# Department of Electrical Engineering and Computer Science

## Senior Design 2 Report

UNIVERSITY OF CENTRAL FLORIDA

4: Senior Design 2 – Summer 2024

Mentor: Dr Lei Wei

# Plant Guardian


**Group 11**

Tony Chau – Computer Engineering

Steven Keller - Electrical Engineering

Keven Hyppolite - Electrical Engineering

Luz Romero - Electrical Engineering


**Reviewers**

Prof. Maddox

Prof. Sahawneh

Prof. Wasfy

# Table of Contents

## List of Tables

## List of Figures

# 1. Executive Summary

One of the core principles driving innovative technologies was the consideration of our environment and the continuous effort to improve it, starting right from our homes. Our main goal was to develop an automatic plant watering system for house plants, which could be conveniently managed through a app. This innovative system ensured that house plants received the optimal amount of water, enhancing their health and growth while minimizing water waste.

Our system leveraged advanced sensors and smart technology to monitor soil moisture levels, temperature, and pH balance, ensuring that plants were always in their ideal growing conditions. By integrating these sensors with a user-friendly app, we empowered users to monitor and control their plant care routines remotely, receiving real-time updates and notifications. This simplified plant care and ensured that plants thrived even in the owner's absence.

Furthermore, our commitment to environmental sustainability was reflected in our careful selection of cost-effective and energy-efficient components. By using inexpensive yet reliable sensors and materials, we aimed to make this advanced plant care system accessible to a broad audience, promoting widespread adoption of eco-friendly practices. Our approach combined innovation with practicality, creating a product that not only met the exacting standards of modern smart home technologies but also contributed positively to the environment.

Our automatic plant watering system represented a significant step towards smarter, more sustainable home gardening solutions. By harnessing the power of technology to care for our plants, we created a greener, healthier living environment and inspired more people to adopt sustainable habits, starting right from their homes.

# 2. Project Description

## 2.1 Project Description

When we thought about plants, we thought about life because they helped us in various ways, including providing food, medicine, reducing stress, producing oxygen, and eliminating air pollutants. Plants play an integral role in maintaining the balance of our ecosystem and improving our quality of life. Millennials became increasingly obsessed with houseplants, seeking them primarily for air quality improvement and the encouragement of self-care. The presence of greenery in living spaces enhanced mood, boosted creativity, and reduced stress. Having a houseplant came with the responsibility of taking care of it and ensuring it thrived, which involved regular watering and appropriate sunlight exposure. Proper plant care was essential to maintaining their health and longevity. However, many people were busy or traveled frequently, making it difficult to leave their plants with someone who might not have the knowledge to care for them properly. This led to issues like overwatering or underwatering, both of which could harm plant health. Overwatering caused root rot, while underwatering led to wilting and eventual plant death. Additionally, inconsistent care routines stressed plants, making them more susceptible to diseases and pests.

To address these concerns, we developed an ideal solution for those who worried about their houseplants. Our solution was a smart watering system that monitored soil moisture levels and automatically watered the plant as needed. This system ensured that the plant received the right amount of water at the right time, preventing both overwatering and underwatering. The smart watering system was equipped with sensors that continuously tracked soil moisture levels, temperature, and humidity, providing real-time data to optimize plant care.

The system was user-friendly and could be controlled via a app, allowing users to manage their plants even when they were away from home. The app provided a comprehensive interface where users could set watering schedules, monitor plant health, and receive notifications when the system detected any issues. It also offered reminders for other plant care tasks to ensure a holistic approach to plant maintenance. For frequent travelers, the app included a live-stream feature, enabling users to check on their plants remotely and ensure they were thriving.

This innovation brought peace of mind to plant owners, ensuring that their plants continued to enhance their living spaces, regardless of busy schedules or travel plans. By integrating technology with plant care, our smart watering system not only simplified the process but also fostered a deeper connection between individuals and their houseplants. This project represented a step forward in sustainable living and personal well-being, making it easier for everyone to enjoy the benefits of having healthy, vibrant plants in their homes.

## 2.2 Basic Goals

The primary objective of our smart water planter project is to ensure reliable and efficient plant care through several fundamental features:

- **Remote-Controlled Water Pump**: Integrate a water pump that can be remotely controlled via Bluetooth or Wi-Fi, allowing users to manually water their plants from a distance. This feature is particularly beneficial for users who travel frequently or have a busy schedule, as it provides the convenience of plant care without being physically present. The ability to control the water pump remotely also means that users can respond quickly to their plants' needs, ensuring they always receive adequate hydration.
- **LED Lights for Plant Growth**: Include LED lights specifically designed for plant growth. These lights will provide the essential light spectrum needed for photosynthesis, promoting healthy growth and development. Unlike traditional lighting, LED lights are energy-efficient and have a longer lifespan, making them a cost-effective solution for indoor gardening. The inclusion of LED lights ensures that plants receive the necessary light even in low-light conditions, making it possible to grow plants in any part of the home.
- **Comprehensive Suite of Sensors**: Equip the smart planter with a suite of sensors, including moisture, temperature, pH, and water level sensors. These sensors will continuously monitor the plant's environment, providing real-time data that can be accessed through a dedicated application. This data will allow users to keep track of vital conditions such as soil moisture levels, ambient temperature, pH balance, and water reservoir status. By accessing this information, users can make informed decisions about their plant care routine, ensuring optimal growing conditions.

## 2.3 Advanced Goals

Building upon the basic functionality, our advanced goals focus on enhancing automation and user customization, thereby making plant care more intuitive and efficient:

- **Application Setting Page**: Develop an application setting page within the app that allows users to set desired thresholds for various parameters such as moisture levels, temperature, and pH. For example, users can specify the optimal soil moisture level for their plants, and the system will automatically adjust the watering schedule to maintain this level. This level of customization ensures that each plant receives tailored care based on its specific needs.
- **Automated Watering Schedule**: Implement a scheduling feature for the water pump, enabling automatic watering at specified intervals. This time-based watering system will ensure that plants receive consistent hydration, even when users are away or occupied with other tasks. Users can set a watering schedule that aligns with their plant's growth cycle, ensuring they receive the right amount of water at the right time. This automated watering

system not only saves time but also helps prevent overwatering or underwatering, which are common issues in plant care.

## 2.4 Stretch Goals

To further elevate the functionality of our smart water planter, we have identified several stretch goals that incorporate innovative technology to enhance the user experience and plant health monitoring capabilities:

- **Notification System**: Implement a notification system within the app that alerts users to any significant changes in sensor data or maintenance requirements, such as low water levels, abnormal temperature readings, or pH imbalances. Notifications will help users take immediate action to address any issues, preventing potential damage to the plants.
- **Image Classification Capabilities**: Incorporate image classification capabilities using the integrated camera system. This advanced feature will allow the system to identify and classify different plant species, providing tailored care recommendations based on the specific needs of each plant type. For instance, the system can recommend optimal light and watering conditions for different plants, ensuring they receive the best possible care. Moreover, the image classification system will detect weeds or pests, alerting users to respond appropriately to maintain a healthy and productive garden space.
- **Plant Health Monitoring**: Utilize image recognition technology to monitor plant health over time, detecting signs of disease, pests, or nutrient deficiencies early on. This proactive approach to plant care will help users maintain the vitality of their plants, reducing the likelihood of severe issues that can hinder growth. The integration of these advanced features will transform the smart water planter into a comprehensive plant care system, offering unparalleled convenience and support to users.

## 2.5 Objectives

By focusing on these goals and objectives, we aim to create a comprehensive and innovative smart water planter that addresses the challenges of modern gardening. Our basic goals will establish a solid foundation for efficient plant care, providing users with essential tools and information to nurture their plants effectively. The advanced goals will enhance automation and customization, making plant care more intuitive and tailored to individual needs. Finally, our stretch goals will introduce sophisticated features that further enhance the user experience and plant health monitoring capabilities, ensuring the smart water planter becomes an indispensable tool for all gardening enthusiasts.

## 2.6 List of Requirement and Specifications

*Table 2.1: Requirements and Specifications*

| ***Requirements*** | ***Specifications*** | ***Description*** |
|---|---|---|
| ***Size of the box*** | *1 meter x 1 meter x 1 meter* | *Maximum Dimensions of the system excluding any power cables* |
| ***Power Usage*** | *50W* | *Power of the entire system excluding any external servers.* |
| ***Weight*** | *30 pounds* | *Weight of the system not including the plant or water* |
| ***Data refresh rate*** | *1 minutes* | *How often the data is refreshed on the website* |
| ***pH sensor*** | *95% success rate* | *Identify if a liquid in the reservoir is an Acid, neutral, or Base* |
| ***Moisture sensor*** | *95% success rate* | *Detects if soil is wet accurately at least 95% of the time* |
| ***Temperature sensor*** | *5% error* | *Sensor measurement accuracy compared to thermometer. (Degrees)* |
| ***Water pump control*** | *Within 10s* | *System will water the plant when triggered* |
| ***Light control*** | *Within 10s* | *Latency after activating the light in the app* |

## 2.7 Demonstrable Requirements

*Table 2.2: Demonstrable Requirements*

| **Attribute** | **Description** | **Specifications** |
|---|---|---|
| **Water pump control** | Within 10 seconds | How fast manually triggering the pump in the app reacts |
| **Moisture sensor** | 95% success rate | Detects if soil is wet accurately at least 95% of the time |
| **Camera data** | 1 minute refresh rate | How often the camera should update |

## 2.8 Constraints

*Table 2.3: Constraints table*

| ***Constraints*** | ***Description*** |
|---|---|
| ***Indoor Only*** | This system will only work inside and will not be weather proofed for outdoor weather. |
| **Must be made in 7 months** | Reduced time compared to normal senior |

| | |
|---|---|
| | design due to starting in the summer. |
| **TI product availability** | Due to TI moving facilities chips we would potentially use could be out of stock. |
| **Expenses** | Not funded by sponsor so on student budget. |

## 2.9 Project Block Diagram

For our block diagrams, we strategically allocated responsibilities to each team member to ensure a seamless and efficient project workflow. The entire system was controlled by the MCU, highlighted in orange, with Steven overseeing its implementation and functionality. Our inputs, highlighted in blue, comprised a variety of sensors, including pH, temperature, water level, and a camera, all managed by Luz. These sensors were crucial for gathering real-time data to optimize our smart irrigation system. On the output side, highlighted in black, we had the water pump and lighting components, which Keven handled. These outputs delivered the precise amount of water and light required by the plants, based on the input data received. Additionally, the system's monitoring and control were facilitated through a user-friendly app, highlighted in green, as depicted in our software block diagram. Tony led this aspect, ensuring the app provided an intuitive interface for users to interact with the irrigation system, monitor sensor data, and adjust settings as needed.

Our collaborative approach ensured that each subsystem was developed with precision and seamlessly integrated into the overall design. Regular team meetings and progress reviews aligned our efforts, troubleshot any issues, and refined our strategies. This cohesive workflow leveraged our individual strengths and fostered a synergistic environment where innovation and efficiency thrived. By clearly defining and distributing these roles, we leveraged our collective expertise and ensured each component of the project was developed with the highest level of precision and integration, ultimately delivering a robust and user-friendly automatic plant watering system.

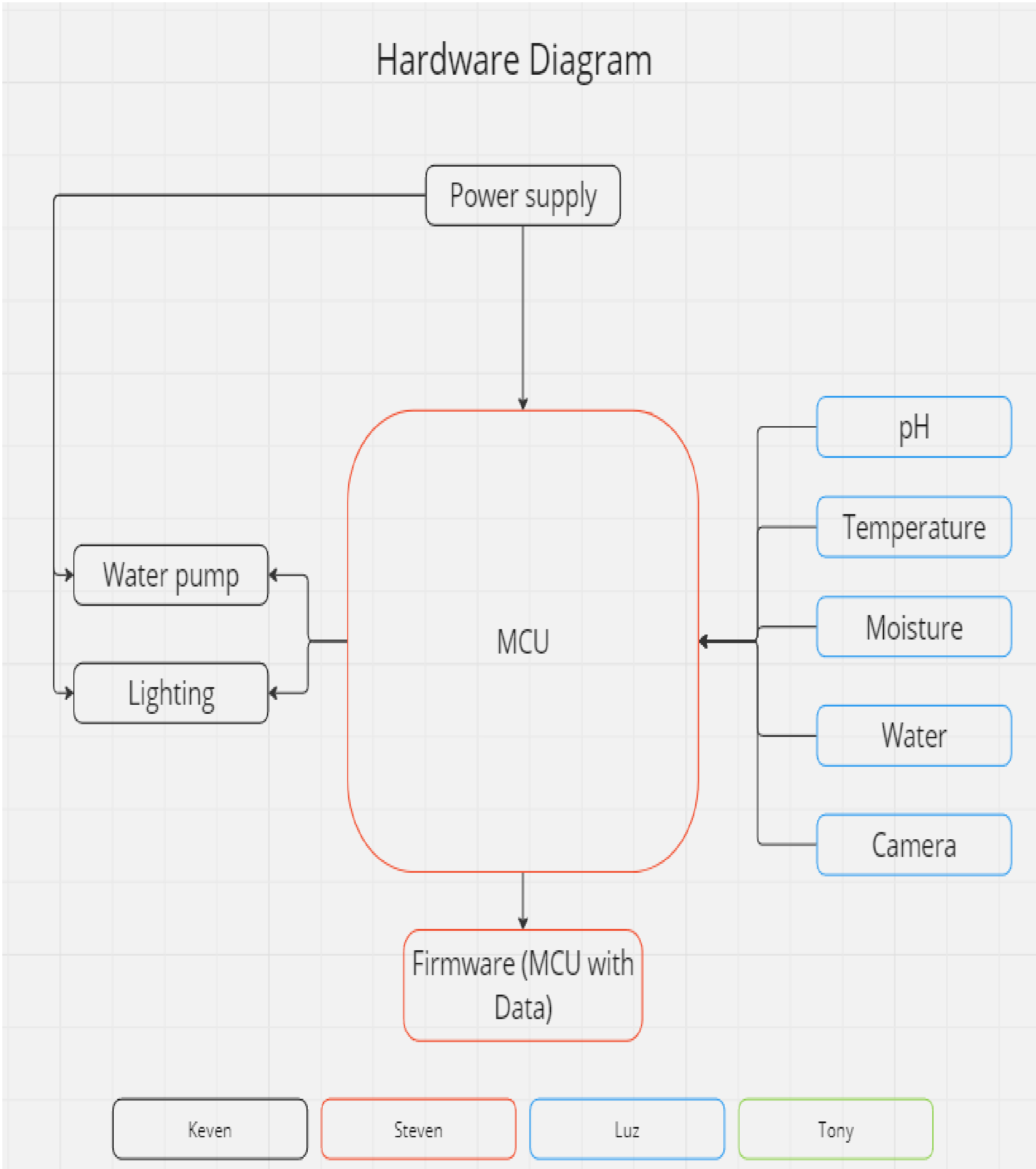



*Figure 2.1: Hardware Diagram*

At the core of the software design was the app's connectivity with the ESP32-S microcontroller via WiFi. This connection allowed for real-time data exchange between the app and the hardware components of the system. Tony ensured that the app seamlessly communicated with the sensors and microcontroller, enabling users to monitor and control the watering system remotely.

