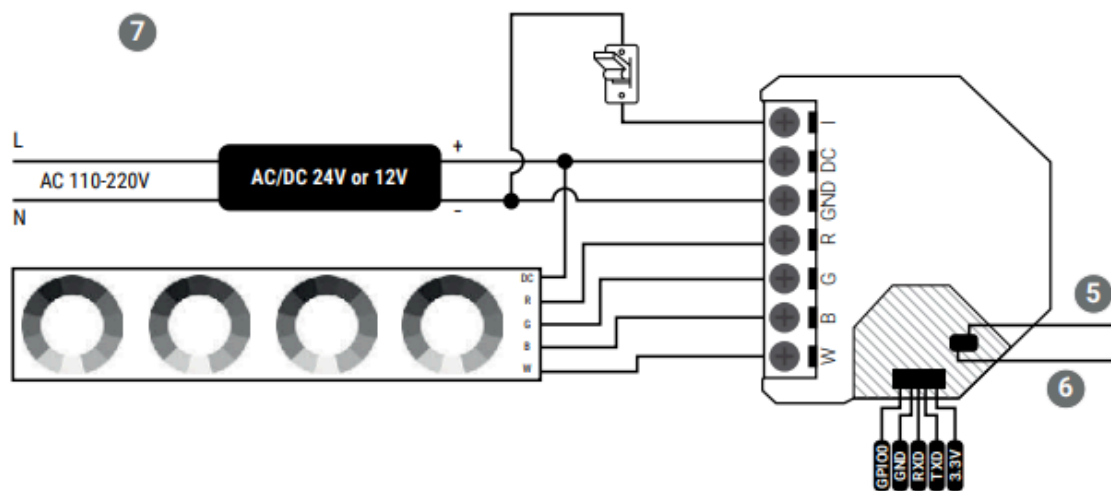
- 3** EN black cable
 DE schwarzes Kabel
 IT cavo nero
 ES cable negro
 PT cabo preto
 FR câble noir

- 4** EN red cable
 DE rotes Kabel
 IT cavo rosso
 ES cable rojo
 PT cabo vermelho
 FR câble rouge

- 5** EN LED
 DE LED
 IT GUIDATO
 ES LED
 PT LED
 FR LED

- 6** EN Reset button
 DE Reset-Knopf
 IT Pulsante di reset
 ES Botón de reinicio
 PT Botão de reiniciar
 FR Bouton de réinitialisation

Figure 14: Shelly Driver Schematic 1



- 7
- EN DC (12-24V) on/off/dimming switch
 - DE DC (12-24V) Ein / Aus / Dimmschalter
 - IT Interruttore di accensione / spegnimento / regolazione della luminosità CC (12-24 V)
 - ES Interruptor de encendido/apagado/dimmer de CC (12-24V)
 - PT Interruptor CC (12/24V) ligado/desligado/intensidade
 - FR Interrupteur marche/arrêt/gradation CC (12-24V)

Figure 15: Shelly Driver Schematic 2

6.6 Overall Schematic Design

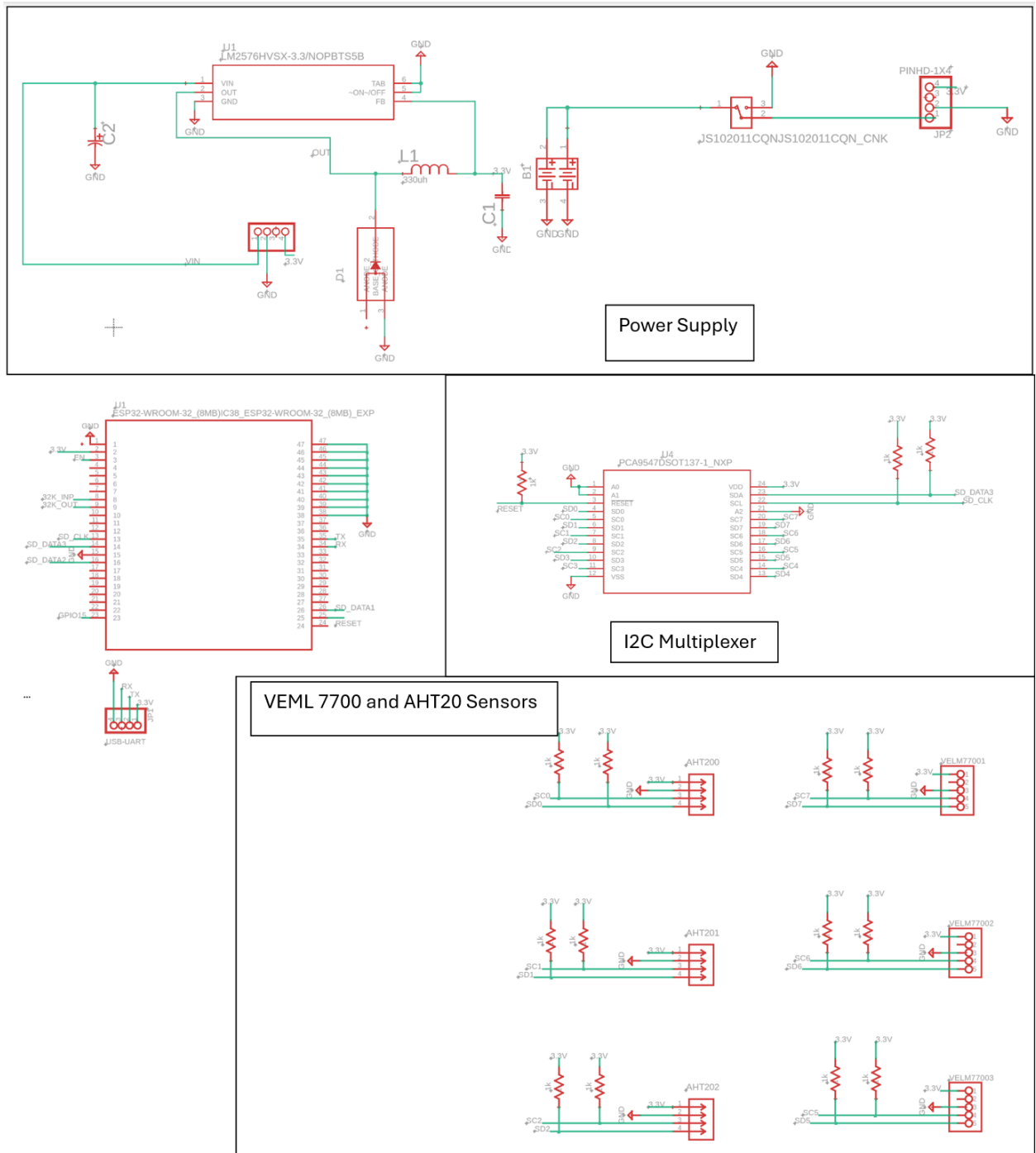


Figure 16: Overall PCB Schematic

7.0 Software Design

7.1 Embedded Design

The embedded design will encompass all of the software servicing the main components on the physical system. There will be two implementations of this relating to (1) the soil moisture sensors and (2) the main CPU which will handle driving the entire system operation such as taking in and outputting information in regards to the mobile app, comparing the actual and desired soil moisture levels, engaging the watering and lighting features, etc. For example, consider the operation of the soil moisture sensors; they will require embedded firmware that reads and processes analog data, converts it to a digital format, and communicates the findings to the main CPU. This will involve periodic polling of the sensor data, filtering noise, and ensuring accurate readings before making irrigation decisions. The main CPU, also an ESP32, will run a real-time control loop that continuously evaluates sensor data and triggers appropriate responses. Additionally, the embedded software will include communication protocols such as Wi-Fi or Bluetooth to relay system status to the mobile app, allowing users to monitor and adjust settings remotely. Efficient power management and task scheduling will also be incorporated to ensure smooth and responsive system performance.

7.1.1 Technologies

To develop the embedded software for the ESP32, we used the Arduino IDE and relied on developing our software in the embedded C++ language. Due to Arduino familiarity with $\frac{3}{4}$ of our group members, this seemed the best choice for implementation as opposed to the VSCode's PlatformIO or ESP-IDF environments for development. As a whole, Arduino resources are vast and abundant, which is another reason we found this development environment to be attractive.

While developing the embedded software existing now, it was understood that there would be multiple group members working with their own soil moisture sensors and ESP32s simultaneously, meaning that there would be concurrent changes to the program. In order to keep all parties up to date on the changes, we used GitHub to keep track of edits, ensuring frequent updates and communication about changes in a strict time-sensitive approach via Discord.

7.1.2 Soil Moisture Sensors

Our design will implement at least 3 soil moisture sensors, each composed of the same design and logic.

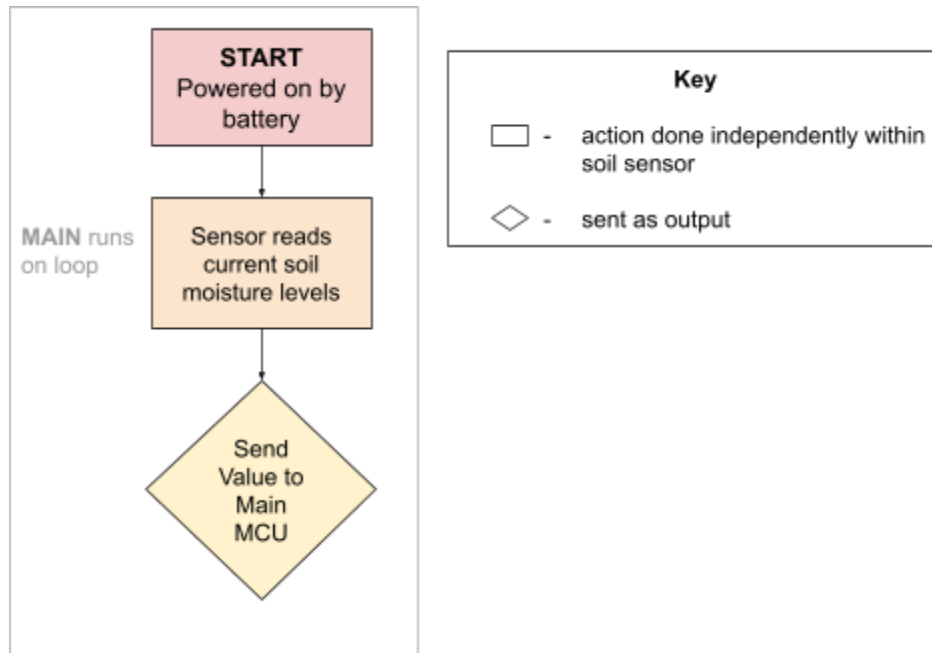


Figure 17: Soil Moisture Embedded Software Flowchart

The soil moisture sensors will be extremely simple in operation. They will serve the sole purpose of measuring the soil moisture levels. Since the computation and comparison of current soil moisture versus desired soil moisture will require communication with the mobile app to gather the user's input, no calculation will take place within this embedded code. Instead, as Figure 7 shows, the sensors will gather the moisture value and immediately send it to the Main MCU in order to be used for comparison.