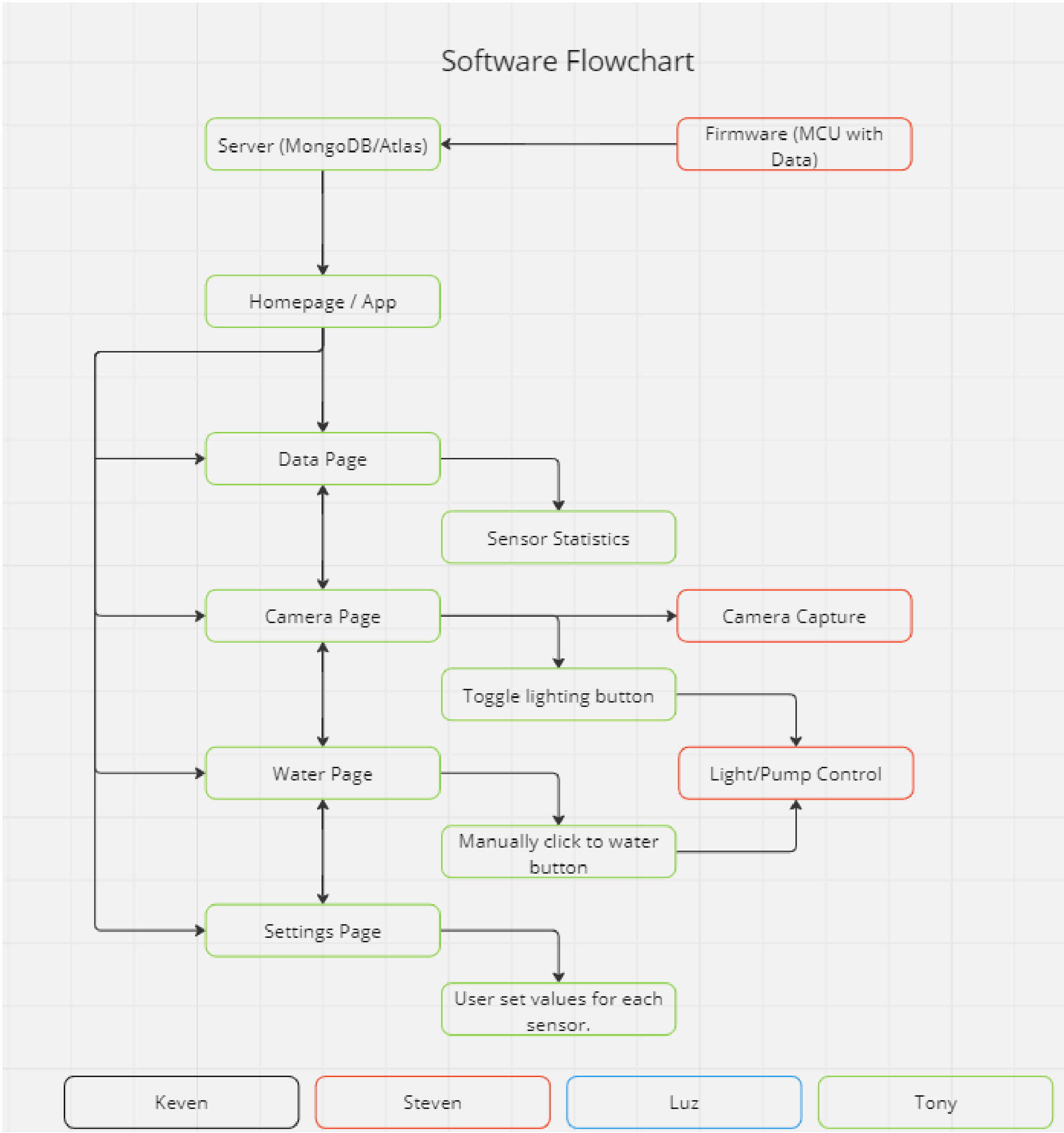


*Figure 2.2: Software Diagram Flowchart*

## 2.10 House of Quality Table

The House of Quality (HoQ) was a crucial tool in the development of our project as it helped translate customer needs into specific technical requirements. By using HoQ, we systematically identified and prioritized customer desires such as ease of use, reliability, and water efficiency, and then aligned these with design specifications like sensor accuracy, battery life, and material durability. This structured approach ensured that the final product not only met but exceeded user expectations, enhancing customer satisfaction and market competitiveness. Additionally, the HoQ facilitated cross-functional collaboration, ensuring that every team member focused on the critical features that drove the project's success.

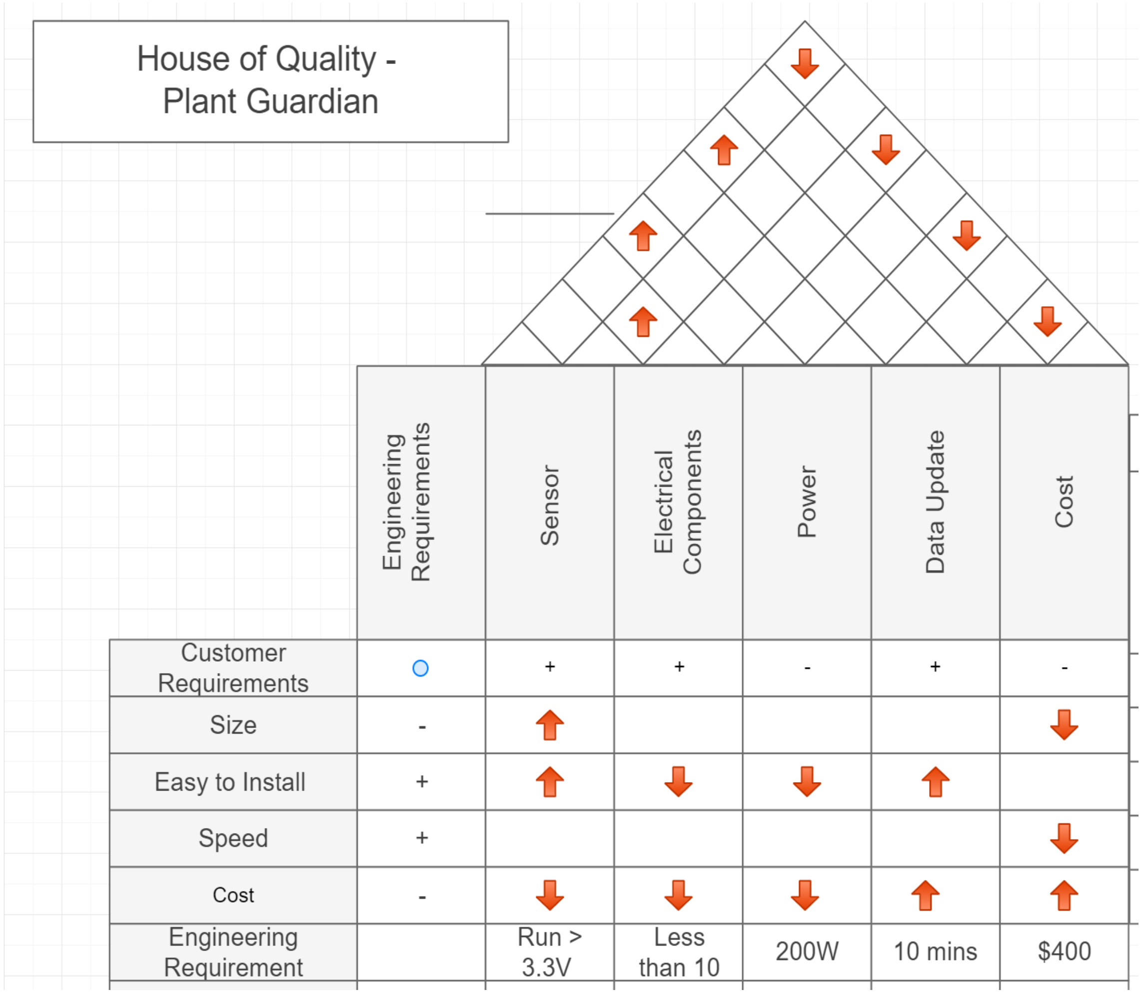



*Figure 2.3: House of Quality Table*

*Table 2.4: Description of House of Quality*

| Symbol | Description |
|---|---|
| ↑ | Positive Correlation |
| ↑↑ | Strong Positive Correlation |
| ↓ | Negative Correlation |
| ↓↓ | Strong Negative Correlation |
| **+** | Positive Polarity |
| **-** | Negative Polarity |

# 3. Research related to Project definitions

## 3.1 Lighting Option

### Why is plant lighting important?

Lighting is crucial for indoor plant health because it powers photosynthesis, the process through which plants convert light, carbon dioxide, and water into energy. This energy is essential for growth, blooming, and seed production. Without adequate light, plants cannot generate the necessary carbohydrates, leading to energy depletion and potentially plant death. Insufficient light causes plants to turn pale, grow leggy with elongated stems, drop leaves, and fail to produce flowers. Conversely, excessive light can scorch and bleach leaves, damaging the plant.

Matching plant light requirements with the natural light available in a home or office is essential. Low-light plants, such as snake plants and pothos, are suited for dim corners or north-facing windows, where they grow slowly and require careful watering to avoid overwatering. Medium-light plants, including spider plants and peace lilies, need more light and are best placed near east-facing windows or slightly away from west-facing windows, often needing supplemental lighting to flourish. High-light plants, such as succulents and cacti, thrive in bright spots like south- or southwest-facing windows but dry out faster and require more frequent watering.

When natural light is insufficient, artificial lighting can serve as a valuable supplement. LED and fluorescent bulbs are popular choices due to their energy efficiency and ability to provide the full light spectrum that plants need. Incandescent and high-pressure sodium bulbs are less common due to inefficiency and heat production. By carefully assessing indoor light conditions and selecting suitable plants or artificial lighting solutions, a healthy and vibrant indoor garden can be maintained.

### Why use LED Grow Lights?

LED grow lights are preferable for indoor plant cultivation due to their superior energy efficiency, longevity, and low heat emission. Compared to traditional lighting sources like incandescent, fluorescent, and high-intensity discharge (HID) lamps, LEDs consume significantly less energy, reducing both operational costs and carbon footprint. Their long lifespan of up to 50,000 hours minimizes the need for frequent replacements, offering a cost-effective solution over time.

Additionally, LEDs produce minimal heat, allowing them to be placed close to plants without causing thermal damage. This proximity maximizes light absorption and supports optimal plant growth. LEDs also offer the advantage of customizable light spectra, enabling growers to tailor the light wavelengths to specific plant needs. For instance, blue light can enhance chlorophyll production for healthier foliage, while red light can promote flowering. These benefits make LED grow lights a highly efficient and effective choice for both commercial and home gardening applications.

**Why use Fluorescent Lighting?**

Fluorescent lighting is favored in indoor gardening, particularly for seed starting and plant growth, due to several key advantages. Firstly, fluorescent lights provide a broad spectrum of light that closely mimics natural sunlight, which is essential for supporting healthy photosynthesis in plants. This balanced spectrum includes wavelengths across the visible spectrum that plants need for various stages of growth, from seedling development to flowering.

Another significant advantage of fluorescent lights is their affordability and accessibility. They are widely available in several types such as T5, T8, and T12, which refer to the diameter of the tubes. T5 tubes are 5/8 in diameter, T8 1 inch in diameter, and T12 tubes 1.5 inches in diameter. These varied sizes offer flexibility in fitting various fixtures and meeting different lighting needs. Fluorescent lights are more economical upfront compared to newer technologies like LED grow lights, making them a practical choice for home gardeners who are starting out or operating on a budget. This cost-effectiveness, combined with their broad-spectrum light output, makes fluorescent lights a reliable option for promoting healthy plant growth.

Furthermore, fluorescent bulbs are known for their efficiency in converting electrical energy into light rather than heat. This characteristic not only reduces energy costs but also helps maintain a stable temperature around plants, minimizing the risk of overheating or damaging sensitive seedlings.

**Why can’t we use incandescent lights?**

Incandescent lights are not preferable for growing plants compared to fluorescent or LED lights. This is primarily due to their inefficiency in converting electricity into light; about 90% of the energy, they consume is emitted as heat rather than visible light. For indoor plants, especially those with higher light requirements like tropical, succulents, and flowering plants, this excessive heat output can be detrimental, potentially causing heat stress and affecting plant growth negatively. Moreover, incandescent bulbs emit a broad spectrum of light that includes infrared and ultraviolet wavelengths but lacks the specific wavelengths (such as blue and red) that plants need most for photosynthesis and optimal growth. In contrast, fluorescent and LED grow lights are designed to emit light within the ideal spectrum for plants, are more energy-efficient, produce less heat, and have longer lifespans, making them more suitable and effective for indoor gardening and plant growth. Therefore, while incandescent lights may suffice for low-light houseplants in terms of illumination, they are not recommended for providing the necessary light intensity and spectrum required by most indoor plants to thrive.

**Why use High Pressure Sodium Bulb?**

High pressure sodium (HPS) lights offer several advantages for plant growth and care, particularly during the flowering stage. Their spectral output, rich in the orange-red spectrum, is highly beneficial for promoting robust flowering and fruiting in plants, making them ideal for flowering crops such as tomatoes, peppers, and flowering ornamentals. HPS bulbs are also known for their efficiency in converting electricity into visible light, producing up to 140 lumens per watt. This efficiency makes them a cost-effective choice for growers concerned with energy consumption.

Compared to other lighting options discussed, such as fluorescent and LED lights, HPS lights stand out due to their high light intensity and efficacy in the flowering phase. They provide a strong light output that penetrates deeply into plant canopies, promoting healthy growth and maximizing yield during the flowering period. However, HPS lights do emit a significant amount of heat, which requires careful management to prevent heat stress and potential damage to plants. This characteristic necessitates maintaining a sufficient distance between the lights and the plant canopy or implementing effective cooling systems in indoor grow environments.