7.1.3 Main System MCU Design

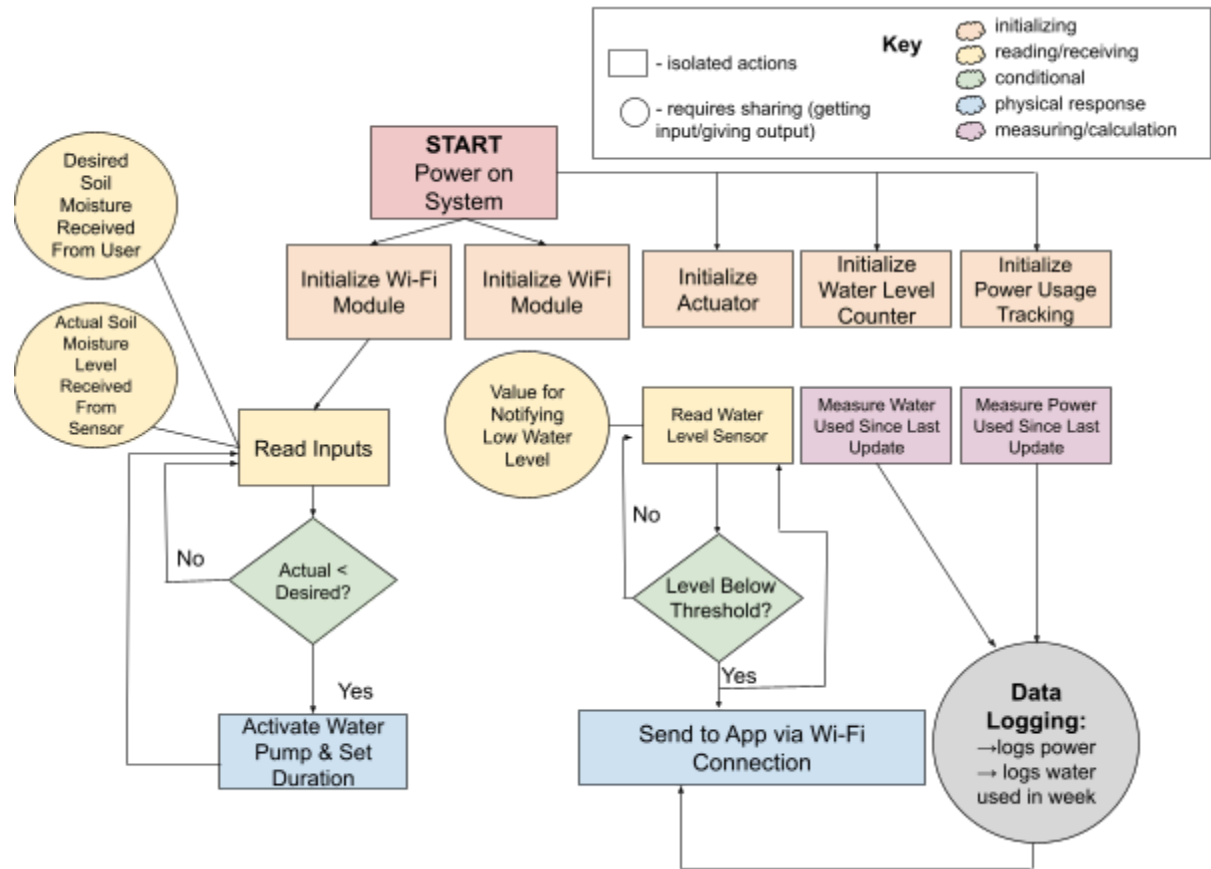


Figure 18: Main System MCU Software Flowchart

The main system's MCU design will handle all of the driving characteristics of the project. For example, the ESP32 driving the center of the physical plant environments will be in charge of handling connections to the mobile environment, connections to the soil sensors, and driving the water pump. The soil sensors and mobile connections are interfaced externally to the main module while the actuator will be internally connected.

The ESP32 will be doubly linked to the soil moisture sensors and the mobile application via Wi-Fi. Connection to the soil sensors will be a one-way receiving tie in which the soil moisture sensor uses the Wi-Fi capabilities within its chip in order to send the soil readings to the ESP32. Conversely, the connection to the mobile app will be a reciprocated link as the app and ESP32 both send to and receive from one another. The mobile app, via the user's input, will be responsible for communicating all of the data which draws a boundary for the ESP's 32 values (for example, the desired soil moisture/lighting features and the value at which the user would like to be notified of low water level). On the other

hand, the ESP32 unit will be in charge of tracking and sending data to the user such as overall metrics like the power consumption and water use as well as information related to each individual plant environment (lighting, soil moisture), so that the user can follow along with these stats in real time.

The main MCU will be in charge of calculations and comparisons as well. The main way in which this will be expected is the conversion of the received soil sensor values to soil moisture percentages. Through rigorous testing, we will determine what the method or formula is for this conversion. This conversion will be crucial for comparing the user's input to the current soil readings to determine the conditional "Actual < Desired?". This will be the factor which determines whether or not the water pump will send more water to the environment, so it is critical that this works properly

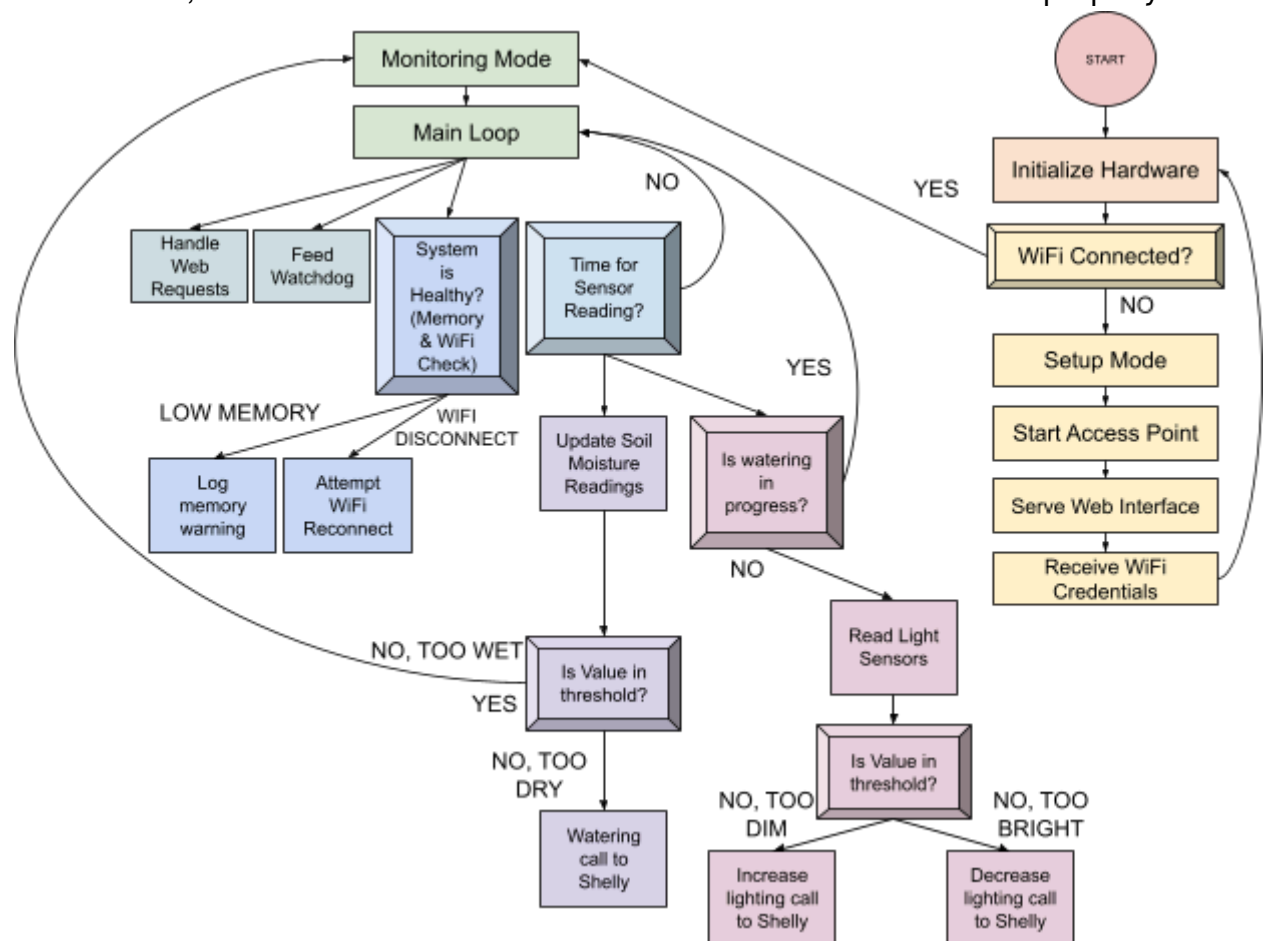


Figure 19: Soil Moisture Embedded Software In-Depth Flowchart

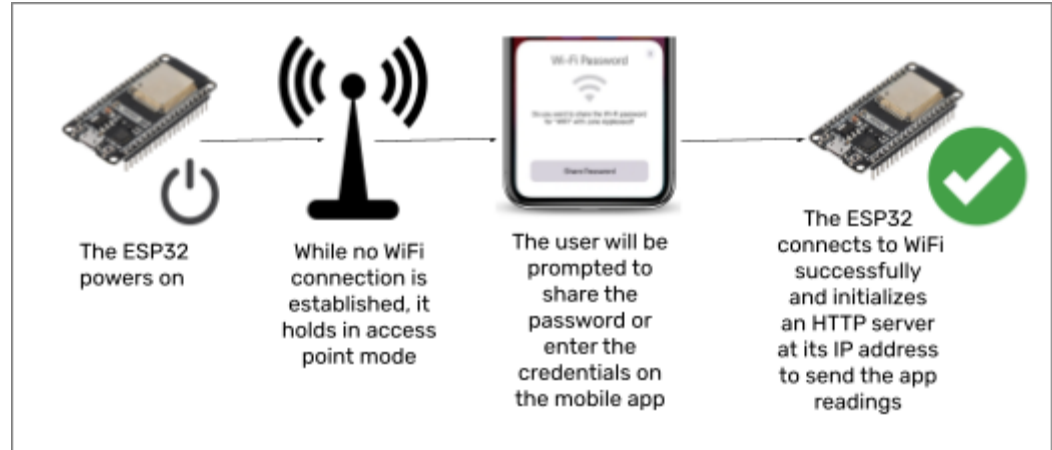


Figure 20: Access Point Process for WiFi Connection

7.2 User Application Design

As discussed in section 3, the overall design choice for the app development was Flutter, due to its ability to cater to UI needs, manage real-time system syncing with the ESP32, and high performance in such a project. While an aesthetic application was appealing to our group, our highest priority was functionality and ability to communicate with the main MCU.

7.2.1 Technologies

Flutter is currently being incorporated into our project as the main mobile development platform. Due to the easy integration, through the VSCode IDE, the Flutter extension was implemented to allow simple editing in one familiar place.

However, as development with Flutter continues, the initial technology used to aid in the development and planning was Figma. Not only did Figma allow us to make a full to-scale mock-up of the inner workings of our application, but the AI plugin tool UX Pilot allowed this initial sketch to be created easily and allowed prompt-assist to ensure any features difficult to enable on our own were incorporated in that manner (Figures 13 and 14). This approach was important in ensuring consistency in the mobile environment despite the possibility of multiple contributors.

In order to keep the code relevant for all members of the group, we decided to keep our code in the same GitHub Repository as the embedded program, hosting as a folder for the full project and all software.