**Full Spectrum LED grow lights**

Choosing full-spectrum LED lights for our smart planter project is the best decision due to their unparalleled versatility, energy efficiency, and long lifespan. Full-spectrum LEDs provide a comprehensive range of wavelengths that closely mimic natural sunlight, making them suitable for a wide variety of plants and all stages of growth. This adaptability ensures that our smart planter can accommodate everything from low-light houseplants to high-light succulents, supporting healthy photosynthesis and robust plant development. Compared to other lighting options, such as incandescent, fluorescent, or high-pressure sodium bulbs, LEDs consume significantly less energy, generate minimal heat, and last much longer, making them a cost-effective and eco-friendly choice. Their low heat emission also allows for closer placement to plants, maximizing light absorption without the risk of thermal damage.

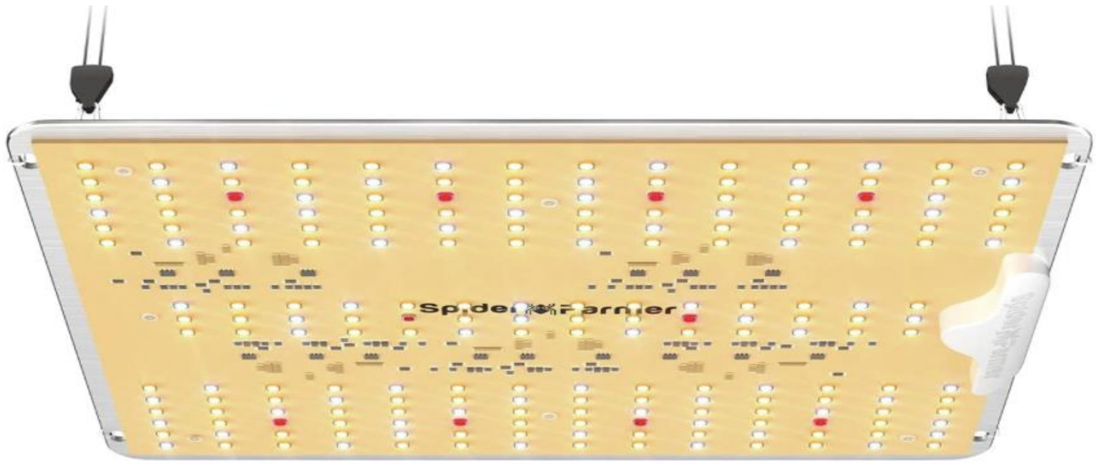
To implement full-spectrum LED lights in our smart planter, we can integrate a programmable LED system that allows users to modify the light spectrum and intensity via a app. This programmability ensures that the lighting conditions can be tailored to the specific needs of different plant species and growth stages, providing optimal care. By incorporating a user-friendly interface, users can easily adjust the light settings based on plant type, growth phase, and environmental conditions, ensuring the plants receive the precise light they need to thrive. This feature not only enhances the user experience but also makes our smart planter a versatile and efficient solution for modern indoor gardening, suitable for both novice and experienced plant enthusiasts.

***VIPARSPECTRA 2024 New V1000 LED Grow Lights***


The VIPARSPECTRA XS Series LED Grow Light stands out as the best option for indoor plant cultivation due to its advanced features and exceptional performance. The V1000 model utilizes a new diode layout and dimmer function, ensuring the most uniform PAR map and deep canopy penetration to maximize plant yields. The dimmer knob allows precise manipulation of brightness levels, tailoring the light to each growing phase for optimal growth. With the latest SMD LED technology, the V1000 delivers higher PAR output and stronger light intensity while maintaining lower running costs and minimal heat emission. This silent, fanless design ensures zero noise and enhances the LED's longevity.

The full-spectrum light consists of 11 red diodes, 207 white diodes at 3000K, 105 white diodes at 5000K, and 1 IR diode at 730nm, mimicking natural sunlight to support all growth stages from seedling to flowering. This comprehensive spectrum promotes balanced growth and healthier plants, leading to higher yields. Consuming only 100W, it rivals traditional 250W HPS/MH lights, making it highly energy efficient. The V1000 is designed for ease of use with plug-and-play functionality, and its excellent customer service ensures a seamless user experience. This combination of innovative technology, energy efficiency, and user-friendly features makes the VIPARSPECTRA XS Series LED Grow Light the best value for the investment, providing a superior indoor gardening solution.

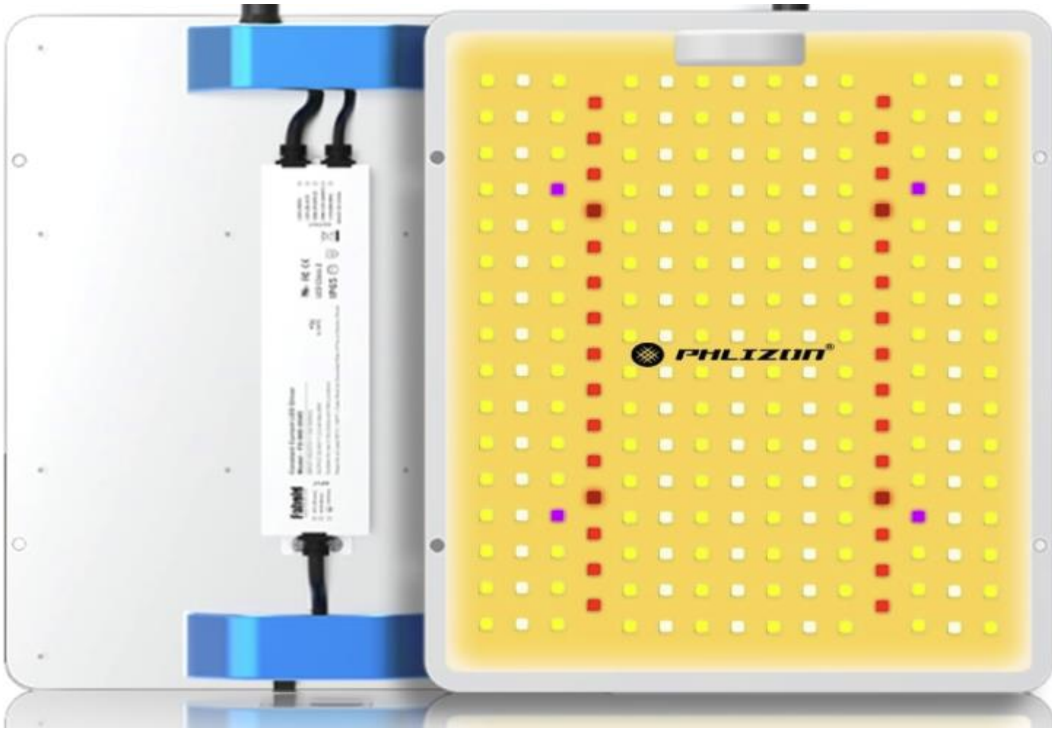
***Spider Farmer 2024 SF1000D LED Grow Lights***




The Spider Farmer SF1000D LED Grow Light is an exceptional choice for indoor gardening, featuring the latest in LED technology and premium Samsung LM301B diodes. This new 2024 version offers powerful light output and uniform canopy penetration, ensuring better light absorption by plants. With a vegetative footprint of 3 x 3 feet and a flowering footprint of 2 x 2 feet, it is perfect for all growth stages. The SF1000D's sunlight spectrum, including blue, full-spectrum 660-665nm, 3000K, 5000K, and IR, is ideal for germination, clones or cuttings, mothers, vegetative, and flowering applications. This comprehensive spectrum helps overcome insufficient natural light, enabling you to grow high-yielding, beautiful plants indoors.

The SF1000D also stands out for its high efficiency and lower cost. Consuming only 100W, it delivers results comparable to 200W HID lights with 50% less heat and power consumption, offering 40% higher yields with less energy. This energy efficiency and better light penetration make it a top choice for indoor growers. Safety and reliability are also paramount, with features such as a fanless design for zero noise, large solid aluminum heat sinks, protective covers for cables, and a waterproof LED board. These quality components extend the light's lifespan by 50% compared to other brands.

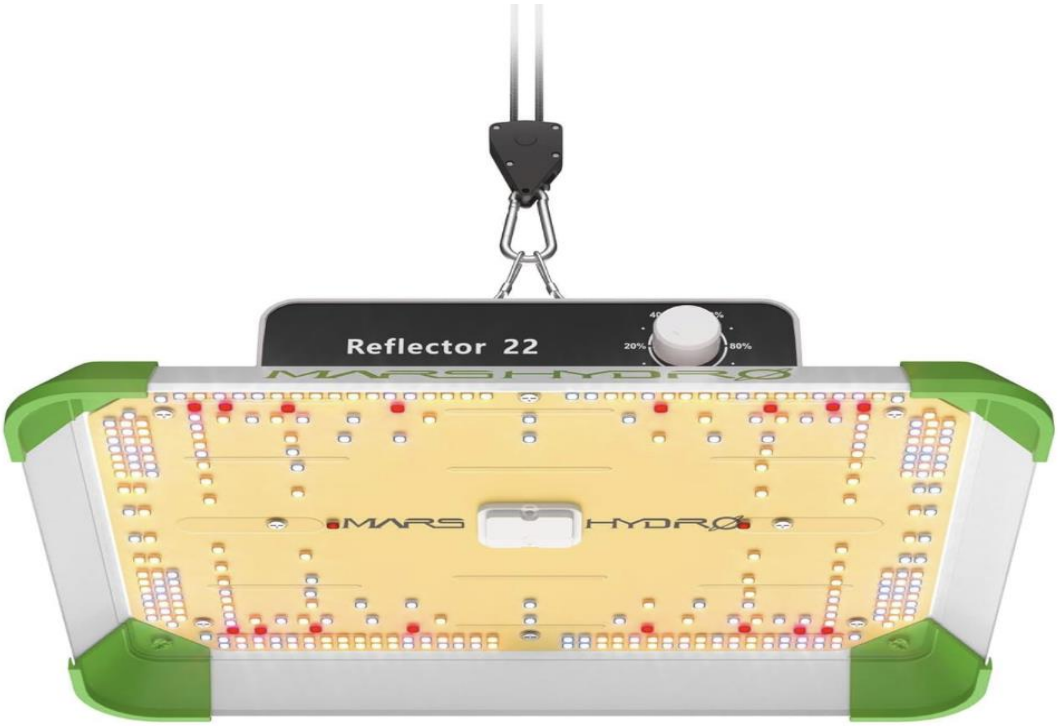
***Phlizon Pro Series LED Grow Light***




The Phlizon Pro Series LED Grow Light is a top-tier option for indoor plant cultivation, offering unbeatable value through high-quality manufacturing in our own factory, ensuring superior quality and reliability. This model features UV and IR LEDs, along with a dimmable design that allows the intensity of light to be adjusted according to the needs of the plants. Equipped with 238 LEDs, including LM281B chips, it provides higher light intensity, less light decay, better energy saving, and enhanced PAR values. The light composition includes 124 pcs 3000K LEDs, 80 pcs 6000K LEDs, 26 pcs 660nm red LEDs, 4 pcs UV LEDs, and 4 pcs IR LEDs, making it versatile for various growth stages.

Phlizon Pro Series stands out with its extra two unique functions: a fanless, noise-free design, and waterproof functionality. The advanced waterproof power supply is installed externally for easy replacement, and the light board's sealing technology ensures the entire unit is water-resistant. Energy efficiency is a highlight, with actual power consumption of only 100 watts and a light efficacy of up to 2.8 umol/W. The light output PPFD value is 500 umol/m² at 18 inches, designed to replace traditional HPS lights while maintaining lower temperatures and higher efficiency. This leads to a 30% higher yield compared to HPS lights. Ideal for a 2x4ft growing area, it delivers higher intensity in a Phlizon mylar grow tent. Supported by a professional LED light manufacturer with a strong R&D team, Phlizon provides excellent service, ensuring customer satisfaction with prompt and thorough responses to any issues.

***MARS HYDRO RP1000 Led Grow Light***




The Mars Hydro RP1000 Reflector22 LED Grow Light is an exceptional choice for indoor plant cultivation, featuring a patented design with a unique 120° angle reflector. This design allows for wider coverage and deeper canopy penetration, ensuring plants receive optimal light for maximum yield. The new diode layout ensures a more even distribution of PPFD, enhancing light absorption for healthier, more productive plants. The full spectrum of this grow light includes 660-665nm red light, 730nm IR light, 2800-3000K warm white light, and 4800-5000K white light, making it suitable for all stages of plant growth.

The RP1000 Reflector22 also offers a dimming daisy chain feature, with independent dimming buttons that enable 0 to 100% brightness adjustment based on the specific growth stage of your plants. The AC daisy chain function allows for easy connection of multiple grow lamps, supporting up to 6 lights for scalable growing setups. Advanced cooling technology, including advanced powder coating and a removable driver, ensures the light stays cool even in compact grow tents like 2x2 or 3x3. The unique rounded corner design and double-layer shielding enhance safety and sturdiness. Consuming only 130W, the Mars Hydro RP1000 yields higher output with lower power consumption. Its non-isolated power sources provide faster energy conversion, resulting in a 5% energy savings, and its plug-and-play design ensures easy installation.

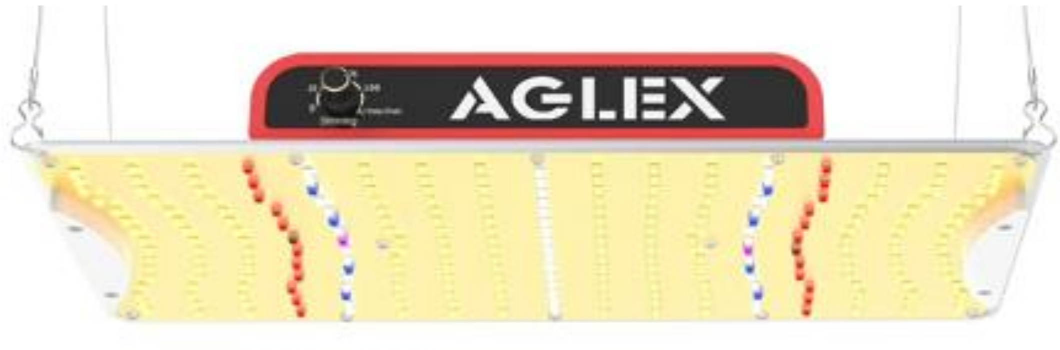
### *Horticulture Lighting Group (HLG) Quantum Board LED Grow Light*


The Horticulture Lighting Group (HLG) Quantum Board LED Grow Light is a premier choice for indoor gardening, renowned for its high efficiency and superior light output. The Quantum Board design features a unique and innovative layout that ensures a more even distribution of PPFD, aiding plants in optimal light absorption and maximizing yields. The full-spectrum light includes a blend of 660nm red light, 730nm IR light, 3000K warm white light, and 5000K white light, making it suitable for all growth stages from germination to flowering.