7.2.2 Mobile UI Design

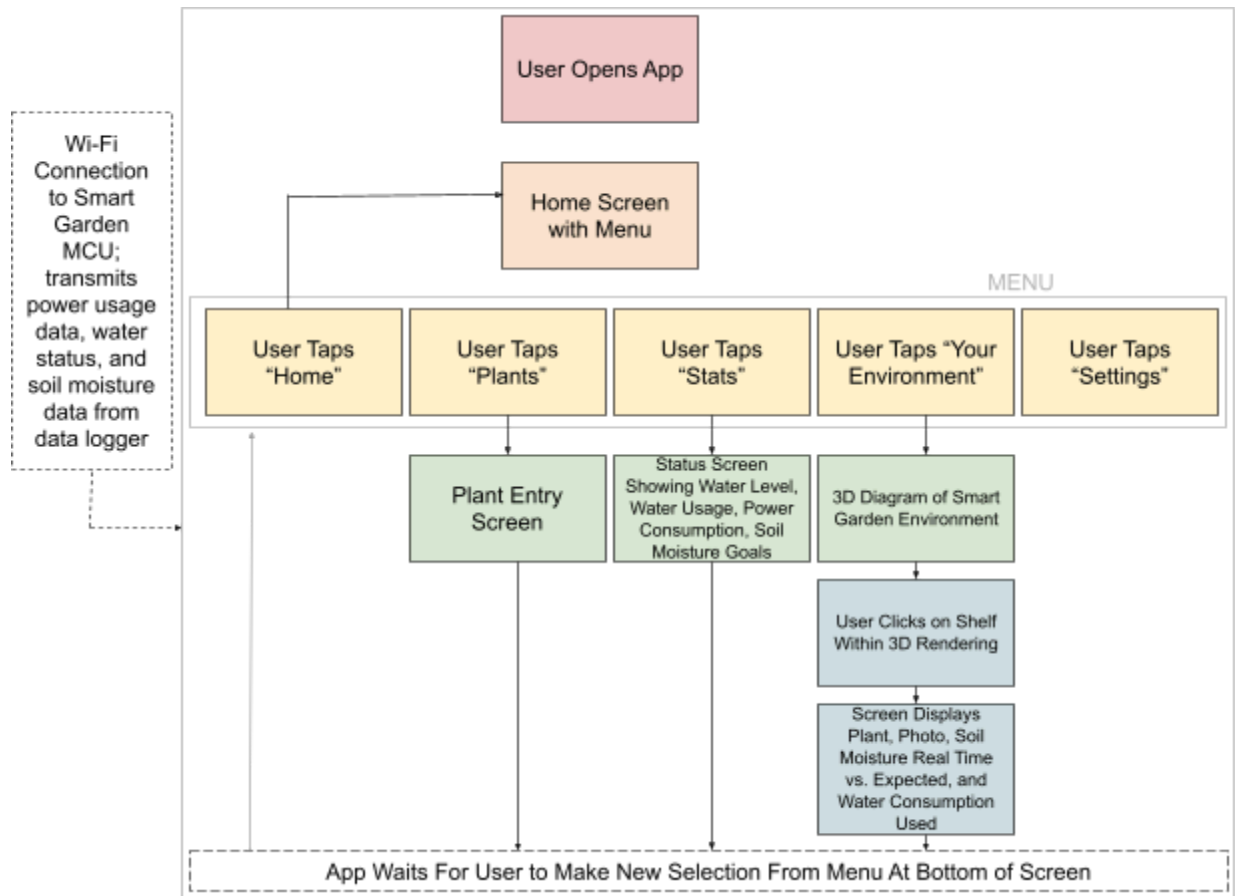


Figure 21: General App Logic Flowchart

The application will act as the user's one-stop-shop to access and update information about their personal F.L.O.W.E.R.S. environment. Safety measures will be included in the format of a traditional login to ensure secure access to only one's own personal smart garden hub. Considering that the environment does not hold any sensitive data, it is unlikely that we will implement two factor authentication (2FA), but that is a possibility which still remains to be decided.

The user will have access to 4 different main tabs, hosted in a menu bar which permanently remains at the bottom of the screen in the app: Home, Plants, Stats, and Settings. The app automatically loads into the home screen with a few quick buttons available to easily view water usage, power consumption, and lighting levels. This home page will highlight any plants which may need attention and host a notifications tab where users can see the history of water deployment and any soil moisture readings in a "critical range" resulting from lack of water in the storage basin.

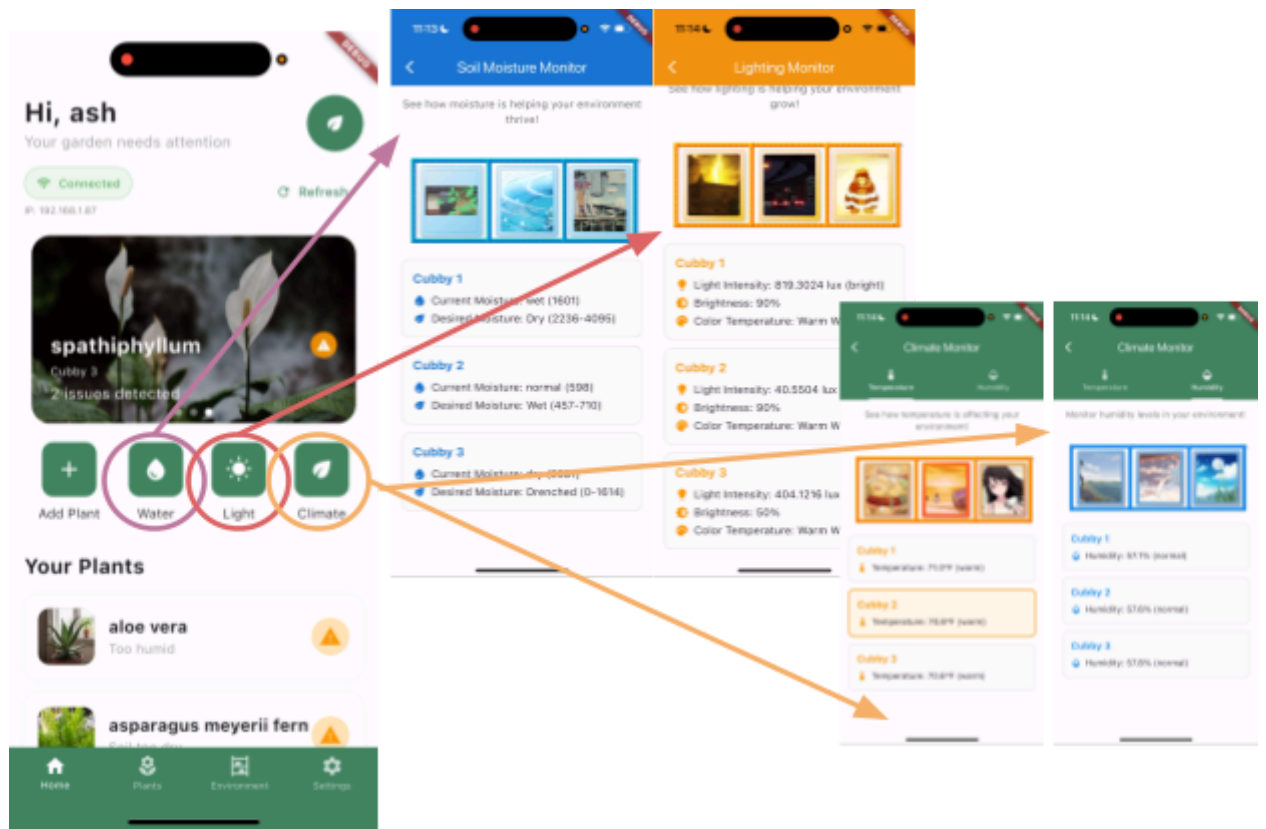
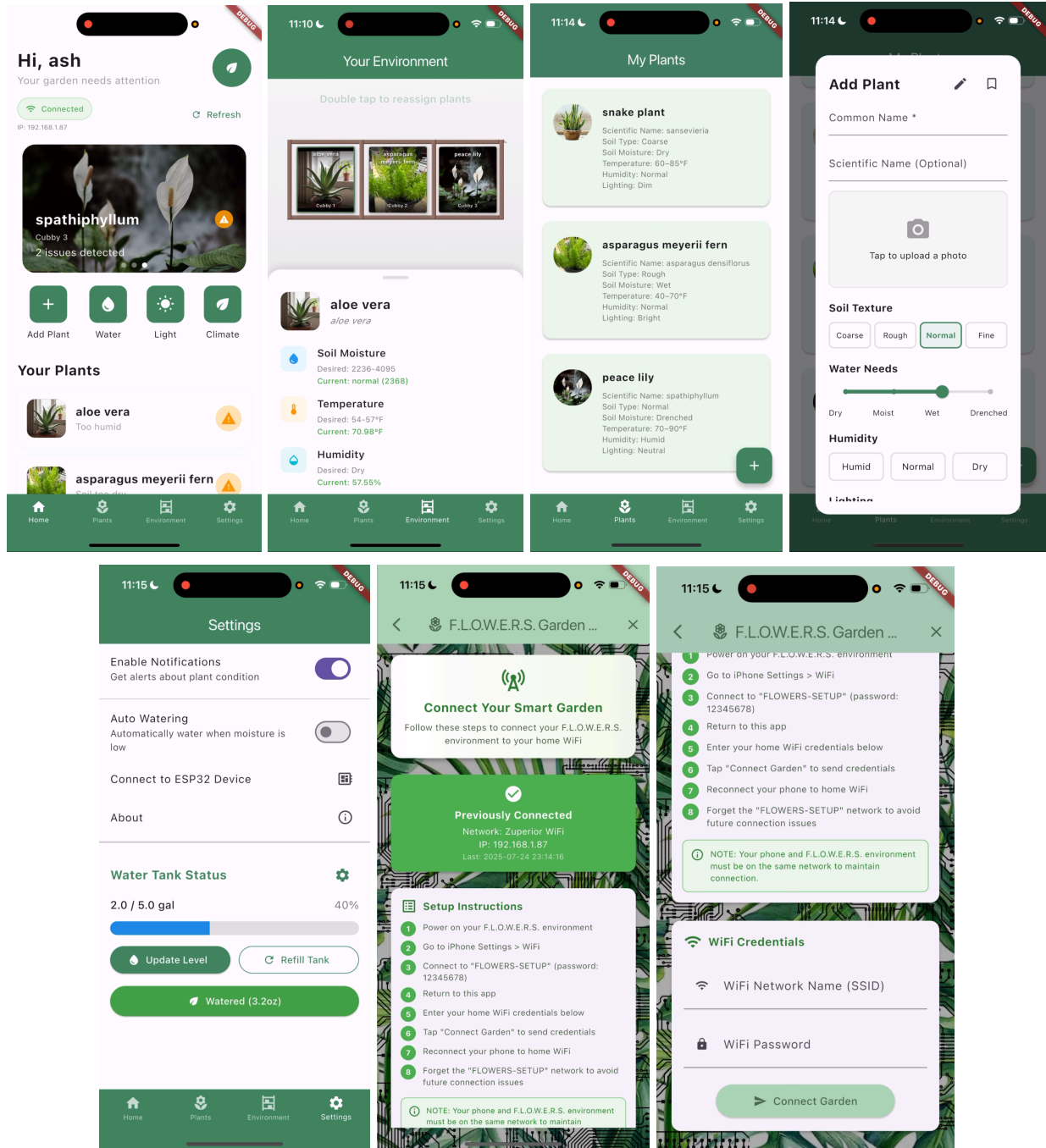


Figure 22: Home Page with Hot Keys Explained



Figures 23: General Mobile App Walkthrough

While notifications will always be sent to the notifications tab and viewable there, the user will be able to toggle push notifications for certain events. For example, most users probably do not want to be push notified each time that water is deployed to a plant. However, they will probably want to be push notified if their main water canteen is getting empty. On the home page, there will also be quick access buttons to the 3 plants currently in the smart garden environment.

When the user clicks on the “Plants” tab it transitions them into their plant library. They will be able to see existing plants, edit them, delete them, search them, and add them. In this tab, users will have full availability to the CRUD operations: Creation, Retrieval, Updating, and Deletion. When the user clicks to add a new plant, it allows them to put in an optional picture of the plants, a name, and all of the information to guide the environment for that plant such as soil moisture needs, lighting needs, etc.

When the user clicks on “Stats” it will give them 2 options: to either view the system stats or the individual environment stats. If the former is chosen, the power and water consumption for the entire system over the previous week, as well as the current water level in the main water basin servicing the system.

The “Settings” tab will allow the user to change their push notification settings, change the Wi-Fi settings in the system.