The HLG Quantum Board LED Grow Light also incorporates a dimming feature, allowing growers to adjust brightness levels from 0 to 100% to match the specific needs of plants at different growth stages. This flexibility ensures that plants receive the right amount of light intensity at every phase, promoting healthier growth and higher yields. Advanced cooling technology, including a passively cooled design, ensures the lights remain cool without the need for noisy fans. The robust construction and high-quality components of the Quantum Board contribute to its long lifespan and reliability, making it a dependable choice for serious growers.

Designed to consume significantly less power than traditional HPS/MH lights while providing higher light intensity, the HLG Quantum Board LED Grow Light is both energy-efficient and cost-effective. Its plug-and-play setup makes installation straightforward, allowing growers to quickly integrate it into their grow spaces. With its superior performance, efficiency, and ease of use, the HLG Quantum Board LED Grow Light is an excellent investment for indoor plant cultivation, ensuring robust plant growth and maximized yields.

**AGLEX K1000 LED Grow Light**




When comparing various LED grow lights for an indoor smart planter project, the AGLEX K1000 LED Grow Light emerges as a top-tier option, demonstrating several compelling advantages over its competitors. This light boasts a high Photosynthetically Active Radiation (PAR) output, ensuring efficient photosynthesis across all plant growth stages. Its full spectrum design, incorporating red, blue, white, and infrared LEDs, effectively mimics natural sunlight, which is crucial for promoting vigorous plant growth, flowering, and fruiting. The AGLEX K1000 also features a daisy chain function, enabling multiple units to be connected and controlled simultaneously, making it an excellent choice for larger growing setups. Its energy efficiency is comparable to other high-quality models such as the VIPARSPECTRA 2024 V1000 and Spider Farmer 2024 SF1000D, utilizing advanced diodes that deliver more light while consuming less power.

Moreover, the AGLEX K1000 is equipped with a powerful cooling system that includes a robust fan and heat sink, ensuring the unit remains cool and prolonging its operational lifespan. This active cooling system is particularly effective compared to the passive cooling found in some models, such as the Phlizon Pro Series, which, while quieter, might not manage heat as efficiently. The Phlizon Pro Series does offer the added advantage of waterproofing, which is beneficial in certain environments, but this feature is not emphasized in the AGLEX K1000.

In terms of spectrum and energy efficiency, the AGLEX K1000 holds its own against both the VIPARSPECTRA V1000 and the Spider Farmer SF1000D, with a slight edge due to its inclusion of infrared LEDs, which are beneficial during the flowering stage. The AGLEX K1000's active cooling system also provides superior temperature control, potentially enhancing the light's durability and performance. For those prioritizing robust, versatile, and efficient lighting solutions in a smart planter project, the AGLEX K1000 LED Grow Light represents an outstanding choice, combining high performance with flexibility and reliability.

**LED Light Bulbs**

When choosing lighting for an indoor smart planter, the decision between specialized grow lights and regular LED light bulbs is crucial for optimizing plant health and system efficiency. Although specialized grow lights are designed specifically for horticultural purposes, regular LED light bulbs offer several compelling advantages that make them a preferred choice for many indoor gardening enthusiasts.

Firstly, regular LED light bulbs are highly cost-effective. They are generally less expensive than specialized grow lights, both in terms of initial purchase and long-term energy use. LEDs are renowned for their energy efficiency, consuming significantly less power compared to traditional lighting solutions. This efficiency translates into lower electricity bills, making regular LEDs an economical option for those looking to maintain a budget while still providing adequate lighting for their indoor plants. Moreover, LEDs have a long lifespan, reducing the frequency and cost of replacements.

Another significant advantage of regular LED bulbs is their ease of installation. Unlike specialized grow lights that often require specific fixtures or mounting setups, standard LED bulbs can be used with existing light sockets. This compatibility allows for straightforward and quick installation, eliminating the need for additional hardware or complex setup procedures. This simplicity not only saves time but also reduces installation costs and potential technical difficulties.

In terms of performance, modern LED bulbs are designed to offer a broad spectrum of light that can support various stages of plant growth. While they may not provide the exact spectrum or intensity of specialized grow lights, many regular LEDs are engineered to emit light across a range of wavelengths that are beneficial for plant health. This makes them suitable for various indoor plants, from leafy greens to flowering varieties. Additionally, LEDs produce minimal heat compared to other lighting options, which helps maintain a stable environment and reduces the risk of heat stress on plants. This characteristic allows them to be placed closer to plant canopies without causing damage, further enhancing their effectiveness.

Overall, regular LED light bulbs offer a practical, cost-effective, and versatile alternative to specialized grow lights. Their affordability, ease of installation, energy efficiency, and broad-spectrum light make them an excellent choice for indoor gardening. By opting for regular LEDs, you can create an efficient and aesthetically pleasing indoor garden while keeping operational costs low and ensuring optimal plant growth.

Opting for regular LED light bulbs as the final choice for our indoor smart planter system is the most optimal solution for both designers and consumers due to their exceptional combination of cost-efficiency, ease of installation, and versatility. LED bulbs are widely available and less expensive than specialized grow lights, making them a budget-friendly option that aligns with cost-conscious design principles. Their broad compatibility with standard light sockets simplifies the installation process, reducing setup complexity and associated labor costs. Additionally, modern LED bulbs are designed to emit a spectrum of light that supports various stages of plant growth, offering adequate lighting for a range of indoor plants while producing minimal heat to prevent plant stress. This balance of affordability, simplicity, and effective performance ensures that the final product is both user-friendly and economically viable, providing an accessible solution for consumers looking to cultivate healthy plants in an indoor environment.

***LED Strip Lighting***

In the Plant Guardian project, the use of a Commercial Electric LED strip light offers significant advantages over other lighting options. This LED strip light kit, which includes four 12-inch linkable segments and operates at just 8.5 watts, delivers energy-efficient, low-voltage lighting ideal for indoor plant care. Unlike standard LED bulbs, LED strip lights are more flexible in installation, allowing customized placement around plants to ensure even light distribution and improved coverage for optimal growth. The strip’s modular design, coupled with warm white lighting, replicates a natural light spectrum suitable for indoor environments, promoting healthy photosynthesis without excessive heat. Additionally, the kit’s simple plug-and-play configuration and included mounting hardware simplify integration and reduce setup time, making it an ideal solution for the Plant Guardian project’s goals of low energy consumption, easy installation, and enhanced plant vitality.

*Table 3.1 Growth Light Comparison*

| | VIPAR | Spider Farmer | Philizon | Mars Hydro | HLG | AGLEX | Light Bulb |
|---|---|---|---|---|---|---|---|
| Product Dimensions | 11.7”D - 10.4”W - 2.2”H | 12.4”D – 11.02” W – 2.12”H | 13”L – 11”W – 2.4”H | 16”L – 13.3”W – 4.9”H | 11”L – 2”W – 12.8”H | 20”D – 30”W – 40”H | 2.36”W – 3.94”H |
| Weight (lb) | 3.9 | 3.2 | 3.3 | 5.77 | 3.8 | 3.3 | 0.46 |
| Power (W) | 100 | 100 | 100 | 130 | 95 | 100 | 60 |
| Voltage (V) | 120 | 230 | 85 | 110 | 90 | 277 | n/a |
| Manufacturer | VIPAR SPECTRA | Spider Farmer | Phlizon | Mars Hydro | HLG | AGLEX | Linkind |
| Price ($) | 49.99 | 89.99 | 59.99 | 119.99 | 149.00 | 69.99 | 13.99 |

## 3.2 Water Pump Options

For a smart planter project with a 1-gallon water reservoir, the ideal choice for a water pump would be a submersible pump designed specifically for small-scale irrigation applications. These pumps are designed to operate efficiently when submerged in water, ensuring effective water circulation within the planter system without the risk of leaks. It is recommended to select a pump with an adjustable flow rate typically ranging from 50 to 200 liters per hour (LPH) or 13 to 53 gallons per hour (GPH), which suits the watering requirements of indoor smart planters. When choosing a pump, factors such as power requirements (commonly in low voltages like 12V or 24V DC), compact size for easy installation inside the reservoir, and additional features like suction cups or mounting brackets should be considered. Brands like Jebao, Homasy, and VIVOSUN offer reliable options with various specifications suitable for different project needs.

### Creating our own Pump using Fusion360

Creating a water pump using Fusion 360 for design, 3D printing for fabrication, and integrating a DC motor offers a customizable solution for various projects requiring a 1-gallon water reservoir. Fusion 360 facilitates precise digital modeling of pump components such as housing and impeller, tailored to specific project requirements. 3D printing allows for manufacturing these components with materials like ABS or PLA, ensuring durability and water resistance. DC motors, chosen based on voltage, speed, and torque requirements, are integrated into the design to provide controlled water pumping. This approach enables efficient assembly and testing of the pump's functionality, supporting customization and learning opportunities in design and electronics integration.

***4Pcs DC 3-5V Micro Submersible Mini Water Pump***

**SPECS:**

*Table 3.2: Specifications of the pump*

| | |
|---|---|
| **Product Dimension** | 3.94 X 3.15 X 0.39 inches |
| **Power Source** | Corded Electric |
| **Item weight** | 5.6 ounces |
| **Max Flow Rate** | 120 Liters per Hour |
| **Voltage** | 4.5V |
| **Amperage** | 250mA |
| **Manufacturer** | SIPYTOPF |
| **Price ($)** | 9.99 |

The DC 3-5V Micro Submersible Mini Water Pump is an excellent addition to our smart planter project, offering efficient and reliable water management for indoor plants. With its rated voltage of DC 3V or 4.5V and current range of 150-250mA, this pump is designed to deliver consistent water flow while maintaining low energy consumption. Its quiet design ensures that the pump operates without producing excessive noise, making it ideal for indoor environments where a peaceful ambiance is preferred. The compact size of the pump allows for easy installation and removal, fitting seamlessly within the 1-gallon reservoir of our smart planter, which is essential for space-saving and maintaining a clean aesthetic.

Implementing this micro submersible pump in our smart planter will significantly enhance its functionality by providing automated watering, ensuring that plants receive the right amount of water precisely when needed. The pump can be controlled via the smart planter's integrated sensors and app, allowing users to set watering schedules or trigger watering remotely based on real-time soil moisture levels. This automation not only takes the guesswork out of plant care but also prevents common issues like overwatering or underwatering, promoting healthier plant growth. The pump's wide applicability means it is versatile enough for various water features, making it a robust and reliable component for our hydroponic system. Overall, the DC 3-5V Micro Submersible Mini Water Pump is the best option for our project due to its energy efficiency, quiet operation, compact design, and ease of integration, ensuring our smart planter delivers optimal performance and convenience to users.

The water pump is an essential component of our smart water planter, playing a pivotal role in maintaining optimal hydration for the plants. Integrating the water pump into the system involves several steps, each aimed at ensuring reliability, efficiency, and ease of use. Our chosen water pump can be controlled manually via Bluetooth or Wi-Fi, allowing users to activate it remotely through a application. This flexibility is especially beneficial for users who may not always be present to manually water their plants. By utilizing sensors such as moisture, temperature, pH, and water level, the pump can automate watering processes based on real-time data, ensuring precise water delivery and preventing both overwatering and underwatering.

To integrate the water pump effectively, we will connect it to a microcontroller that receives input from the sensors. The microcontroller will process this data and control the pump's operation accordingly. This setup ensures that the pump activates only when necessary, conserving water and promoting plant health. Additionally, the water level sensor in the reservoir will monitor the water supply, preventing the pump from running dry and potentially damaging itself. This automated system will not only make plant care more efficient but also enhance user convenience by reducing the need for constant manual intervention.

When selecting a water pump, several factors must be considered, including flow rate, power consumption, noise level, and durability. For our project, we compared various pumps to identify the best option. Submersible pumps, for instance, are known for their quiet operation and efficient water delivery, making them ideal for indoor gardening. They are also relatively easy to install and maintain. In contrast, diaphragm pumps, which use a reciprocating diaphragm to move water, offer precise control over water flow but can be noisier and more complex to install. Another option is peristaltic pumps, which are highly accurate and can handle a variety of liquids, including nutrient solutions, but they tend to be more expensive and require more maintenance.

While 3D printing technology has significantly advanced, creating a water pump from scratch using this method presents several challenges and disadvantages. Initially, we explored designing and manufacturing a custom water pump using Fusion 360 and 3D printing. However, we encountered several obstacles that led us to abandon this approach in favor of commercially available pumps.

One major challenge is the complexity and precision required for a functional water pump. Designing a pump involves intricate mechanical components that must fit together with high precision to ensure efficient operation and prevent leaks. Achieving this level of precision with 3D printing can be challenging, especially for components like impellers and seals that require tight tolerances and smooth finishes. Even slight imperfections in the 3D-printed parts can lead to inefficiencies and potential failures in the pump's operation.

Material limitations also pose a significant hurdle. Most 3D printers use plastic materials, which may not be suitable for all parts of a water pump. Components that are constantly in contact with water need to be resistant to wear and corrosion. While some high-end 3D printers can use materials like resin or metal, these options are often more expensive and less accessible. Moreover, the material properties of 3D-printed plastics might not match those of industrial-grade materials used in commercial pumps, potentially compromising the pump's durability and performance.

Durability and reliability are critical factors in the design of any water pump. Commercially available pumps are typically designed and tested for long-term reliability and durability. In contrast, a 3D-printed pump might lack the robustness needed for continuous operation, leading to potential failures and increased maintenance requirements. The stress and wear experienced by pump components, such as the impeller, can quickly degrade 3D-printed parts, especially if they are not made from the highest quality materials.

Cost and time considerations also play a role in the decision to avoid 3D printing a custom water pump. Designing, prototyping, and testing a custom pump can be time-consuming and costly. Iterative design changes, multiple print attempts, and troubleshooting mechanical issues can add significant delays to the project timeline. In contrast, purchasing a pre-built pump is more cost-effective and ensures that the pump meets industry standards, allowing us to focus resources on other critical aspects of the project.

Finally, maintenance and repairs of a custom 3D-printed pump would require specialized knowledge and skills. If a component fails, it may be challenging to replace it quickly, potentially leading to extended downtime. Commercial pumps, on the other hand, come with customer support and readily available replacement parts, ensuring that any issues can be resolved promptly.

Given these considerations, integrating a commercially available water pump into our smart water planter is the most practical and reliable solution. This approach allows us to leverage the expertise and quality assurance of established manufacturers, ensuring that our system is both efficient and durable. By focusing on integrating and optimizing existing components, we can deliver a high-quality product that meets the needs of our users while avoiding the pitfalls associated with custom 3D-printed solutions.

**VIVOSUN**

When selecting water pumps for small indoor irrigation systems, it’s essential to balance flow rate, power consumption, and size. The VIVOSUN 80 GPH Submersible Water Pump is a strong alternative with a max flow rate of 300 liters per hour (80 GPH). It operates at 4 watts and has a compact size of 1.8 x 1.6 x 1.2 inches, making it easy to integrate into your smart planter design. Its adjustable flow rate allows precise water control, which is beneficial for managing the hydration of plants. Despite its relatively higher flow rate, its quiet operation and ease of installation make it suitable for both small and large planter systems.

**Homasy**

Another excellent option is the Homasy 80 GPH Submersible Water Pump, which is known for its low energy consumption, using only 2 watts. Its compact dimensions of 1.5 x 1.5 x 1.1 inches make it one of the smallest options available, making it ideal for projects where space is a limiting factor. Like the VIVOSUN, it also has an adjustable flow rate and operates quietly, but its lower power consumption gives it an edge in energy efficiency. This makes it an ideal choice for projects focusing on sustainability and energy conservation.

In comparison to your original pump, which operates at 120 liters per hour with around 4.5V and 250mA, the VIVOSUN and Homasy both offer significantly higher flow rates but provide greater flexibility due to their adjustable flow settings. The original pump is sufficient for consistent, low-power watering, but these alternatives can handle larger-scale systems while maintaining energy efficiency. Both options provide more customization with regard to watering intensity, making them versatile for a range of plant types and environments.

## 3.3 Power Supply Options

### Direct Power via AC Adapter

Using an AC adapter to power the smart planter is a practical solution for indoor environments with readily accessible power outlets. This method provides a stable and continuous power supply, ensuring that all components, including LED lights, water pumps, and sensors, operate efficiently without interruption. It supports high power requirements, making it ideal for setups with multiple or power-intensive components. However, this method necessitates the planter's proximity to an electrical outlet, which might limit placement options. Also, careful design considerations are needed to safely manage water and electricity, preventing potential hazards.

### Battery Power

Battery power, whether rechargeable or disposable, offers portability and flexibility for smart planters. Rechargeable batteries are eco-friendly and cost-effective overall, providing a reusable power source that reduces waste. They are particularly useful as a backup during power outages. Disposable batteries, on the other hand, offer convenience for short-term use without the need for recharging. However, both types require regular monitoring to ensure continuous operation, with rechargeable batteries needing periodic recharging and disposables needing replacements. High-capacity batteries may be necessary for continuous operation, especially for power-hungry components.

### Solar Power

Solar power presents an environmentally sustainable option for powering the smart planter, harnessing natural sunlight to generate electricity. This method is ideal for outdoor or well-lit indoor areas, significantly reducing electricity costs and carbon footprint. Solar panels can be paired with rechargeable batteries to store excess energy for use during low sunlight conditions, ensuring consistent power supply. However, the efficiency of solar power is heavily dependent on sunlight availability, necessitating a larger solar panel setup in less sunny regions. The initial installation cost can also be higher compared to other power options due to the need for solar panels and related infrastructure.

### USB Power

USB power provides a versatile and convenient solution, leveraging various USB sources such as power banks, computers, and USB wall adapters. This method is portable and easy to connect, making it suitable for small, low-power devices within the smart planter. It's an excellent option for those who need a flexible power source that can be moved easily. However, USB power is limited by its capacity and may not be sufficient for high-power components like large LED grow lights. It requires access to USB ports or compatible adapters, which might not always be available or convenient depending on the planter's location.

### Hybrid Power Solutions

Hybrid power solutions combine the strengths of solar and battery power to create a reliable and sustainable power system. By utilizing solar energy and storing it in rechargeable batteries, this method ensures that the smart planter has a consistent power supply even during periods of low sunlight. This approach reduces dependence on external power sources and can lower operational costs over time. However, hybrid systems are more complex to set up and have higher initial costs due to the need for both solar panels and battery storage. Efficient energy management systems are required to balance solar input and battery usage, ensuring optimal performance and longevity of the power supply.

### Power Source for this project

For an indoor project with an average power requirement of 200W that needs to run continuously, the best power source would be a direct power supply via an AC adapter. This method provides a stable and uninterrupted power source, crucial for maintaining the health and growth of the plants and the consistent operation of the system's components. An AC adapter can easily meet the 200W demand, unlike batteries or solar power systems, which would be significantly more expensive and complex to implement for continuous indoor use. Additionally, using an AC power supply is more economical eventually compared to battery systems that require regular recharging or replacement. The setup is straightforward, involving simply plugging the device into a standard electrical outlet, making it a plug-and-play solution. For enhanced safety, ensure proper insulation and waterproofing to prevent electrical hazards, and consider incorporating a UPS (Uninterruptible Power Supply) to provide temporary power during brief outages. Overall, an AC adapter offers the most practical, reliable, and cost-effective solution for powering the smart planter continuously.

**USB – A to Micro USB**

Using USB-A as the power source for our indoor smart planter system is an optimal choice due to its widespread availability, standardization, and ease of integration. USB-A ports are found in a vast array of devices, from power adapters and computers to power banks, making them a convenient and familiar option for consumers. This widespread usage ensures that users can easily power their planter system without needing special adapters or power supplies, enhancing user convenience and accessibility. Moreover, the standard 5V output of USB-A is well-suited for powering low-power electronics, making it an excellent match for the needs of our smart planter system, which includes components like LED lights, sensors, and microcontrollers.

To integrate USB-A into our PCB, several key steps must be followed. First, we need to determine the power requirements of our system, ensuring that the total power consumption does not exceed the USB-A limits of 500mA (USB 2.0) or 900mA (USB 3.0). Selecting a suitable USB-A connector for PCB mounting is crucial, with options including through-hole or surface-mount types. Proper placement and secure mounting on the PCB layout are essential. If our components require different voltage levels, a voltage regulator circuit can step down or stabilize the 5V supply from the USB-A port. For instance, using a linear regulator or a buck converter can provide the necessary voltage for components requiring 3.3V. Additionally, power filtering with capacitors helps maintain a stable power source by filtering out noise and voltage spikes. Implementing current-limiting resistors or protection circuits, such as polypuses or current-limiting ICs, prevents overcurrent situations that could damage the USB port or device. Finally, ensuring that PCB traces connected to the USB-A power lines are appropriately sized to handle the current is crucial for reducing resistance and improving reliability. By following these steps, we can effectively integrate USB-A as a power source into our PCB, providing a reliable, user-friendly, and widely compatible power solution for our indoor smart planter system.

**Barrel Jack**

Using a barrel jack as the power source for our indoor smart planter system offers several advantages, making it an optimal solution for both designers and consumers. Barrel jacks are widely used in various electronic devices due to their simplicity, robustness, and ability to handle a wide range of voltages and currents. This flexibility allows for the use of different power adapters, catering to specific power requirements and enhancing the system's adaptability. For consumers, barrel jacks provide a reliable and straightforward connection that minimizes the risk of incorrect plugging or disconnection. This user-friendly design ensures a secure power connection, reducing potential issues related to power interruptions.

Integrating a barrel jack into our PCB involves several crucial steps to ensure efficient and reliable operation. First, we need to determine the voltage and current requirements of our smart planter system and select an appropriate barrel jack that matches these specifications. Standard barrel jacks are available in many sizes and ratings, making it essential to choose one that can handle the system's power needs without overheating or failing. Proper placement and secure mounting of the barrel jack on the PCB are vital to ensure durability and stability, especially for a device that might be moved or adjusted frequently.

To provide a stable and consistent power supply to the components, a voltage regulation circuit may be necessary. This circuit, which can include linear regulators or buck converters, ensures that the voltage delivered to the sensitive electronics remains within the specified range, even if the input voltage from the power adapter varies. Power filtering using capacitors is also important to smooth out any voltage spikes or noise, protecting the components from potential damage. Additionally, implementing overcurrent and overvoltage protection circuits, such as fuses or transient voltage suppression diodes, can safeguard the system against power surges or accidental incorrect adapter usage.

Finally, ensuring that PCB traces connected to the barrel jack are appropriately sized to handle the expected current is crucial for maintaining reliability and efficiency. By following these steps, we can effectively integrate a barrel jack as the power source for our indoor smart planter system, providing a robust, flexible, and user-friendly power solution. This approach not only enhances the system's overall performance but also meets the needs of both designers and consumers by offering a reliable and adaptable power connection.

### Final Decision

In the development of our indoor smart planter system, we have decided to utilize LED strip lights, a water pump, and a barrel connector as the power source. This combination was chosen based on a comprehensive evaluation of efficiency, user convenience, and overall system performance.

Our choice to use LED strip lights over full-spectrum grow lights or LED bulbs offers several advantages. LED strip lights are highly energy-efficient, using only 8.5 watts for the full 48-inch length, which is essential for an environmentally friendly and cost-effective product. The low power consumption reduces operational costs and supports sustainability goals. LED strip lights also have a long lifespan, minimizing the need for frequent replacements and reducing maintenance efforts and costs. Additionally, LED strip lights offer flexible installation options that allow customized placement around plants, ensuring even light distribution for effective photosynthesis and growth. This flexibility in arrangement is an advantage over LED bulbs, as it provides greater control over lighting intensity and coverage while still being easy to integrate into the system. The straightforward plug-and-play configuration of LED strip lights further simplifies the design process, making them a practical, efficient choice for the Plant Guardian system.

The inclusion of a water pump in the smart planter system is crucial for maintaining optimal moisture levels in the soil. An automated water pump, controlled by soil moisture sensors, ensures that plants receive the right amount of water precisely when they need it. This automated irrigation system eliminates the risk of overwatering or underwatering, which are common issues in manual plant care. The water pump is integrated with a control system that can be programmed to adjust watering schedules based on the specific needs of different plants. This customization enhances plant health and growth by providing consistent and appropriate moisture levels. Additionally, an automated system saves users time and effort, making plant care more convenient and reliable.

Using a barrel connector as the power source offers significant benefits in terms of reliability and ease of use. Barrel connectors are known for their robustness and secure connections, which are essential for maintaining continuous power supply to the system. They can handle a wide range of voltages and currents, providing the necessary flexibility to power all components effectively. The barrel connector's design ensures a stable and reliable power connection, reducing the risk of accidental disconnections or incorrect plugging. This reliability is crucial for a system that relies on consistent power for its automated functions, ensuring that the LED bulbs and water pump operate smoothly and efficiently.

To integrate the LED bulbs, water pump, and barrel connector into a cohesive smart planter system, careful design and planning are essential. The barrel connector will be mounted on the PCB and connected to the power input, providing a stable and reliable power supply. Voltage regulation circuits will ensure that all components receive the correct voltage. Standard bulb sockets will be included in the design, allowing easy installation and replacement of LED bulbs. A driver circuit will be implemented to control the LED bulbs, allowing for adjustable brightness and timing. This will be managed by a microcontroller to facilitate programmable lighting schedules tailored to plant needs.

The water pump will be controlled by a dedicated driver circuit, interfacing with soil moisture sensors to automate irrigation. The microcontroller will manage watering schedules based on sensor data, and soil moisture sensors will be strategically placed to provide accurate data, ensuring precise control over the watering process. A microcontroller will act as the system's brain, managing the LED bulbs, water pump, and sensors. It will run control algorithms, handle user inputs, and communicate with connected apps or interfaces. An app or web interface will enable users to monitor and control the system remotely, providing a convenient and user-friendly experience.

Protective components such as fuses and transient voltage suppression diodes will safeguard the system against electrical faults, and capacitors will filter out noise and stabilize the voltage supply, ensuring smooth and reliable operation. In conclusion, the combination of LED bulbs, an automated water pump, and a barrel connector as the power source forms a robust, efficient, and user-friendly indoor smart planter system. LED bulbs provide practical and energy-efficient lighting, ensuring optimal plant growth while reducing maintenance efforts. The automated water pump ensures precise irrigation, enhancing plant health and making plant care more convenient. The barrel connector offers a reliable and versatile power solution, simplifying the setup and ensuring continuous operation. By thoughtfully integrating these components, we can create a smart planter system that meets the needs of both designers and consumers, promoting healthier plants and a more enjoyable gardening experience.

## 3.4 Microcontroller options

### 3.4.1 Main Microcontroller

In the development of a plant watering system with integrated video streaming and sensor data transmission to a remote server, the microcontroller will play a pivotal role as the project's central intelligence. This compact computer on a chip will have to manage tasks such as processing sensor inputs for soil moisture levels, control water dispensing mechanisms through GPIO pins, and encode video for streaming via communication interfaces via UART or a wireless communication interface.

The microcontroller's embedded memory will have to store both program instructions for automated watering routines and captured video frames, ensuring seamless operation for the end user. It will have to be programmable in languages like C or Python to utilize timers for periodic sensor readings, implement algorithms to adjust watering based on real-time data, and use wireless communication for transmitting sensor readings and video feeds to the server.

Our system will leverage a microcontroller's capabilities, including its efficient use of power and integration of diverse peripherals. The plant watering system will achieve autonomous operation with enhanced monitoring capabilities. This setup will not only optimize plant care by responding dynamically to environmental conditions but also facilitates remote monitoring and control, allowing our users to oversee plant health and adjust watering schedules remotely through the phone app interface.

Due to having the requirement of streaming video. Our MCU priorities are for a fast processor with enough RAM to both buffer camera data and wireless communications. It also must have enough IO ports to connect our sensors and control surfaces. Finally, we also want to consider power consumption. Although this is secondary to the other requirements.

### 3.4.2 ESP32-S

The ESP32-S (used in the ESP32-CAM dev board) stands out as a robust microcontroller known for its significant RAM capacity (520KB). This ram can be expanded on by external SRAM up to 8MB. This external ram is useful for IOT devices due to wifi and camera data taking up a lot of RAM. It also has two fast Xtensa 32-bit LX6 microprocessors (160Mhz), making it highly capable for a wide range of applications. Its enhanced performance compared to other microcontrollers makes it particularly popular in hobbyist projects, benefiting from a wealth of tools designed for ease of use. The integration of the ESP32-S with the Arduino IDE, ESP-IDE, and Platform.IO allows for a lot of options for development environments. The ESP32-S also supports a variety of programming languages such as C/C++/Micropython/Rust/Lua. Which gives flexibility for development teams switching to the platform due to not having to learn an entirely knew language.