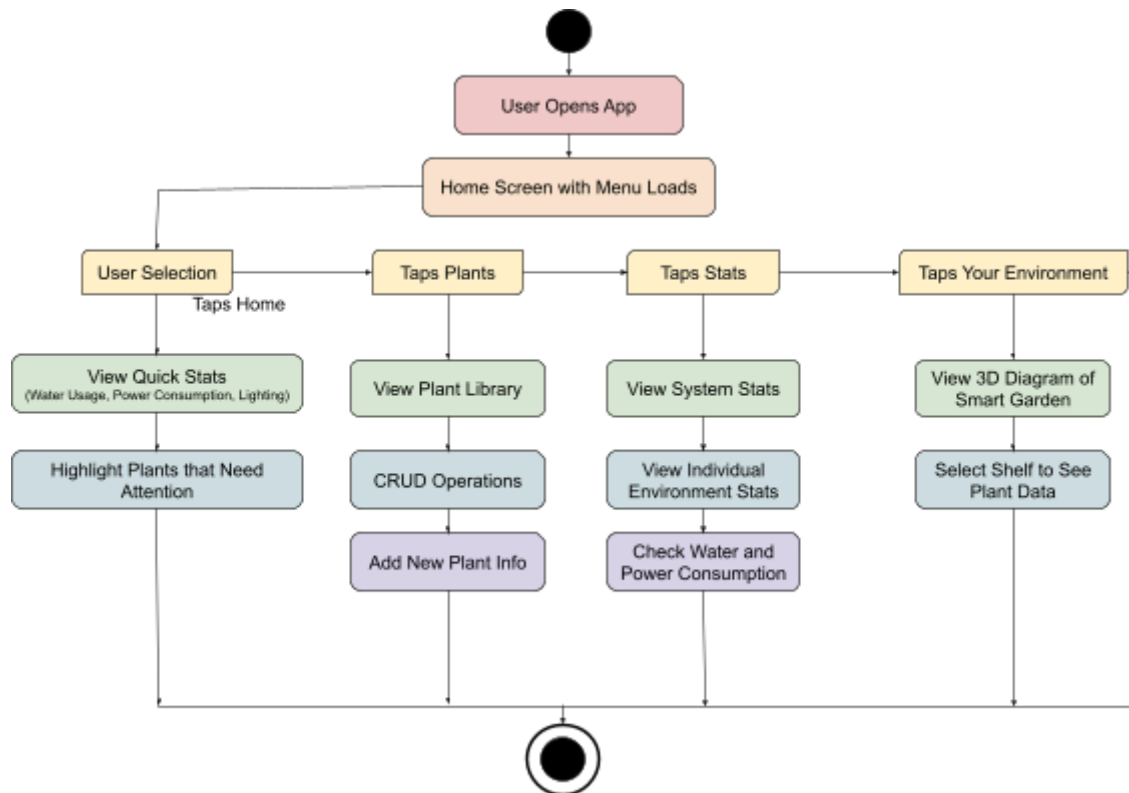


Figure 24: Activity Diagram for the Mobile Component

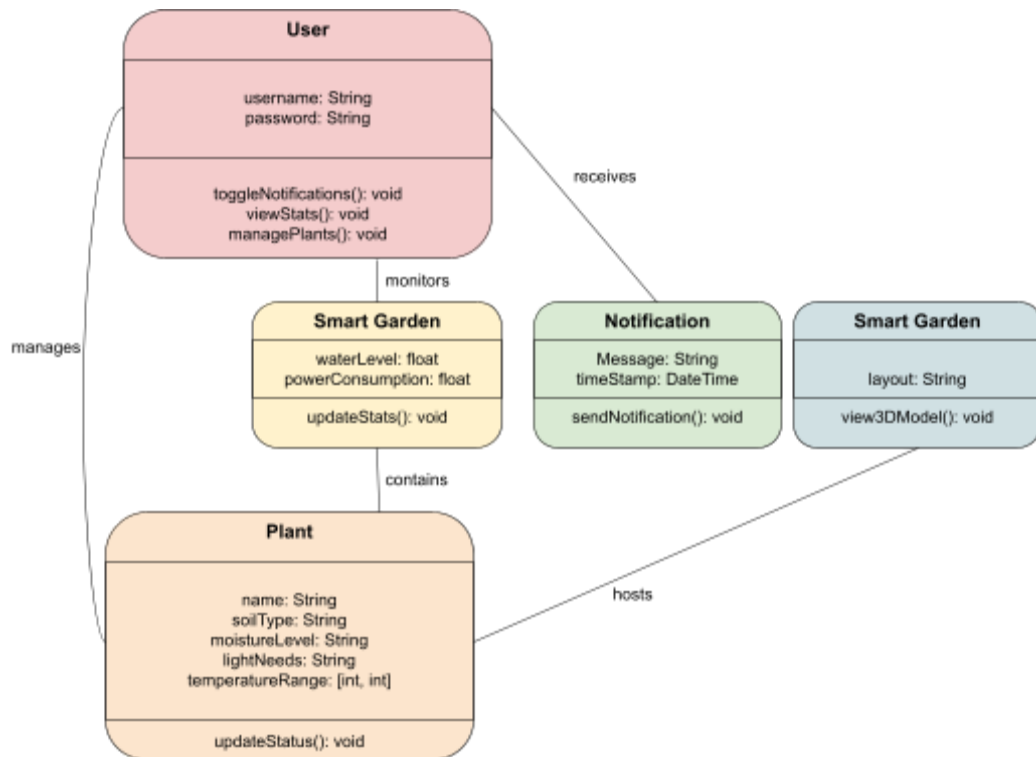


Figure 25: Class Diagram for the Mobile Component

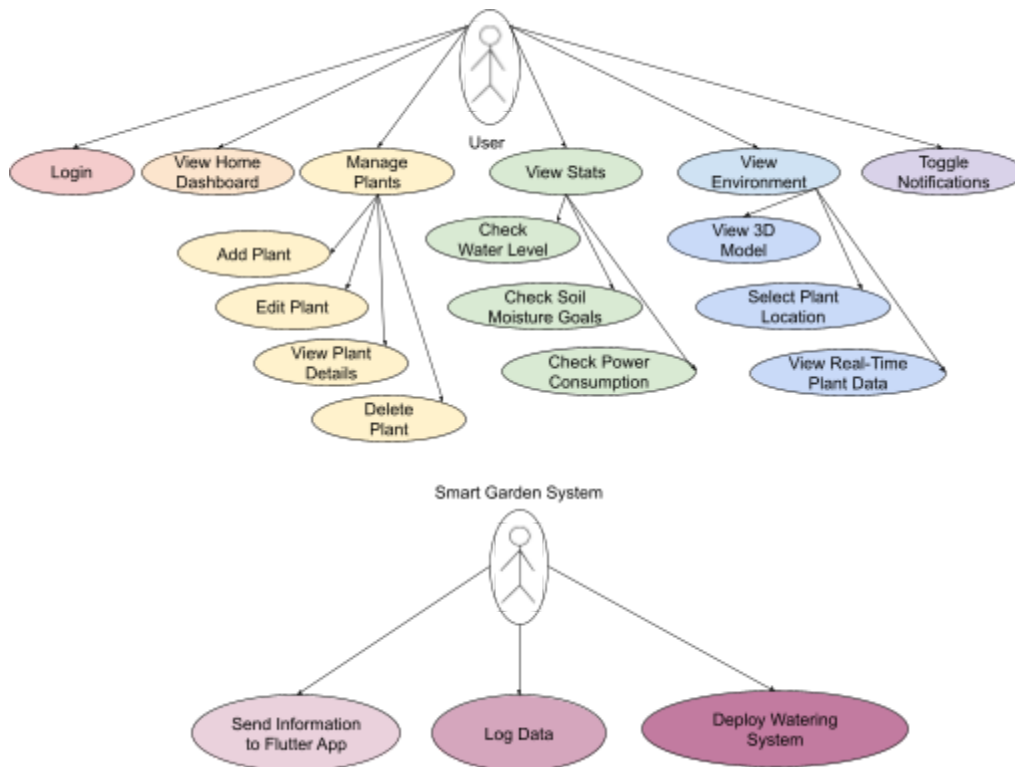


Figure 26: Use-Case Diagram for the Mobile Component

7.3 Hard-Coded Values

As a general rule of thumb, software development principles discourage the implementation of hard-coded values. While this is usually the case, it does not apply to grounding values, which are used as values that determine a certain constant.

7.3.1 Constant Values in the Soil Moisture Sensor Readings

In this case, we will be required to include some hard-coded values as reference for the system. These hard coded values will be relative to the soil moisture values gathered by the sensor. While the testing is still currently in the process of being completed, it is important to understand that the values being gathered will be the ranges that determine if the soil is dry or wet. Essentially, after several bouts of testing and value determinations, there will be a maximum value and a minimum value. These values will be integrated within the distributed software to allow the ESP32 to determine the “driest” which soil can be and the “wettest” which soil can be. Thus, from these values we will be able to derive a percentage of moisture which the soil can be, ranging from 0% (driest) to 100% (wettest). It is worth noting that the 100% moisture value will be gathered in testing without drowning the soil. So, this value of the range will refer to testing in which water is not pooling underneath the soil, as this most definitely exceeds the highest moisture which the soil is able to handle.

While the trials for soil testing are still in place, the gathered value for regular water is available to be analyzed (Appendix E). The values which determine that the moisture is in a full container of plain water should greatly exceed the values expected for greatest soil capacity in terms of moisture. Therefore, it would be expected that soil measured to its “full moisture capacity” (for our project’s purpose) would be expected to be greater than 1000, but closer to about 1200 than 2500. However, the soil moisture sensor may be closer to a 2500 value when submerged in completely dry soil.

7.3.2 Constant Values for ESP32 Broadcasting

Since the communication via ESP-NOW will require the broadcasting to occur using the MAC address, these MAC addresses corresponding to the main MCU (the receiver) will be placed directly into the programming for each soil moisture sensor, in order to ensure that each sensor is sending their soil moisture readings to the main ESP32. It is likely that our project will include a filtration feature in this broadcasting from the receiver’s side, in order to ensure that any

signals from outside the system's concern are not able to interfere with this broadcasting process. In order to do this, the ESP32 attached to the driving unit of the smart garden will have each sender's MAC address included in the program in order to only accept these relevant signals from the soil moisture sensors.

8.0 System Fabrication

8.1 PCB Layout

In our PCB design we chose to separate the regulator board from the main ESP32 board. This was implemented so that we can test the regulator board to make sure that it works properly without having the concern of ruining the entire board. The trace sizes to connect each of these components is 32 mil. This is due to the current requirement from our esp32 which can reach up to 780mA when WiFi is enabled. For how large the regulator board is and how large some of the parts are, we will need to have some type of support structure underneath this board to avoid any stress that can be caused to the pins and prevent any damage to the board.

The second board shown is the main board for our project. The 3.3V regulator board will be placed on top of the esp32 board where enough space will be given to access the rest of the parts that are on our board. We will be using vias as test points in our design so having the rest of the board accessible is important, that way we can verify connections or route wires to other traces along the board as necessary. The main part of our board will have three SHT45 temperature/humidity sensors and three VEML 7700 light intensity sensors. These sensors are connected to an I2C multiplexer to allow for these sensors to be addressed correctly since they have the same addresses. The pin headers are placed on the edge of the board to allow for ease of access for the user so that they can be pulled off of the side of the board.

The current state of the PCB layout is most likely not finished. Further development will need to be done by the user and is awaiting review from Dr. Weeks. The overall design could change as well if different criteria need to be met for our PCB.