Moreover, the ESP32-S supports FreeRTOS, an open-source real-time operating system that enhances its multitasking capabilities and reliability. This feature is instrumental in managing concurrent tasks efficiently, ensuring precise timing and response in applications requiring real-time operation. As a result, the ESP32-S is not only versatile but also well-suited for projects demanding robust performance, multitasking capabilities, and seamless integration with existing development environments and tools.

Another benefit of the ESP32-S is there is a dev board called the ESP32-CAM that integrates the external flash with a camera connector and a daughter board for flashing allowing for easy prototyping. Also the GPIO/Wifi/Bluetooth is all built into the microcontroller. As well as greater familiarity due to members of the team having experience in C as well as having used the ESP32 and freeRTOS in previous classes.

### 3.4.3 STM32H747XI

The STM32H747XI, the microcontroller powering the Arduino Giga, represents Arduino's most powerful offering yet. Equipped with 1MB of RAM and a 480 MHz CPU, it stands capable of managing complex tasks like WiFi, Bluetooth, and camera interfacing, utilizing external chips for enhanced functionality. One of its primary advantages is its extensive array of available IO ports, facilitating versatile connectivity options for various peripherals and sensors.

Moreover, the STM32H747XI leverages the extensive Arduino library ecosystem, tailored specifically for the Arduino Giga platform. This integration grants developers access to a wide range of pre-built functions, making development more efficient and reducing time spent on low-level programming tasks. Additionally, the microcontroller benefits from comprehensive debugging tools inherent to the Arduino IDE, ensuring robust development and troubleshooting capabilities throughout the project lifecycle. This combination of high performance, abundant IO, and Arduino support positions the STM32H747XI as a formidable choice for projects demanding advanced connectivity and processing capabilities, from IoT applications to sophisticated embedded systems

One downside of the STM32H747XI is that wifi and bluetooth would require a more complicated PCB design. It also suffers from no team member having familiarity with this microcontroller which can greatly slow down development time in order to spend time gaining experience.

### 3.4.4 MSP430

The MSP430 microcontroller features a 16 MHz CPU and 4KB of RAM, notable for its simplicity as it does not operate with an operating system. While it may not match higher-powered chips in raw processing capability, the MSP430 excels in applications requiring low power consumption, making it ideal for battery-operated devices and other energy-efficient designs.

One of the advantages of the MSP430 for our development team is that our team has a lot of familiarity with it due to past experience using it. This also means having access to more help and debugging tools.

While the MSP430 comes with a decent amount of IO options, it does not come with wifi or bluetooth. Also, due to major RAM and CPU limitations would not be able to utilize wifi or bluetooth if added with an additional chip.

### 3.4.5 Raspberry Pi 0 W

The Raspberry Pi Zero W features an Arm Cortex-A53 CPU combined with the RP3A0 system-in-package, incorporating integrated WiFi and additional RAM. With its 1GHz CPU and 512MB of RAM, it blurs the distinction between a traditional microcontroller and a single-board computer (SBC). This configuration offers significant computational power and versatility, suitable for diverse applications ranging from IoT projects to multimedia applications. It also benefits from the Raspberry pi ecosystem which is very well maintained and easy to learn due to its popularity with hobbyists.

However, one notable drawback of the Raspberry Pi Zero W is its higher power consumption compared to simpler microcontrollers. This can be a concern for battery-operated applications or projects requiring energy efficiency. Moreover, integrating the Raspberry Pi Zero W into custom PCB designs entails incorporating all necessary components from the development board, adding complexity to the design and potentially increasing production cost/time. In recent history raspberry pi has had unreliable availability which could affect prototyping both the system and PCB.

Despite these considerations, the Raspberry Pi Zero W remains popular for projects requiring robust processing capabilities in a compact form factor. Its compatibility with a wide range of software tools and libraries, coupled with its community support, makes it an attractive choice for developers.

**Comparison**

*Table 3.3: Main MCU comparison*

| Specification | ESP32-S | STM32H747XI | MSP430 | Raspberry Pi 0 w |
|---|---|---|---|---|
| RAM | Internal 520KB External SRAM 4MB SRAM | 1MB | 4KB | 512MB |
| CPU frequency | 160Mhz | 480 Mhz | 16Mhz | 1ghz |
| bluetooth | 4.2 BR/EDR and BLE | N/A | N/A | Bluetooth 4.2, BLE |
| Operating system | freeRTOS | N/A | N/A | Raspian (Linux) |
| Dimensio | 10x11 | 14 x | 24.3x | 65x30 |

| | | | | |
|---|---|---|---|---|
| ns (mm) | | 14 | 9.4 | |
| Cost | $3 | $20 | $3 | $15 |
| Manufacturer | Expressif | STMicroelectronics | Texas Instruments | Raspberry Pi Foundation |

### 3.4.6 MCU Selection

After evaluating various microcontrollers, it became evident that not all were equally suitable for our project's specific requirements. Initially considering the MSP430 for its broad overview of microcontroller capabilities, we determined it lacked the necessary processing power for streaming images, prompting its exclusion. Similarly, while the Raspberry Pi 0 W could meet our objectives, its excessive power consumption and complex integration requirements for our PCB design made it impractical.

This evaluation led us to focus on the ESP32-S and the STM32H747XI. Ultimately, we opted for the ESP32-S due to its robust performance capabilities tailored precisely to our needs and prototyping options. It is often used by hobbyist for security cameras so the dual 32 bit processors can handle the processing load. While the external SRAM allows us to expand up to 8MB of slower memory, which should be more than enough to handle streaming a webcam and sensor data. It also has a large number of GPIO pins that are enough to handle our plant sensors as well as light and pump. Another major benefit is that it comes pre-packaged with Wifi, Bluetooth, and Bluetooth BLE. This only leaves LORA out of the communication protocols we investigated (which can be added via an external chip via UART if need be).

The ESP32-S's selection was also influenced by its familiarity within our team, ensuring efficient development and troubleshooting processes. Its integrated features and compatibility with Arduino IDE, Vscode via ESP-IDE, and Platform.io provide a comprehensive platform for implementing our project requirements effectively. This decision not only aligns with our technical specifications but also enhances our ability to deliver a reliable and optimized solution for streaming and sensor integration in our application.

### 3.4.7 Secondary Microcontroller

The secondary microcontroller serves a crucial role in our project by managing a diverse range of analog and digital inputs and outputs from our planned peripherals. This pivotal component necessitates careful consideration of several key requirements to ensure seamless integration and efficient operation.

Firstly, UART communication capabilities are essential for facilitating reliable serial communication between the secondary microcontroller and external devices, enabling efficient data exchange and control. Analog-to-Digital Converters (ADCs) play a vital role in converting analog signals from sensors into digital data, providing accurate measurements and feedback crucial for monitoring and decision-making processes.

In addition to UART and ADCs, the availability of digital ports is critical for interfacing with various digital devices and components, supporting robust connectivity and control within our system architecture. Furthermore, support for I2C communication protocol enhances compatibility with a wide range of sensors and peripherals, facilitating streamlined data transmission and synchronization.

### 3.4.8 ATMEGA328

The ATMEGA328 microcontroller stands out prominently for its integral role in powering the Arduino UNO, ensuring seamless compatibility and full support within the Arduino IDE ecosystem. Sporting an 8-bit processor clocked at 20 MHz and equipped with 32 KB of Flash memory, it provides a robust foundation for executing embedded applications with efficiency and reliability. In terms of input-output capabilities, the ATMEGA328 features essential components such as one UART line for serial communication, 8 Analog-to-Digital Converters (ADCs) for precise analog signal processing, 14 digital pins for versatile interfacing with external devices, and support for one I2C interface to facilitate communication with compatible peripherals. Another consideration is the power requirements of the Arduino UNO, which operates on a 5V voltage supply for VCC and employs 5V logic levels on its digital GPIO pins. This standardized voltage level ensures compatibility with a wide range of electronic components and simplifies integration into various hardware configurations. But also can cause issues with compatibility due to 3.3v being a common voltage for primary MCUs. Finally, one of the ATMEGA328's greatest strengths lies in its widespread familiarity among team members, derived from their extensive experience with Arduino-based projects.

### 3.4.9 IMXRT1062DVJ6

The IMXRT1062DVJ6, prominently featured in the Teensy 4.1, stands out for its powerful 32-bit ARM Cortex-M7 processor running at 600 MHz, accompanied by 1 MB of RAM and 8 MB of storage. This microcontroller is fully compatible with the Arduino IDE, offering extensive capabilities such as 8 UART ports, 3 I2C interfaces, 55 digital pins, and 18 analog pins. Operating at 3.3V, it aligns with standard voltage requirements in electronics, ensuring broad compatibility with various components and systems.

Despite its robust specifications and Arduino support, the IMXRT1062DVJ6 presents challenges due to its limited familiarity among team members, necessitating potential training efforts. Moreover, its closed-source bootloader could pose procurement issues, restricting flexibility in component sourcing and complicating maintenance and upgrades. Careful consideration of these factors is essential when leveraging the IMXRT1062DVJ6 for projects requiring high performance and extensive connectivity options.

### 3.4.10 RP2040

The RP2040, known for powering the Raspberry Pi Pico microcontroller board, features a capable 32-bit ARM Cortex-M0+ processor clocked at 133 MHz. It offers essential connectivity options with 2 UARTs and 2 I2C interfaces for versatile communication between devices. The board includes 25 digital pins and 4 analog pins, enabling flexible interfacing with various sensors and peripherals. Operating at 3.3V, it adheres to standard voltage requirements, ensuring compatibility across different electronic components and systems.

Despite its technical capabilities, the RP2040 faces a challenge due to its lower familiarity among team members, which may require additional training and adaptation efforts to maximize its potential effectively. However, its integration into the Raspberry Pi ecosystem and support within the Arduino IDE make it a compelling choice for projects requiring a balance of performance, connectivity, and affordability.

### 3.4.11 Comparison

*Table 3.4: Secondary MCU comparison*

| Specification | ATMEGA328 | IMXRT1062DVJ6 | RP2040 |
|---|---|---|---|
| I/O | 1 UART, 1 I2C, 14 Digital Pins, 8 Analog Pins | 8 UART, 3 I2C, 55 Digital Pins, 18 Analog Pins | 2 UART, 2 I2C, 25 Digital Pins, 4 Analog Pins |
| Clock Speed (Mhz) | 20 | 600 | 133 |
| Cost | $2.63 | $16 | $1 |
| Dimensions (mm) | 7.1x7.1 | 10x10 | 7x7 |
| Manufacturer | Atmel | NXP USA Inc | Raspberry PI Foundation |

### 3.4.12 Secondary MCU Selection

Selecting our secondary microcontroller (MCU) became crucial after choosing the ESP32-S as our main MCU. Our final choice was the ATMEGA328, despite its fewer UARTs and I2C ports compared to other options, it aligns perfectly with our project's specific requirements. However, a significant drawback is its reliance on 5V for all digital pins, necessitating level shifting for communication between the ESP32-S and ATMEGA328, which adds complexity to our circuit design.

An advantage of the ATMEGA328 over alternatives like the IMXRT1062DVJ6 is its straightforward integration into a PCB layout. It requires only an external 16 MHz clock and a few capacitors, simplifying the hardware design process compared to chips that demand proprietary external boot loader chips. Moreover, our team's extensive familiarity with the ATMEGA328 significantly reduces implementation risks associated with managing two MCUs in our system. This familiarity not only accelerates development and debugging but also enhances our confidence in achieving reliable performance.

In summary, while the ATMEGA328 may have limitations in terms of I/O capabilities and voltage requirements, its ease of implementation, compatibility with Arduino IDE, and our team's proficiency make it a practical choice for ensuring the success and robustness of our project. These factors collectively contribute to a streamlined development process and effective management of potential complexities associated with dual MCU integration.

## 3.5 Communication Protocols

To integrate our plant watering system with a app, wireless connectivity is essential to avoid the complexity and barriers of physical data wires like Ethernet. Wireless communication ensures easy deployment and user interaction flexibility. A communication protocol defines rules for data exchange between devices or systems, encompassing message formatting, timing synchronization, error detection, flow control, and addressing. Examples include Wi-Fi, Bluetooth, and LoRaWAN, each suited to specific needs like data rate, range, power efficiency, and network structure, crucial for selecting the optimal protocol based on application requirements and environmental constraints.

### 3.5.1 Wifi (802.11n)

Wi-Fi is a ubiquitous method for connecting devices to the internet wirelessly through radio waves, typically operating on the 2.4GHz and 5GHz frequency bands. It facilitates high-speed data transmission, making it ideal for applications requiring substantial bandwidth, such as streaming images and videos. Wi-Fi's widespread adoption in households worldwide ensures compatibility and accessibility, enabling seamless integration of devices into existing networks without the need for additional infrastructure.

We chose to look at version 802.11n was picked over the other versions because it was the highest performing variant supported by most wifi enabled microcontrollers. It also has a legacy mode that can support the Previous 802.11 B/G versions.

### 3.5.2 Bluetooth

Bluetooth is a widely adopted wireless technology operating in the 2.4 GHz band. While it can reach 100m, it is typically used in short-range communication up to approximately 10 meters. Its advantages include widespread compatibility across various devices, low power consumption ideal for battery-operated gadgets, straightforward pairing processes, and built-in security features like encryption. However, Bluetooth's limitations include its short operational range, potential interference in crowded wireless environments, moderate data transfer speeds compared to Wi-Fi, occasional compatibility issues between different versions, and audio quality that may not match wired connections for high-fidelity audio and stable performance. Overall, Bluetooth remains popular for its convenience in connecting devices wirelessly, particularly for tasks like audio streaming, file sharing, and IoT connectivity.

### 3.5.3 BLE

Bluetooth Low Energy (BLE) is a variant of Bluetooth optimized for low-power applications, operating in the 2.4 GHz band. It is used today in IoT devices, hand helds, and wearables by minimizing energy consumption for extended battery life. BLE facilitates quick connection establishment, making it ideal for random data transfers like sensor readings and notifications. While it maintains robust security features such as encryption, BLE trades data transfer speed compared to classic Bluetooth (by a factor of 3), suitable primarily for small data packets. Its range is limited to 10-100 meters, and development complexity arises from managing power efficiently across its operational modes. Overall, BLE is widely adopted for its energy efficiency and compatibility across devices, serving as a foundational technology in the realm of connected devices and smart environments.

### 3.5.4 Lora

LoRaWAN (Long Range Wide Area Network) is a low-power, long-range wireless communication protocol designed for connecting low-power IoT devices to the internet. Operating the 915 Mhz unlicensed radio frequency band, LoRaWAN supports communication over distances of 15 kilometers in rural settings and several hundred meters in urban environments. It emphasizes low power consumption, enabling devices to operate on battery power for extended periods, and offers scalable network architecture with robust security features like encryption. LoRaWAN is interesting for our project due to its current applications in agriculture, asset tracking, smart metering, and smart city infrastructure due to its ability to provide reliable connectivity over large areas without requiring extensive infrastructure investments. Its combination of long-range capability, energy efficiency, and affordability makes it a good choice for an IoT device like our plant watering system.

Lora comes in 3 different modes. There is Class A, B, and C. Class A is useful for sensors that only need to transmit data. It waits until it has to transmit and only after transmitting does it listen for a receiving message. This is the most power efficient Class. Class B has a timer and at that time listens for receiving messages and sends its own. It requires being in sync with its target so they both check at the same time. Finally Class C is the least efficient and is always listening for incoming messages.

### 3.5.5 Comparison

*Table 3.5: Communication Protocols comparison*

| Specification | Wifi (802.11n) | Bluetooth | BLE | LoRaWAN |
|---|---|---|---|---|
| Max Distance | 50 meters | 100m | 100m | 15km |
| Data Rate | 150 mbps | 3mbps | 1mbps | 22kps |
| RF band | 2.4ghz or 5ghz | 2.4ghz | 2.4ghz | 915Mhz |

### 3.5.6 Wireless Communication Standard Selection

From the list of wireless technologies we researched, we found that LoraWAN is the most power efficient and has the furthest distance but severely lacks in data transfer speeds. It also isn’t embedded in any of the microcontrollers we looked at meaning we would have to add an independent chip which adds complexity for the PCB.

For Bluetooth BLE, it has better data transfer than lora and is also low power, but has a significantly shorter range. It benefits over lora by being widely adopted in various microcontrollers. Similar to BLE, regular Bluetooth shares the same range and microcontroller ubiquity as BLE. Where it varies is in its data transfer which is 3 times larger as well as pulls more power.

Finally, looking at Wifi (802.11n) it has the highest data rate making it ideal for streaming images and sensor data. It has the smallest maximum distance, however in terms of practical distance it is slightly better than bluetooth making it suitable for indoor usage. Just like Bluetooth it is widely adopted among wireless microcontrollers meaning easier PCB development. The last major benefit to wifi is that the infrastructure to connect to the internet allows for flexibility in how the message gets received to the target device. Bluetooth and Lora require being in range of the transmitter. Where as the internet facilitates communication between a phone and the wifi device anywhere within an internet connection.

## 3.6 Cameras

In our project, the essential task is streaming images to a application, necessitating the selection of an appropriate camera. Key specifications critical for evaluating each camera include resolution, which determines image clarity and detail, wavelength sensitivity for capturing specific light spectra tailored to the application's needs, and frame per second (FPS) for ensuring smooth and responsive image streaming. These specifications guide our decision-making process to ensure the chosen camera aligns perfectly with the technical requirements essential for seamless integration and optimal performance within our application.

### 3.6.1 OV2640

The Ov2640 sensor is a high-resolution camera with a 1600 x 1200 resolution. Designed to capture the visible light spectrum ranging from 500 to 950nm, it stands out with its ability to deliver detailed images and videos across various lighting conditions. Its maximum frame rate is 60 frames per second (FPS), ensuring smooth and fluid motion capture.

This camera sensor holds significant appeal due to its widespread adoption and frequent inclusion as a default option on many development boards. Its popularity stems from its reliability and compatibility, making it a preferred choice among developers and engineers alike. The Ov2640's ubiquity simplifies integration into projects, offering a straightforward solution for applications requiring high-quality image and video capture.

### 3.6.2 OV5640

The OV5640 camera has a high resolution of 2592 x 1944 pixels, ensuring exceptional image clarity and detail. One of its standout advantages lies in its versatile frame rate options, catering to a wide range of applications and developmental needs. Operating at its maximum resolution, the OV5640 achieves 15 frames per second (fps), while switching to 1080p mode increases the frame rate to 30 fps, and it can handle even higher speeds—up to 120 fps—in 320p mode. This flexibility enables developers to tailor performance to specific project requirements, accommodating various scenarios with ease.

### 3.6.3 OV7725

A 640 x 480 resolution camera. Despite being discontinued It was selected to be researched due to still being accessible and its low bandwidth. The downsides to this are that that low bandwidth is due to a low resolution and frame rate. Also due to being discontinued could have poor support from the manufacturer if their help is required.

The 640 x 480 resolution camera, although discontinued, was chosen for research primarily because it remains accessible and offers low bandwidth consumption. This particular model was selected due to its availability and the practicality of integrating it into projects with constrained bandwidth requirements. However, its advantages come with notable trade-offs. The low resolution and frame rate limit its capabilities compared to more modern counterparts, which may affect the quality and smoothness of image and video capture.

Moreover, one significant concern with using a discontinued camera model is the potential lack of manufacturer support. Issues such as troubleshooting and updates could pose challenges if technical assistance is needed. This lack of ongoing support may impact the long-term reliability and maintenance of our project, necessitating careful consideration and contingency planning in development cycles.

Despite these considerations, the 640 x 480 resolution camera remains a viable option for specific applications where its low bandwidth and accessibility outweigh the drawbacks associated with its discontinued status. Understanding its limitations and planning accordingly can help mitigate potential issues, ensuring effective utilization within the constraints of the project's requirements.

*Table 3.6: Camera comparison*

| **Specification** | ***OV2640*** | ***OV5640*** | ***OV7725*** |
|---|---|---|---|
| **Pixels (px x px)** | 1600 x 1200 | 2592 x 1944 | 640 x 480 |
| **Cost** | $3 | $4 | $5 |
| **Max supported FPS** | 60 | 120 | 60 |
| **Max power (mW)** | 140 | 420 | 120 |

### 3.6.4 Camera Selection

Right away the OV7725 is a weak candidate. Despite its low power and bandwidth usage, its low resolution and small light spectrum renders it essentially useless for monitoring a plant. The OV2640 is a strong choice with a decent 1600p resolution and wavelength range. Its inclusion in most development board kits also helps reduce the time and cost of the prototype. It also means it is well documented. Also due to its resolution and FPS, its bandwidth requirements aren’t unreasonably high. However, the OV5640 is easily the most versatile camera. It has a wide range of resolutions and framerates making it suitable for prototyping and tolerant to power and bandwidth budgets.

## 3.7 Sensor

One of the most fundamental components of our project was the set of sensors placed near the plants. These sensors were vital for monitoring the plants' needs and assessing their health, providing us with real-time data to make informed decisions. Specifically, we used a pH sensor to measure the acidity or alkalinity of the soil, a temperature sensor to monitor environmental conditions, and a water level sensor to ensure the plants received the right amount of water. Each of these sensors provided essential data to ensure optimal growing conditions, allowing us to implement timely interventions and maintain the plants' health and growth.

**Calibration Importance–** Calibration of sensors played an important role in ensuring the accuracy, reliability, and functionality of measurement systems across various applications. By calibrating sensors, we established a known relationship between the sensor's output and the quantity being measured, such as pH levels, temperature, water levels, moisture content, and electrical signals. This process not only verified the sensor's accuracy against known standards or reference values but also corrected any inherent errors or drift that occurred over time or due to environmental factors. In our project, precise calibration of sensors such as pH, temperature, water level, moisture, and relay sensors were important for making informed decisions and taking precise actions, whether it was adjusting nutrient solutions, regulating environmental conditions, optimizing water usage, or automating control systems. Calibration ensured that our sensor data was trustworthy and consistent, enabling us to effectively monitor and manage plant growth conditions, enhance operational efficiency, and ultimately achieve reliable and sustainable outcomes in both small-scale indoor gardening and larger agricultural settings.

## 3.8 pH sensor

The pH sensor was a critical component of our plant watering system project. It helped the user measure the alkalinity and acidity levels in the soil and water, which were essential factors for plant health. Maintaining the correct pH level was vital for nutrient availability; most plants thrived in slightly acidic to neutral pH levels (around 6.0 to 7.0). By continuously monitoring the pH, the system alerted the user when the soil conditions were outside the optimal range, allowing for timely adjustments. This ensured that nutrients were effectively absorbed by the plants, promoting healthy growth and preventing nutrient deficiencies. Research showed that soil pH significantly impacted plant metabolism and growth, making it a key factor in sustainable agriculture practices (Smith, 2020).

To **calibrate** the GAOHOU pH0-14 and S-pH-01-pH sensor, we first prepared the pH calibrating buffer solutions at pH 4.0, pH 7.0, and pH 10.0 (Faranux, 2020). We ensured that the sensor was connected to the microcontroller according to its specifications. Then, we submerged the sensor probe into the pH 7.0 buffer solution and allowed it to stabilize for several minutes, verifying the sensor's readings against established pH values to ensure accuracy across its entire measurement range.

The **GAOHOU pH0-14** sensor module was tailored for Arduino controllers, offering a user-friendly integration process with straightforward and practical connections. It included an LED power indicator for real-time status monitoring, a BNC connector designed for seamless interfacing with pH sensors, and a PH2.0 interface that directly linked to the analog input port of a controller, facilitating rapid deployment and reliable operation in pH sensing applications. Calibration of the pH probe was crucial to ensure accurate measurements. According to the manufacturer's guidelines, regular calibration using standard pH solutions was necessary to maintain precision and reliability in pH readings (Researcher, 2020). This process involved adjusting the sensor's readings to correspond accurately with known pH values, thus enhancing the module's effectiveness in various environmental and agricultural contexts.

The **S-pH-01 pH** The sensor was an advanced device used to measure pH levels in industrial settings. It met international standards and featured cutting-edge technology like a solid dielectric and a large-area polytetrafluoroethylene liquid junction, which reduced clogging and simplified maintenance. Automatic Temperature Compensation (ATC) ensured accurate pH readings even when temperatures fluctuated. The sensor was rated IP65, meaning it was waterproof and could withstand tough professional environments, ensuring long-lasting durability. It outputted data via MODBUS-RTU RS485 or a 4~20mA current, making it easy to integrate with various devices such as data loggers and controllers. With its low power consumption, compact size, and versatile design, the S-pH-01 sensor was suitable for both indoor and outdoor applications, providing reliable pH measurements wherever needed.

***Final Selection***

***The pH sensor we used was the GAOHOU pH0-14.***

*Table 3.7: pH sensor comparison*

| | **GAOHOU pH0-14** | **S-pH-01** | **Atlas Scientific EZO – pH** |
|---|---|---|---|
| **Power Supply** | 5V | 3.6 - 30V DC | 3.3V - 5V |
| **Dimensions** | 20mm x 10mm | 160mm x 30mm | 13mm x 20mm |

| | | | |
|---|---|---|---|
| **Measuring Range** | 0 – 14pH | 0 – 14pH | 0 - 14pH |
| **Measuring Temperature** | 0 – 60 C | -40 – 85 C | 0 – 85 C |
| **Accuracy** | +- 0.1pH (25 degree C) | +- 0.1pH | +- 0.002 pH |
| **Price** | $30.99 | $139 | $45.99 |

We selected the **GAOHOU PH0-14** sensor for our project due to its precision in measuring pH levels. Although this sensor was designed for use with liquids, we planned to adapt it for soil pH measurements by creating a soil-water mixture. The process involved mixing a soil sample with distilled water to create a solution that could be tested. This method allowed us to indirectly measure the soil's pH by analyzing the pH of the resulting liquid. To implement this, we designed a system to extract a sample of soil and mix it with water automatically. The solution was then filtered to remove any particulates before being measured with the GAOHOU PH0-14 sensor. Additionally, we developed a calibration process to ensure accurate readings, accounting for potential variations in soil composition and water quality. This approach enabled us to continuously monitor soil pH and make necessary adjustments to the watering system to maintain optimal plant health.

## 3.9 Temperature sensor

This sensor was a very important component of our project, monitoring and managing environmental conditions vital for plant growth. Ensuring optimal growth conditions was essential, as the sensor helped maintain the ideal temperature range for indoor plants, which, according to our research, was 70°F to 80°F during the day and 65°F to 70°F at night (Johnson, 2024). The sensor detected ambient temperature and converted it into an electrical signal, which the system's control unit processed to make real-time adjustments. Additionally, the sensor adjusted the watering needs based on temperature fluctuations, providing more water during higher temperatures to compensate for increased evaporation and transpiration, and reducing water during cooler periods to prevent overwatering and root rot. By continuously monitoring temperature, the sensor helped prevent heat stress and frost damage, ensuring plants remained within their optimal growth range. This automated approach not only optimized water usage but also contributed to the overall health and productivity of the plants, making the system efficient and reliable for both small-scale indoor gardening and larger agricultural operations.

**Calibrating** a temperature sensor involved ensuring its readings accurately corresponded to known temperatures, which was critical for maintaining precision in environmental monitoring and control systems. We started by connecting the temperature sensor to the microcontroller and ensuring it was properly powered and grounded. Next, we placed the sensor in a stable temperature environment, such as a water bath with a known temperature. We recorded the sensor's output and compared it to the actual temperature using a calibrated thermometer or reference temperature source. We adjusted the sensor's calibration parameters, typically done through software adjustments, until the sensor's readings aligned closely with the reference temperature readings. We repeated this process across a range of temperatures if necessary, ensuring accuracy throughout the sensor's operational range. Regular recalibration was recommended to account for any drift or environmental changes, maintaining the sensor's accuracy over time in applications such as climate control, industrial processes, or scientific research.

The **TMP117AIDRVR**, a high-precision digital temperature sensor produced by Texas Instruments, was renowned for its accuracy and low power consumption. Its integration into our project offered significant advantages. The sensor's high accuracy, with a typical precision of ±0.1°C, ensured precise environmental monitoring, creating optimal conditions for plant growth (Texas Instruments, 2023). Additionally, its low power consumption of just 3.5 µA in active mode and 150 nA in shutdown mode enhanced the energy efficiency of our system, making it ideal for battery-operated or energy-conscious applications. The sensor's reliability was further ensured by its wide operating range, from -55°C to 150°C, making it robust and suitable for various environmental conditions, thus contributing to the overall efficiency and effectiveness of our plant watering system.