8.1.1 3.3V Regulator Board

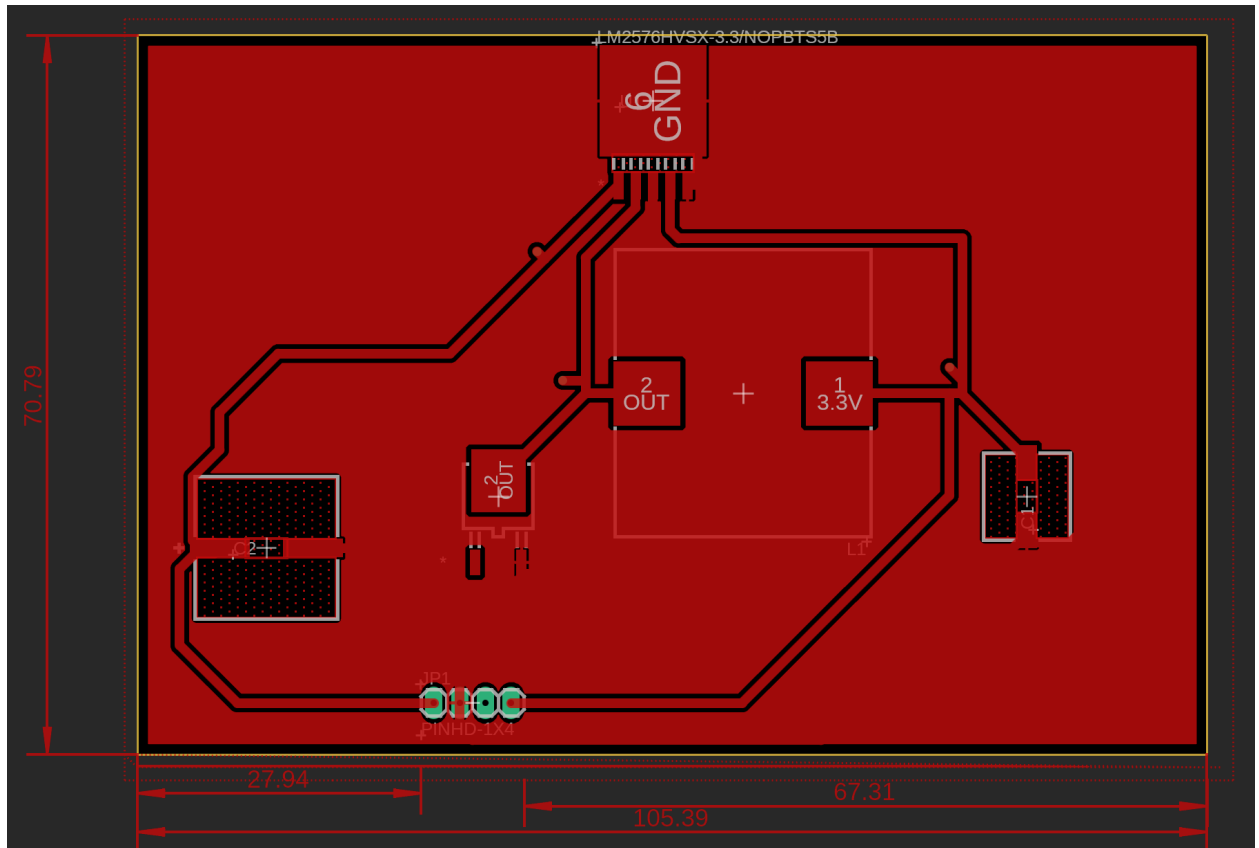


Figure 27: 3.3V Regulator PCB Layout

8.1.2 ESP32 Board

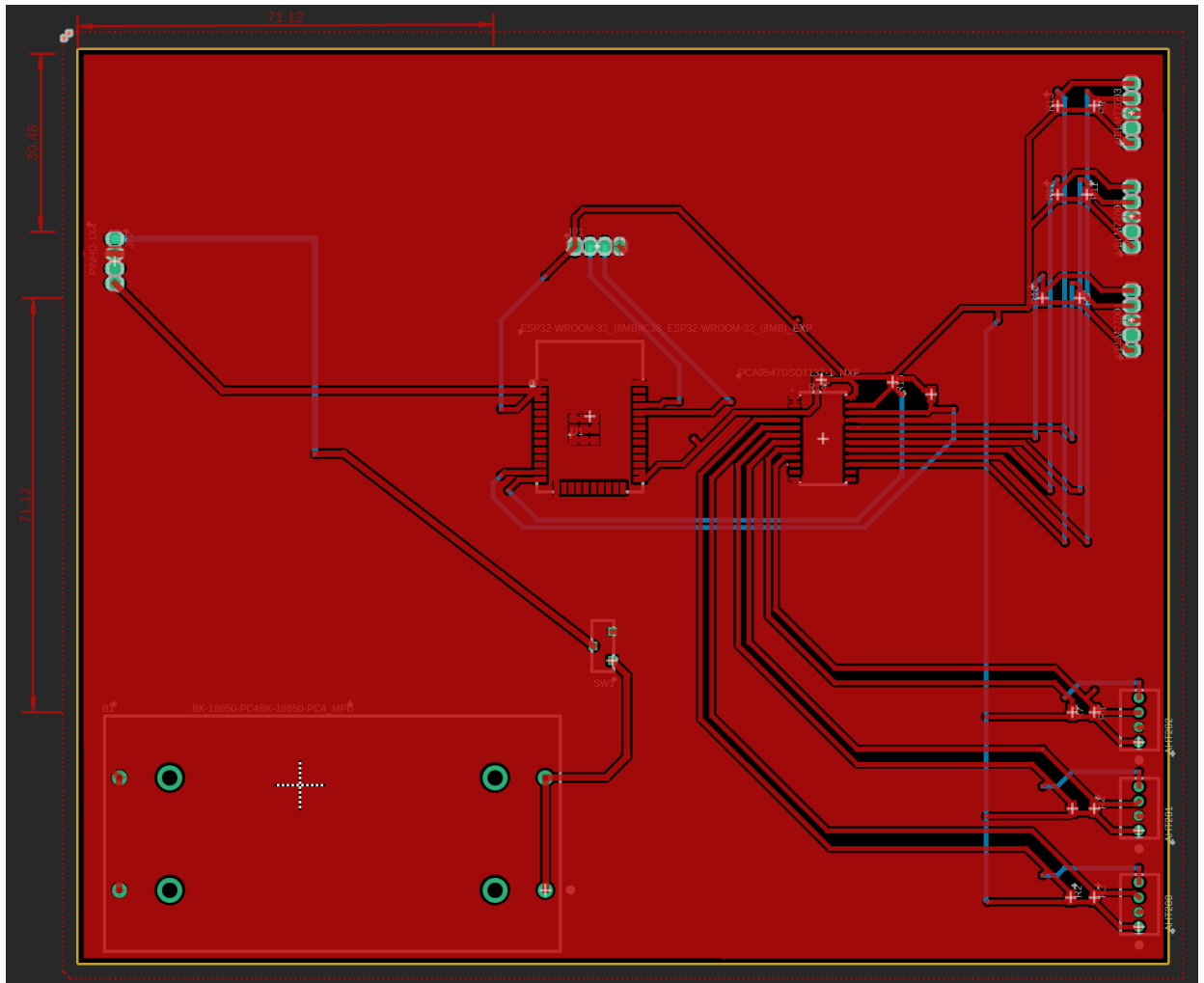


Figure 28: ESP32 PCB Layout

8.1.2 VEML7700

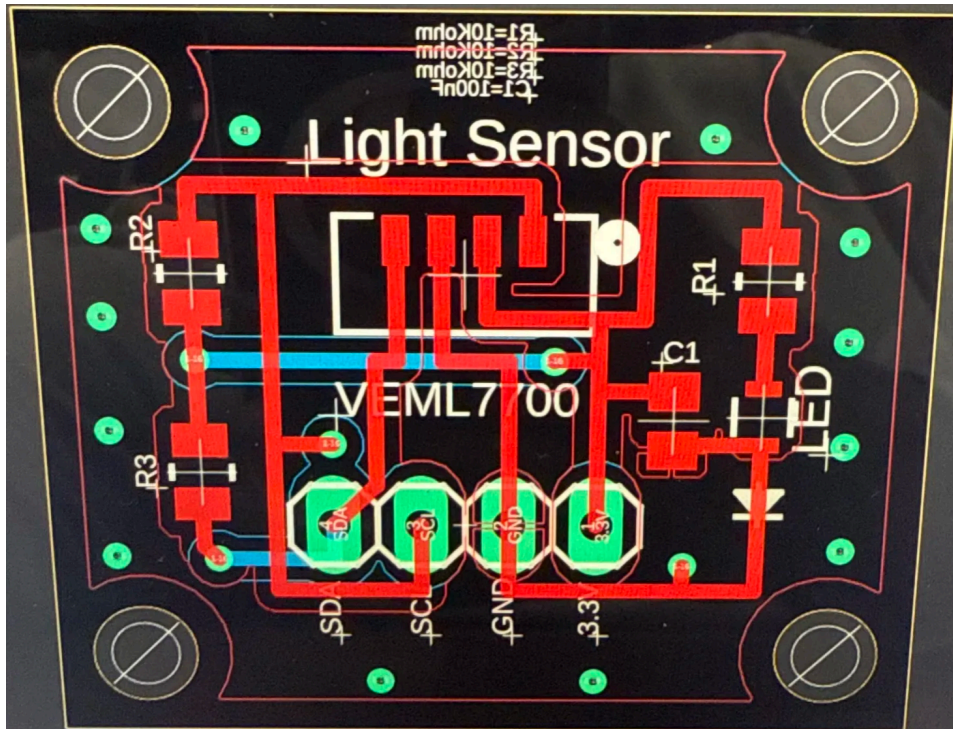


Figure 29: VEML7700 Layout

8.1.2 SHT45

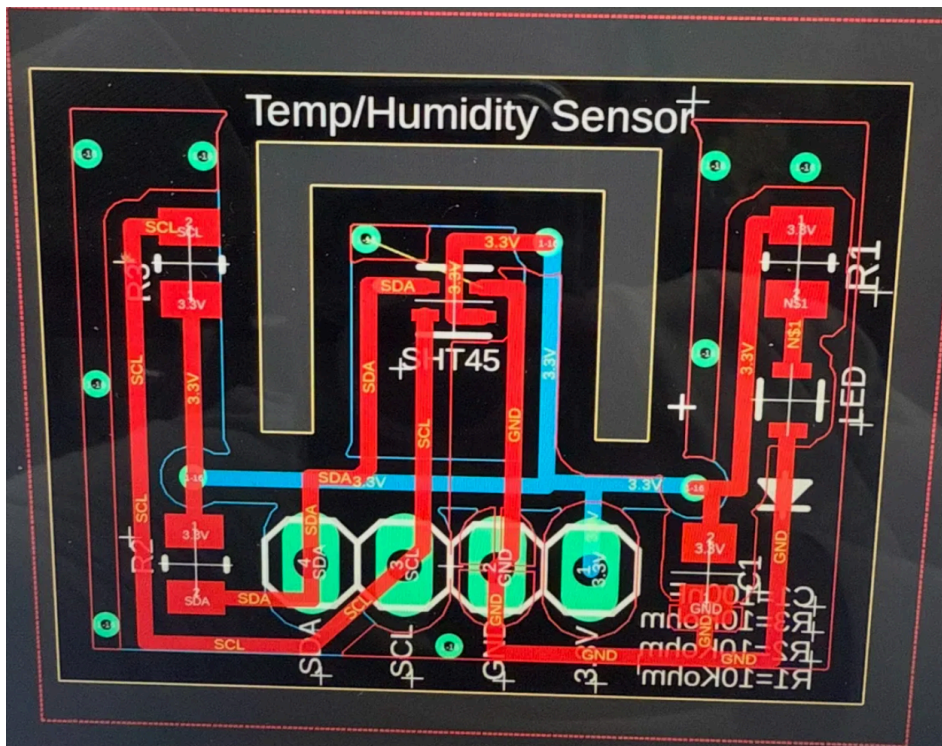


Figure 30: SHT 45 Layout

8.1.2 Capacitive Moisture Sensor

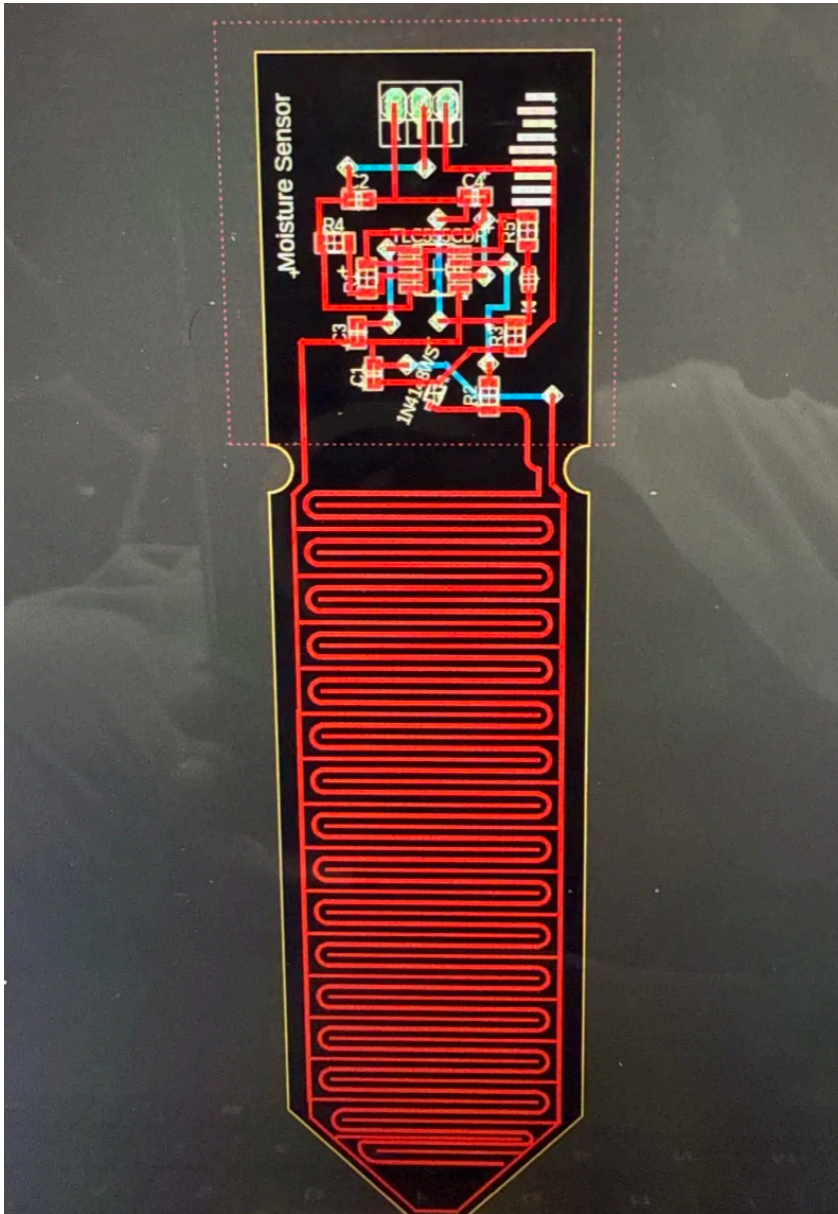


Figure 31: Capacitive Moisture Sensor Layout

8.2 Plant Shelving

One rather difficult section to include is the small portion on the shelving used to actually hold the plant pots, by virtue of the subject matter. As discussed very briefly in the research section, literally any grade of commercial plywood will suffice for holding plant pots and the associated soil. All grades can be easily cut to size, drilled for piping, and all grades can be easily sanded and painted and for these reasons, simply the cheapest plywood, being 1/4" untreated plywood was chosen. However, because the shelving is one of the last portions of the project from a chronological point of view, the actual construction itself has not yet occurred, and therefore, fabrication has not been completed. However, the steps of the fabrication will be included below, as well an AI generated envisioning of how the shelving will look to provide the reader with a concept of how it will look. This portion will be edited after system-wide fabrication has occurred.

8.2.1 Shelf Fabrication

The Shelving fabrication will be a rather simple process. Fundamentally, the shelving can be thought of like a bookshelf; there will be three enclosed sides, top and bottom closed, and one open side. However, the difference is that the partitions will be vertical, as each plant will sit adjacent to each other, instead of on top of each other, like shelves in a book shelf. Additionally, the backing plate of the shelving will itself have some room behind it; this is where the water pump, water tank, main MCU, piping, LED driver, and AC to DC converter will be located; obviously this will require careful space management, but should be feasible given the verticality of the shelving. In terms of how the actual structure will be fabricated, sheets of 1/4" plywood will be purchased, and then cut to match the rough dimensions outlined in the footprint section in Chapter 3. From here, the 5 different panels that enclose 5 sides surfaces will be screwed together with wood screws. Additionally, the vertical partitions will be cut and screwed to the top and bottom panels, forming three volumes in which a plant pot can be stored. The back panel, which provides access to the water tank for filling, will be on a hinge, and can be opened by the user to supply the system with water.

Ultimately, a slightly different water configuration was used; instead of using a back panel, a standalone tank was used to draw water from. The final system configuration can be seen in the picture below:

9.0 System Testing and Evaluation

As of now, the system hardware has been mostly acquired, and therefore preliminary testing and evaluation was able to be done on every sub-system to some extent. For example, the water pump and solenoid valves have been tested for functionality, the ESP32's have been tested both for their ability to collect data via I2C from the temp/humidity sensors as well as its ability to read the moisture level from the moisture sensors. Additionally, the ESP32's have been tested to ensure that two-way communication between the ESP32s is possible via WiFi, and furthermore, this communication can be daisy chained to simulate communicating with each of the moisture sensors separately, as is intended for the project. Furthermore, app development has begun and evaluation performed on Flutter to ensure it provides a user friendly UI, and can communicate with the main ESP32 MCU. Each of these systems and the associated testing will be included below.

9.1 ESP32 Testing

As should be clear from the sections above, our ESP32 MCU's will need to provide a variety of functions. On the one hand, the ESP32s used for the moisture sensor will need to be able to collect data from the moisture sensor located on the PCB, and then send this data to the ESP32 which we refer to as the "main MCU". This central ESP32 will receive the data from each ESP32 located on the moisture sensors, and then switch on the solenoid valves depending on the received value. Given this structure, testing was done to ensure that communication between the moisture sensor ESP32 and the central ESP32 was smooth and working correctly. From the beginning, the intention was to use WiFi in order to facilitate this communication, but a protocol developed by Expressif called "ESP-NOW" allows for seamless communication between the chips, as long as they have an internet connection, which streamlined the entire process. In our concept, this communication simply needs to allow one way communication; the moisture sensor ESP32 never needs to receive information, as the control information sent by the main ESP32 is sent to the Shelly RGBW2 to control the valves, not the moisture sensor ESP32. However, in case of LED driver problems, it was thought by the project developers that establishing two way communication, in case the WLED driver was needed as a replacement for the Shelly, would be prudent. Therefore, using the Arduino IDE, the ESP's were tested and confirmed to be able to send data values simultaneously and supported both TX and RX (Transmit and Receive) for both devices; given that two way communication was established, one way communication can be achieved by simply not sending anything from the main ESP32 to the moisture sensing ESP32.

In order to connect the ESP32s for sending and receiving, it was crucial for the MAC addresses on the devices to be included. In consideration of one-way communication, though, it became clear that only the receiver's MAC address needed to be included within the embedded code corresponding to the sender. This is because the sender must know where to send the data, while the receiver has no need to identify the received location. However, as the project proceeds, it is noted that if for any reason filtration is implemented, in order to only allow a sent signal from an approved sender, then the MAC address of the ESP32 which is sending the signal will need to be included in the receiver's programming.

9.2 Moisture Sensor Testing

As mentioned above, the information actually being sent between the moisture sensing ESP32 and the central ESP32 is soil moisture data. One of the critical goals of this project is to be able to collect soil moisture data, and send it wirelessly to drive the water system, therefore making a feedback loop. A critical component of this is being able to accurately measure soil moisture readings. Some specific soil moisture readings have been included in appendix E, which show raw values collected by the capacitive sensing. These values will be converted to a percentage to be displayed to the user via the app, and will also be the basis for which the solenoid valves open. The values generated during the testing reflect those of the two extremes; one set of values reflects the moisture of air conditioned ambient air, and the other set of values were generated by submerging the sensor entirely in a cup of water. It is safe to assume that soil moisture readings will fall somewhere in this range, and the job of the project developers will be to assign an absolute moisture value to some range of readings from the sensor, and then based on the type of plant being inside the plant pot, the valve will open to allow enough water to meet this threshold. In this case, the testing proved successful, as the sensor was able to provide the moisture data, which was sent in the ESP32 testing section (see above).

9.3 Humidity/Temp Sensor Testing

The humidity and temperature sensing are a bit different in implementation compared to the moisture sensing. This is because the Humidity and temperature sensors are directly connected to the central ESP32, and are not transmitted over WiFi, but instead directly over a wire. This of course reduces the complexity in one sense, but raises another problem to deal with, which is multiplexing. For the moisture sensors, the main ESP32 is simply able to switch which device it is connected to in order to receive the different information, but from directly connected wires, the data is sent over I2C. Therefore, an I2C multiplexer is needed to know which temp sensor data is being read by the main

MCU, as this information will be displayed on the app for that particular plant. For the preliminary testing done, only functionality was tested for the humidity and temperature sensors, and not the multiplexing ability. The values received were accurate, given that the testing occurred in an air conditioned room (High relative humidity, moderate temperatures were expected and received). For the next step, when the I2C multiplexer arrives and is connected to the sensors, the functionality will need to be tested by ensuring that the ESP32 is able to correctly determine which plant environment has which values, and correctly display this on the app.

9.4 Water Pump/Solenoid Valve Testing

The water Pump and Solenoid valves are of course another critical aspect of the project that requires testing. The testing of this particular subsystem is a bit complex, as there are multiple factors that need to be considered for the product to work. For example, both devices need to first and foremost actually turn on when powered. Additionally, the water pump must be able to bring water in a stretch of pipe up to 50 PSI. The pump then needs to turn off via its internal pressure regulating switch. The solenoid valves are on the end of this pipe, and must be able to support the 50 PSI of pressure that are developed by the pump. Then, after a voltage is sent to the valves, they must stay open until the power is no longer supplied, in which case they close, and the pressure is re-developed across the system. All of this functionality needs to be tested to ensure a working system before it is fully connected. The testing that has occurred so far is simply to ensure that the pump turns on when applied with power, and ensuring that all the solenoid valves open and close within the time frame given in the specification table, which is the case by a large margin. What remains to be tested is whether or not the pump successfully maintains a 50 PSI of pressure and ensures that the valves close even while pressure is developing behind them. Based on the specifications given by the manufacturers, this should not be a problem whatsoever, but it does need to be tested to ensure the system works properly. After testing took place, all related devices were ensured to function properly

9.5 Light System Testing

The Last major hardware sub-system that requires testing and evaluation is the lighting system. The lighting system is expected to take input from the app user about the type of light they want to plant to receive; they will be informed of the effects of different colors (frequencies) of light and how it affects plant growth stages. After the information is entered into the app, this data is sent to the central ESP32, which then sends a control message to the LED driver. This

message will include information about what brightness to run the lights at (or conversely, how dim the lights should be relative to maximum brightness), as well as which color should be on the LED strips. After the Driver receives this information, it should accordingly turn on the LED strips. In order to ensure the correct brightness, Light sensors within the information will track the incident light, and send this information back to the ESP32, which will again create a feedback loop, as the ESP32 will then lower or brighten the lights by sending another signal to the LED driver. For this subsystem, the light sensor was tested to ensure that it was able to read values correctly. As far as the sensor goes, it only needs to be calibrated such that a certain reading corresponds to a certain threshold of light intensity, as similarly to the moisture sensor, it only outputs values, and the range must be constructed such that the values fall within a range of 0% light (no lights are on) to 100% incident light (the LED light strips are on at full power). No testing thus far has been done on the LED driver itself, although the communication between the driver and the ESP32 has been elaborated upon in the 9.1 subsection above. After testing took place, all related devices were ensured to function properly

9.6 SD2 Testing Plan

Given the above information, the following components and functionalities must be tested in SD2 to ensure a working products in each of the subsystems

Water Delivery Subsystem -

- Test water pump can maintain 50 PSI of water pressure across the length of the piping
- Test that the solenoid valves can close while the pump is trying to force it open

Lighting Subsystem -

- Test that LED light strips can be turned on and off by the LED driver
- Test that the LED light strips can be dimmed by the LED driver
- Test that the LED light strips can have the color changed by the LED driver
- Assign a range of values for the raw light intensity values

Main MCU/peripherals Subsystem-

- Ensure ESP32 can communicate with the LED driver

- Ensure ESP32 can use the I2C multiplexer to determine which sensor is sending which value

Moisture Sensing Subsystem-

- Test that the final PCB moisture sensor accurately collects data

- Ensure that the final PCB moisture sensor sends data to main MCU within a reasonable time for watering system's response to not affect reading values

App/UI Subsystem-

- Test that the mobile application works from start to finish: allowing user to create an account, move from tab to tab, edit desired values, and log out

- Ensure that the application involves state-saving storage (i.e. upon logging out and logging back in, the user's account should have no changes

- Test that the specified values on the user's behalf are actually changing the behavior of the driving ESP32 to the desired range

- Ensure that the communication is completely two-way between Flutter and the driving ESP32

10.0 Administrative Content

10.1 Budget and Financing

This project will not have a sponsor external to the group members, and therefore, the entirety of the cost of development will be divided equally between the four members. We will construct the budget table by dividing the components into the following sub-systems: The safety features, capacitive moisture/temperature sensor, the app, the MCU, the water system, and the lighting system.

10.2 BOM

Table 25: BOM table

Bill of Materials EEL 4914			Total Cost		\$375.22	
Part Name	Part Number	Vendor	In stock?	Unit Price	Delivery Date	Total
12V DC Water Pump	Pauline CY/711202470378	Amazon	Yes	1 for \$19.99	Not Yet Purchased	\$19.99
Schedule 40 1/4" PVC pipe	PVC 07200 0600	Home Depot	Yes	15ft at \$1.37/ft	Not Yet Purchased	\$20.55
4' x 8' 23/32" CDX Plywood	303564747	Home Depot	Yes	4 Sheets for \$36/sheet	Not Yet Purchased	\$144.00
ESP32 Home chip	1965-ESP32-WROOM-32E-H4DKR-ND - Digi	DigiKey	N/A	3 for \$2.68	Not Yet Purchased	\$8.04
40 Gallon Plastic Tank	40 Gallon Heavy-Duty Storage Tote	Amazon	Yes	1 for \$20	Not Yet Purchased	\$20.00
Electronic Solenoid Valve	Digiton ROCON-341	Amazon	Yes	3 for \$7.50/valve	Not Yet Purchased	\$22.50
Garpsen LED Growth Lights	BL-G40A	Amazon	Yes	4 for \$8.75/Light	Not Yet Purchased	\$35.00
12V 18Amp *h Battery	B0B3MHJXLF	Amazon	Yes	1 for \$38	Not Yet Purchased	\$38.00
Wood Varnish	N/A	Home Depot	Yes	1 for \$20	Not Yet Purchased	\$20.00
1/4" O.D Clear Vinyl Tubing	N/A	Home Depot	Yes	20ft at 25cents/ft	Not Yet Purchased	\$5.00
Cell Batteries	CR2032AB/10B	Amazon	Yes	10 for \$1/battery	Not Yet Purchased	\$10.00
Temp/Humidity Sensor	SHTC 3	Sensirion	Yes	3 for \$3.27/chip	Not Yet Purchased	\$9.81
10k resistor	RC1206FR-0710KL	DigiKey	Yes	12 for \$0.10/resistor	Not Yet Purchased	\$1.20
.470k resistor	13-RC2512FK-07470RLDKR-ND - Digi-Reel	DigiKey	Yes	3 for \$.32/resistor	Not Yet Purchased	\$0.96
1M resistor	P1.00MHDKR-ND - Digi-Reel	DigiKey	Yes	3 for \$0.1/resistor	Not Yet Purchased	\$0.30
32khz Oscillator	CFS-20632768DZBB	DigiKey	Yes	3 for \$.20 each	Not Yet Purchased	\$0.60
3.3V regulator	TPS613222ADBVR	DigiKey	Yes	4 for \$.58/regulator	Not Yet Purchased	\$2.32
Tandem 15A Circuit Breakers	N/A	Home Depot	Yes	1 for \$16.95	Not Yet Purchased	\$16.95

Beyond accounting for expected and estimated expenses, it is also crucial to factor in the possibility of budget overruns due to unexpected obstacles during development. Fluctuations in component pricing and availability, especially for microcontrollers and semiconductors, could require last-minute substitutions that alter the overall cost. Additionally, extra expenses may arise from shipping fees, import taxes, and expedited manufacturing for prototype revisions. To prevent these issues from surprising our final project cost, it is assumed below that there will be a variety of miscellaneous charges, and the group will simply proceed under the assumption that more money will be spent than initially accounted for; this is standard procedure for any design project. Having this flexibility will ensure that sourcing components and refining the design can continue smoothly without exceeding reasonable spending limits.

We can reference the BOM above and categorize the budget for each subsystem listed below. As an important note, the table below will double count a few items, as some components are used in multiple sub systems, like the CR2032 batteries. This give us:

Table 26: Budget Table

Sub-System	Sub-System Cost	Running Total Cost	Percentage of allocated Budget
Capacitive Moisture Sensors	\$25.19	\$25.19	4.3%
Shelf Frame	\$164	\$189.19	28%
MCU	\$19.64	\$208.43	3.3%
Water System	\$126.04	\$334.47	21.5%
Lighting System	\$135	\$469.47	6%
Safety Features	\$16.95	\$486.42	2.9%
Misc Development	\$200	\$686.42	34%

10.3 Project Milestones/Schedule

Table 27: Senior Design 1 Tentative Schedule

Tentative Schedule For Senior Design 1 - Spring 2025		
Task	Timeline	Responsibility
Form Group	Week 1	Everyone
Brainstorm Ideas	Week 1	Everyone
Final Decision on Project	Week 2	Everyone
Divide and Conquer Document	Week 3 (Fri, Jan. 24)	Everyone
Review Divide and Conquer with Advisor	Week 4 (Weds, Jan. 29 - 3pm)	Everyone
Research Bluetooth Chip	Weeks 5/6	Ash/Junior

Research the User Interface/Methods of Implementation	Weeks 5/6	Ash
Research MCU	Weeks 6/7	Ash
Research the Batteries	Weeks 5/6	James/Junior
Moisture/Temperature Sensor	Weeks 5/6	Junior
Water Pump Parts Acquisition	Weeks 10/11	Noah
PCB Design	Weeks 10/11	Junior
PCB Review and Order	Week 12/13	Junior/Everyone
Power Supply (12V battery)	Week 10/11	Noah/James
Acquire LEDs	Week 14	James
Acquire Electronic Valves	Week 14	Noah
80 Page Draft	Week 12	Everyone
Project Prototype	Week 13/14/15	Everyone
120 Page Final Document	Week 16	Everyone

Table 28: Senior Design 2 Tentative Schedule

Tentative Schedule For Senior Design 2 - Summer 2025		
Task	Timeline	Responsibilities
Cutting, Painting, Staining, Coating, and Assembling Wooden Base	SD2 (While PCB out for order)	Everyone
Building App and Developing the UI	Begin in SD1 Complete By Week 5 of SD2	Ash
Order 2nd PCB	Week 5/6	Everyone
D Day	Week 12	Everyone

10.4 Work Distribution

Table 29: Work Distribution Table

Name	Primary Role	Secondary Role(s)
Ashley Hanzelka	App/UI Development	Embedded Software
Noah Weeks	Water Delivery System	Lighting System Embedded Software
Junior Jestine	Moisture Sensing System	Subsystem PCB Design Embedded Software
James Ventimiglia	Lighting System	Subsystem PCB Design Embedded Software

11.0 Conclusion

The F.L.O.W.E.R.S project is ultimately about streamlining and automating the process of growing plants indoors, while providing optimal growth conditions regarding water and light consumption for the plants. The main goals involve successfully delivering water to the plants, successfully controlling the light color and intensity that reaches the plants, and lastly, creating an app that allows the user to optimize these parameters by simply entering what kind of plant is in the pot. Then, using a moisture sensor developed by our team, the ideal moisture value in the soil is maintained, and similarly, the light sensor connected to the main ESP32 will ensure that the correct amount of light reaches the plant. For the user, they will be able to track certain parameters in the plant environment, like the temperature, light intensity, and relative humidity within the environment, in order to keep up-to-date on the real time condition of their plants.

From a technical perspective, the project relies heavily on two different hardware components, being the ESP32 microcontroller, and the Shelly RGBW2 LED driver. There will be one ESP32 on each moisture sensor, which will be placed inside the dirt of a given plant pot, and these ESP32's will send their information to the fourth, "central" ESP32 that will process the data and open/close the water solenoid valves accordingly. Meanwhile, the Shelly RGBW2 will power the

LED light strips , and control their on/off state, control the brightness via dimming through a PWM signal, and importantly, control the color from the LED light strips, which is an important parameter for plant growth. Furthermore, the LED driver provides a convenient method for simultaneously providing power for the water pump and solenoid valves, while also controlling them via an on/off signal. This is courtesy of the RGBW2's ability to be controlled via Wifi.

One important aspect of every engineering project is understanding the project constraints and associated standards on a deep level, and this project is no different. Three primary constraints were determined, being the battery life of the moisture sensor PCB modules, the corrosion of the capacitive moisture plates, and the security of the ESP32 to ESP32 communication given public wifi networks have inherent risk for accessing. To accommodate the battery life constraint, a specific polling rate, an efficient switching regulator, and other hardware components was chosen to ensure that battery life would be substantial. As for corrosion effects, care was chosen when selecting the capacitive moisture sensor, as opposed to sensor types like resistive sensors. In meeting all constraints, careful consideration was paid to the standards being used. For example, when considering the communication protocol between ESP32's, ESP-NOW was selected in part due to its AES encryption compliance with the 802.11 standard for Wifi communication, which includes a critically important security section regarding encryption compliance. Additionally, certain applicable health and safety standards were kept with regards to thermal regulation and plug safety; this is detailed further in chapter four of the report

In terms of progress, as described previously, most of the testing regarding both the moisture sensors and ESP32 to ESP32 communication has been completed. The bulk of remaining testing is related to the lighting system and the app, but given that these both use components/software that is immediately or very quickly available, this allows the timeline to be met successfully for Senior Design two without any concerns.

Appendices

Appendix A: References

- [1] Multiview. “Galvanic and Electrolytic Corrosion: Reducing Damage with Zinc Anodes.” *Belmont Metals*, 21 Mar. 2018, www.belmontmetals.com/galvanic-and-electrolytic-corrosion-reducing-damage-with-zinc-anodes/.
- [2] *Soil Moisture Sensors—How They Work. Why Some Are Not Research Grade* - *Meter Group*, metergroup.com/measurement-insights/soil-moisture-sensors-how-they-work-why-some-are-not-research-grade/. Accessed 21 Mar. 2025.
- [3] Qi, Qiuju, et al. “Performance of Soil Moisture Sensors at Different Salinity Levels: Comparative Analysis and Calibration.” *Sensors (Basel, Switzerland)*, U.S. National Library of Medicine, 29 Sept. 2024, [pmc.ncbi.nlm.nih.gov/articles/PMC11478466/](https://pubmed.ncbi.nlm.nih.gov/articles/PMC11478466/).
- [4] Miceuz. “Capacitance Measurement Using Low Pass Filter.” *WeMakeThings*, 26 Sept. 2012, wemakethings.net/2012/09/26/capacitance_measurement/.
- [5] Weisenhorn, Julie, and Natalie Hoidal. “Lighting for Indoor Plants and Starting Seeds.” *UMN Extension*, 2024, extension.umn.edu/planting-and-growing-guides/lighting-indoor-plants#factors-to-consider-when-you-evaluate-light-2651611.
- [6] Atherton, Lori. “Study Examines Switching from Fluorescent Lamps to Leds.” *The University Record*, 11 Dec. 2023, record.umich.edu/articles/study-examines-switching-from-fluorescent-lamps-to-leds/#:~:text=LED%20lamps%20outperform%20fluorescent%20amps,dimming%20performance%20than%20fluorescent%20lamps.
- [7] “PCB Trace Corrosion: Why It Happens and How to Prevent It.” *Altium*, 26 Oct. 2024, resources.altium.com/p/pcb-trace-corrosion-why-it-happens-and-how-prevent-it.
- [8] A., Obagade T. “Comparative Analysis of Photoelectric Characteristics of Photosensors: A Case Study between Photoresistor, Photodiode, and Phototransistor.” *IOSR Journal of Applied Physics*, 2022, www.iosrjournals.org/iosr-jap/papers/Vol14-issue3/Ser-2/A1403020112.pdf.
- [9] “Luminous Intensity & Photometry.” *Luminous Intensity & Photometry*, www.auersignal.com/en/technical-information/visual-signalling-equipment/luminous-intensity/#:~:text=Illuminance%20is%20calculated%20with%20the,an%20area%20of%201%20m2.&text=The%20further%20away%2

0the%20area,source%2C%20the%20lower%20the%20illuminance.
Accessed 21 Mar. 2025.

- [10] "Photodiode Basics." *Wavelength Electronics*, Wavelength Electronics, 11 Feb. 2020, www.teamwavelength.com/photodiode-basics/?srsltid=AfmBOoqTcalA3ctSs4N-xZT6jdSBuCLIo1kMeNb63D9THxeWXM0E4acr.
- [11] "LDR – Photoresistor - Electronics Area." *Electronics Area - Electrical and Electronics Tutorials and Circuits*, 29 Aug. 2024, electronicsarea.com/photoresistor-ldr/.
- [12] "Phototransistor : Construction, Circuit Diagram & Its Applications." *ElProCus*, 17 Feb. 2021, www.elprocus.com/phototransistor-basics-circuit-diagram-advantages-applications/.
- [13] Schweber, Bill. "How to Use Photodiodes and Phototransistors Most Effectively." *DigiKey*, DigiKey's North American Editors, 11 Sept. 2018, www.digikey.com/en/articles/how-to-use-photodiodes-and-phototransistors-most-effectively.
- [14] "Incandescent Light Bulb." *Incandescent Light Bulb - Energy Education*, energyeducation.ca/encyclopedia/Incandescent_light_bulb. Accessed 22 Mar. 2025.
- [15] "Choosing a Cross-Platform App Development Framework: A Detailed Comparison." Sendbird, www.sendbird.com/developer/tutorials/choosing-a-cross-platform-app-development-framework-a-detailed-comparison.
- [16] "React Native vs Flutter vs Ionic vs NativeScript vs PWA." Academind, academind.com/tutorials/react-native-vs-flutter-vs-ionic-vs-nativescript-vs-pwa/.
- [17] "Comparing React Native vs Flutter vs Ionic vs Kotlin for Mobile App Development." Blue Whale Apps, www.bluewhaleapps.com/blog/comparing-react-native-vs-flutter-vs-ionic-vs-kotlin-for-mobile-app-development.
- [18] "Flutter vs React Native vs Ionic [Developer's Perspective]." Ulam.io, www.ulam.io/blog/flutter-vs-react-native-vs-ionic.
- [19] "The Six Most Popular Cross-Platform App Development Frameworks." JetBrains, www.jetbrains.com/help/kotlin-multiplatform-dev/cross-platform-frameworks.html.
- [20] "Cross-Platform Mobile App Development: Product Owner's Guide." MobiDev, mobidev.biz/blog/cross-platform-mobile-development-frameworks-comparison.

- [21] "Cross-Platform Mobile Development." The Pragmatic Engineer, newsletter.pragmaticengineer.com/p/cross-platform-mobile-development
- [22] "How to Choose the Right Mobile App Development Approach." Ionic, www.ionic.io/resources/articles/choosing-the-right-mobile-approach.
- [23] "Top 11 Mobile App Development Frameworks in 2025." Flatirons, www.flatirons.com/blog/top-11-mobile-app-development-frameworks-in-2024/.
- [24] "Comparison Between Flutter Vs React Native." Angular Minds, www.angularminds.com/blog/comparison-between-flutter-vs-react-native-for-mobile-app-development.
- [25] "Flutter Performance: How Flutter Achieves Native-Like Performance." *Flutter.dev*, Google, docs.flutter.dev/resources/architectural-overview.
- [26] "Why Flutter's Rendering Engine Is So Fast." *Medium*, Flutter Community, medium.com/flutter-community/why-flutters-rendering-engine-is-so-fast-abc1a44d7ac3.
- [27] "Integrating ESP32 with Flutter Using Bluetooth Low Energy (BLE)." *Pub.dev*, pub.dev/packages/flutter_blue.
- [28] "Flutter for IoT Projects: Connecting Flutter Apps with Hardware Devices." *Medium*, IoT Developers Hub, medium.com/iot-developers-hub/flutter-for-iot-projects-connecting-flutter-apps-with-hardware-devices-fdf2a3b8bcd7.
- [29] "How Hot Reload Speeds Up Flutter Development." *Flutter.dev*, Google, docs.flutter.dev/tools/hot-reload.
- [30] "Flutter Plugin Packages for Hardware Integration." *Pub.dev*, pub.dev/flutter/packages.
- [31] Espressif Systems. "ESP-IDF Programming Guide." Espressif, docs.espressif.com/projects/esp-idf/en/latest/.
- [32] Medium. "Arduino: The Ideal Gateway to Explore Embedded Systems (With some Limitations)" Medium, <https://medium.com/@CTasdemir/arduino-the-ideal-gateway-to-explore-embedded-systems-with-some-limitations-46092f65d416>.
- [33] PlatformIO. "PlatformIO Documentation." PlatformIO, docs.platformio.org/en/latest/.
- [34] MicroPython. "MicroPython for ESP32." MicroPython, docs.micropython.org/en/latest/esp32/quickref.html.
- [35] Espruino. "JavaScript for Microcontrollers." Espruino, espruino.com/ESP32.

Appendix B: Data Sheets Referenced

- [1] Espressif Systems, "ESP32-WROOM-32 Datasheet," Datasheet version 1.3, March 2020 [Revised March 2021]

- [2] Microchip Technology, “ATmega328P Datasheet,” DS40001997A, May 2016 [Revised January 2020].
- [3] Texas Instruments, “MSP430 Family Datasheet,” SLAS691B, March 2019 [Revised November 2020].
- [4] Raspberry Pi Foundation, “RP2040 Datasheet,” DS2022/1, January 2021 [Revised January 2021].
- [5] Espressif Systems, “ESP8266 Datasheet,” Datasheet version 1.6, December 2019.
- [6] Raspberry Pi Foundation, “Raspberry Pi Zero W Datasheet,” January 2017.
- [7] Arduino, “Arduino MKR WiFi 1010 Datasheet,” MKR WiFi 1010, March 2019.
- [8] Particle, “Particle Photon 2 Datasheet,” Particle Inc., October 2021.
- [9] Nordic Semiconductor, “nRF7002 DK Datasheet,” nRF7002, September 2020.
- [10] Wiznet, “WizFi360 Datasheet,” WizFi360, January 2020.

Appendix C: Referenced Software Code

Section C1: Soil Moisture Sensor Plain Water Testing - Arduino Code

```
#define AOUT_PIN 36 // ESP32 pin GPIO36 (ADC0) that connects to
AOUT pin of moisture sensor

void setup() {
    Serial.begin(9600);
    // set the ADC attenuation to 11 dB (up to ~3.3V input)
    analogSetAttenuation(ADC_11db);
}

void loop() {
    int value = analogRead(AOUT_PIN); // read the analog value
    from sensor

    Serial.print("Moisture value: ");
    Serial.println(value);

    delay(500);
}
```

Appendix D: ChatGPT prompts and outcomes

[AI-1] <https://chatgpt.com/share/67e134de-ebb4-8005-bc0f-12b89481d9f8>

Prompt: “What is a clever name and acronym for our smart garden project?”

Response: “*Apps for Light and Plant Growth: Planta - Allows you to input the light conditions of a room to find plants that will thrive there. PictureThis - Helps identify plants and provides care tips, including ideal light conditions. PlantSnap - Identifies plants and shares their growing preferences, including light...*”

[AI-2]<https://chatgpt.com/c/6797edad-af2c-8005-9d97-42498b4d4a09>

Prompt: “I am building a smart garden project. There will be 3 separated plant "cubbies" in a wooden frame to prevent the lighting from one plant's environment leaking into another plant's environment. The main pcb will need to be designed by us. The main pcb will need to have: 1. a water sensor to alert user when tank is running low, run 3 lighting attachments for each environment, take in soil sensor readings from 3 soil sensors (also going to be PCBs built by us), and run the water pump system that delivers different amounts to each environment based on the soil moisture readings and the lighting/water needs are set by the user. Should our main pcb be modeled after the arduino uno and use the atmega chip, or be modeled after the ESP32 board by using the Tensilica Xtensa LX6 chip. which is better”

Response: “*Both the ATmega (Arduino Uno) and ESP32 (Tensilica Xtensa LX6) can work for your smart garden project, but the ESP32 would likely be the better choice for several reasons:*

Why ESP32 is better:

Processing Power and Memory:

ESP32 has dual-core processors with higher clock speeds (up to 240 MHz) compared to ATmega328P (16 MHz on Arduino Uno).

It has significantly more SRAM (520 KB compared to 2 KB on the ATmega) and built-in flash memory (4 MB typical).

Connectivity:

ESP32 has built-in Wi-Fi and Bluetooth capabilities, making it ideal for remote monitoring or control via a mobile app or cloud platform...”

Appendix E: Collected Data

Figures E1 - E3 (left to right) show a few of the readings from the soil sensor testing in plain water. The values from 900-1800 represent the sensor touching water while the values exceeding 2000 show how the sensor responded to being removed from the water.

Moisture value: 2379	Moisture value: 930	Moisture value: 2541
Moisture value: 2384	Moisture value: 933	Moisture value: 2537
Moisture value: 2376	Moisture value: 933	Moisture value: 2530
Moisture value: 2375	Moisture value: 931	Moisture value: 2524
Moisture value: 2382	Moisture value: 933	Moisture value: 2534
Moisture value: 2378	Moisture value: 976	Moisture value: 2531
Moisture value: 2381	Moisture value: 931	Moisture value: 2544
Moisture value: 2387	Moisture value: 938	Moisture value: 2543
Moisture value: 2496	Moisture value: 939	Moisture value: 2529
Moisture value: 1490	Moisture value: 935	Moisture value: 2522
Moisture value: 942	Moisture value: 930	Moisture value: 1575
Moisture value: 930	Moisture value: 931	Moisture value: 1034
Moisture value: 939	Moisture value: 933	Moisture value: 1008
Moisture value: 929	Moisture value: 926	Moisture value: 1008
Moisture value: 929	Moisture value: 919	Moisture value: 1002
Moisture value: 933	Moisture value: 930	Moisture value: 1005
Moisture value: 937	Moisture value: 928	Moisture value: 1004
Moisture value: 939	Moisture value: 931	Moisture value: 1006
Moisture value: 930	Moisture value: 928	Moisture value: 980
Moisture value: 933	Moisture value: 930	Moisture value: 1007
Moisture value: 933	Moisture value: 937	Moisture value: 1002
Moisture value: 931	Moisture value: 1058	Moisture value: 1008
Moisture value: 933	Moisture value: 2035	Moisture value: 1003
Moisture value: 976	Moisture value: 2511	Moisture value: 1003
Moisture value: 931	Moisture value: 2526	Moisture value: 1008
Moisture value: 938	Moisture value: 2524	Moisture value: 1258
Moisture value: 939	Moisture value: 2510	Moisture value: 2390
Moisture value: 935	Moisture value: 2494	Moisture value: 2448
Moisture value: 930	Moisture value: 2509	Moisture value: 2463
Moisture value: 931	Moisture value: 2519	Moisture value: 2380
Moisture value: 933	Moisture value: 2490	Moisture value: 2379
Moisture value: 926	Moisture value: 2522	Moisture value: 2384
Moisture value: 919	Moisture value: 2514	Moisture value: 2386
Moisture value: 930	Moisture value: 2513	Moisture value: 2378
Moisture value: 928	Moisture value: 2512	Moisture value: 2384
Moisture value: 931	Moisture value: 2519	Moisture value: 2390
Moisture value: 928		Moisture value: 2384
Moisture value: 930		Moisture value: 2374