The **DS18B20** was a digital temperature sensor widely used in various applications due to its simplicity and accuracy. It used the 1-Wire protocol for communication, which meant it only required one data line (and ground) to communicate with a microcontroller. The sensor provided temperature readings with a precision of up to 12 bits, resulting in an accuracy of ±0.5°C over much of its range (-55°C to +125°C). It could also operate in parasite power mode, eliminating the need for an external power supply. One of the key features of the DS18B20 was its unique 64-bit serial code, which allowed multiple DS18B20 sensors to be connected to the same 1-Wire bus, enabling temperature monitoring at different points with a single microcontroller pin.

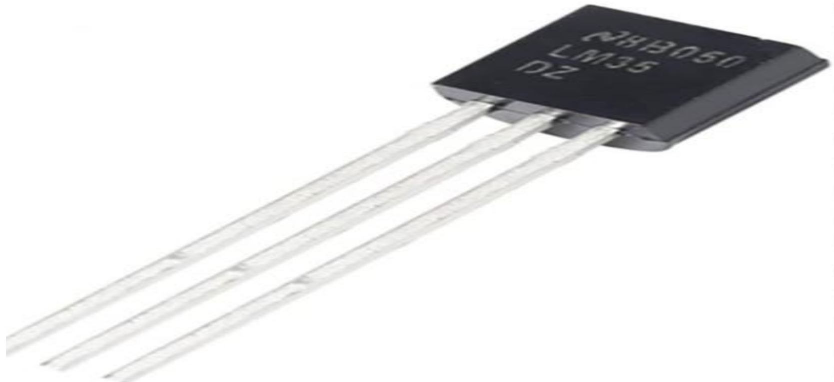
***Final Selection***

*Table 3.8: Temperature sensor comparison*

| | TMP117AID | DS18B20 | LM35DZ |
|---|---|---|---|
| **Sensor Accuracy** | 0.1, 0.2 degrees C | 0.5 degrees C | 0.5 C |
| **Temperature Range** | -55 to 150 C | -55 to 125 C | -55 to 150 C |
| **Supply Voltage** | 1.8V – 5.5V | -0.5V – 6.0V | 4V - 30V |
| **Supply Current** | 3.5u A – 150nA | 750nA – 1mA | 60uA |

| | | | |
|---|---|---|---|
| **Price** | $2.9 | $7.42 | $2.03 |

***The temperature sensor we used was the LM35DZ.***




The **LM35DZ** was an excellent choice for temperature measurement due to its high accuracy of ±0.5°C at room temperature, making it suitable for precise monitoring applications like plant watering systems. Its linear output of 10 mV per degree Celsius simplified integration with microcontrollers, as no complex conversion was required. Additionally, the LM35DZ did not need external calibration, saving time during setup. It also operated over a wide temperature range from -55°C to 150°C, making it versatile for various environments. With low power consumption (60 µA), it was ideal for energy-efficient or battery-operated devices, providing both accuracy and reliability for long-term use.

## 3.10 Water level sensor

Having a water level sensor was very important for several reasons. Firstly, it helped conserve water by ensuring that plants received only the amount of water they needed, thereby reducing waste. Overwatering could be as harmful as underwatering, leading to root rot and other plant diseases. The sensor detected the soil's moisture level, and if the soil was dry, it triggered the system to add water to the pot. Additionally, a water level sensor provided a more automated and efficient approach to plant care. It removed the guesswork involved in traditional watering methods, ensuring that plants were consistently hydrated according to their needs. This was especially beneficial for individuals who may not have had the time to regularly check their plants or for large-scale gardening operations. Furthermore, maintaining the correct moisture level improved plant health and growth. Proper hydration ensured that nutrients in the soil were effectively absorbed by the plant roots. By preventing both underwatering and overwatering, the sensor helped create an optimal growing environment, leading to healthier, more robust plants.

**Calibrating** a water level sensor involved several steps to ensure accurate and reliable measurement of water levels in tanks or reservoirs. We began by installing the water level sensor at the desired location within the tank, ensuring it was securely mounted and properly connected to the microcontroller. We started with an empty tank and recorded the sensor's output reading, which typically corresponded to the lowest water level at 0% capacity. Then, we gradually filled the tank with water in increments, stopping at known intervals of 25%, 50%, 75%, and 100% capacity. At each interval, we recorded the sensor's output reading. We verified the accuracy of the calibration by emptying and refilling the tank, comparing the sensor's output readings against the known water levels.

The **5" eTape Liquid Level** Sensor with Plastic Casing was an innovative and reliable solution for accurately measuring liquid levels in various applications. This sensor utilized a resistive output that varied with the height of the liquid, providing precise measurements essential for effective liquid level management. Encased in durable plastic, the sensor was protected against potential damage from the liquid environment, ensuring longevity and consistent performance. Its compact size and robust design made it ideal for integration into systems where space was limited, and reliable liquid level monitoring was crucial. Whether used in industrial processes, automated plant watering systems, or other applications, the 5" eTape Liquid Level Sensor offered a dependable, high-quality solution for accurate and efficient liquid level measurement.

The **HC-SR04** Ultrasonic Sensor Module was a popular and affordable device widely used in personal projects for distance measurement. This sensor operated by emitting an ultrasonic sound wave at a frequency of 40 kHz through its trigger pin and then listening for the echo via its echo pin. The time taken for the sound wave to return was measured and used to calculate the distance between the sensor and an object. With a range of 2 cm to 400 cm and an accuracy of about 3 mm, the HC-SR04 was well-suited for applications such as obstacle avoidance, liquid level measurement, and proximity sensing.

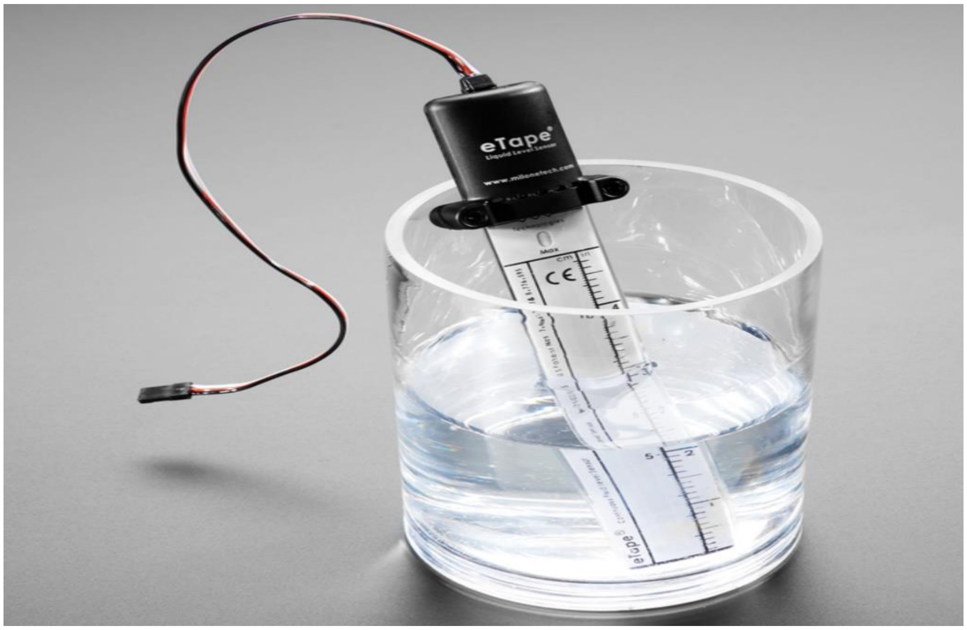
### *Final Selection*

*Table 3.9: Water level sensor comparison*

| | **eTape Liquid Level** | **HC-SR04** | **A02YYUW** |
|---|---|---|---|
| **Sensor Accuracy** | +- 0.5% | +-0.3cm | +-1 cm |
| **Sensor Current** | 500uA | 15mA | 15mA |
| **Dimensions** | 127mm x 9.5mm x 0.38mm | 20mm x 18mm x 4mm | 41mm x 72mm |
| **Sensor Voltage** | 5V | 5V DC | 3V – 5.5V |
| **Price** | $59.95 | $4.50 | $7.90 |

***The water level sensor that we used was the eTape Liquid Level.***

We utilized the **A02YYUW** waterproof ultrasonic sensor to monitor the water level in a gallon of water connected to the plants. This sensor was placed inside the gallon container to continuously measure the water level, ensuring that the system knew when the water was running low. When the water level dropped below a certain threshold, the sensor triggered a notification or an automated refill mechanism, prompting the user to refill the gallon as needed. The waterproof design of the A02YYUW ensured reliable performance in the wet environment, making it an ideal choice for maintaining a consistent water supply for house plants.




## 3.11 Moisture Sensor

The moisture sensor also played an important role in optimizing plant watering systems by providing real-time feedback on soil moisture levels. This data was essential for efficient water management, ensuring plants received adequate hydration without overwatering, which could lead to root rot and other issues. By accurately measuring soil moisture, the sensor enabled automated watering schedules that aligned with the specific moisture needs of different plants or soil types. This capability not only conserved water but also promoted healthier plant growth by preventing drought stress or waterlogging. Additionally, the moisture sensor helped monitor soil conditions over time, allowing adjustments to watering regimes based on seasonal changes or environmental factors. Overall, integrating a moisture sensor enhanced the effectiveness of plant care strategies, supported sustainable water usage practices, and contributed to the overall vitality and productivity of plants in both indoor and outdoor settings.

**Calibrating** a moisture sensor involved establishing a relationship between the sensor's output and the actual moisture content in the soil, ensuring accurate readings for effective plant care. We began by selecting representative soil samples across a range of moisture levels, from dry to saturated. We inserted the moisture sensor probes into each sample and recorded the corresponding sensor readings. We used these readings to create a calibration curve or table that correlated sensor output (analog or digital) with actual soil moisture content. We adjusted the sensor's calibration settings or programming to align its readings with the recorded moisture levels. We calibrated by testing the sensor in various soil conditions and comparing its output against known moisture values. We regularly verified and adjusted the calibration as needed, especially if environmental conditions changed or different soil types were encountered. This calibration process ensured that the moisture sensor provided reliable data for optimizing irrigation schedules and promoting healthy plant growth.

The **Grove-Moisture** Sensor was a versatile and easy-to-use sensor designed for measuring soil moisture levels in agricultural and gardening applications. It featured a simple yet effective design, comprising two probes that were inserted into the soil. The sensor operated on a resistive principle, where the moisture level in the soil affected the conductivity between the probes. As the soil moisture increased, the conductivity between the probes also increased, resulting in a higher analog voltage output. This output could be read by an analog-to-digital converter (ADC) on microcontrollers like Arduino, making it straightforward to integrate into various projects. The Grove connector interface ensured seamless connectivity with Grove-compatible development boards and sensors, eliminating the need for soldering or complex wiring. Its compact size and durable construction made it suitable for both indoor and outdoor environments, facilitating precise monitoring of soil moisture to optimize watering schedules and promote healthy plant growth. Overall, the Grove Moisture Sensor was an essential tool for anyone looking to automate and improve the efficiency of their plant watering systems.

The **Adafruit STEMMA** soil sensor was an I2C capacitive moisture sensor designed to measure soil moisture levels. The sensor featured a built-in temperature sensor for added functionality, making it ideal for plant watering systems. The sensor was corrosion-resistant, ensuring durability when exposed to soil and moisture over extended periods. With its I2C interface, it was easy to integrate into microcontroller projects, providing reliable and precise moisture readings.

### Final Selection

We utilized the **Adafruit STEMMA** soil moisture sensor to ensure precise monitoring of soil moisture levels. This highly reliable sensor was strategically placed directly in the soil, where it continuously measured moisture content. By embedding the sensor in the soil, we obtained real-time data on moisture levels, allowing the system to automatically adjust watering schedules and quantities. The integration of the Adafruit STEMMA sensor into our system exemplified our commitment to leveraging advanced technology for smarter, more efficient plant care.

***The moisture sensor that we used was that Adafruit STEMMA.***

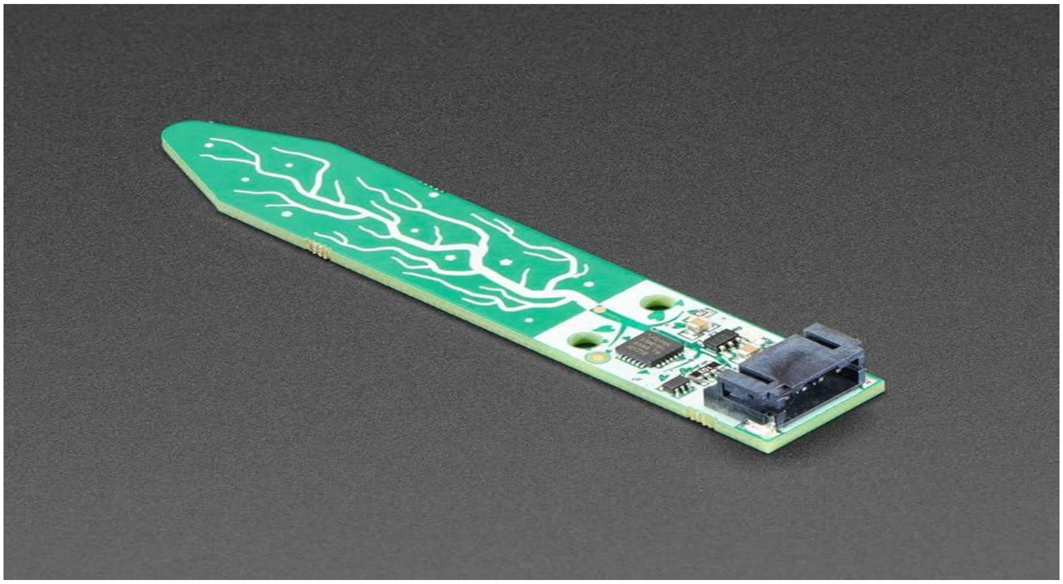


*Table 3.10: Moisture sensor comparison*

| | Grove-Moisture Sensor | Adafruit STEMMA | DFRobot Capacitive Soil |
|---|---|---|---|
| **Operating Voltage** | 3.3 – 5.5 VDC | 3V – 5V | 3.3V - 5.5V DC |
| **Output Voltage** | 0 – 3 VDC | 3.3V – 5V | 1.2V - 2.5V |
| **Operating Current** | 35mA | 20mA | 30mA |
| **Dimensions** | 60mm x 20mm x 8mm | 20mm x 18mm x 4mm | 120mm x 28mm |
| **Price** | $3.30 | $7.50 | $8.74 |

## 3.12 Relay Sensor

The relay sensor played a fundamental role in our plant watering system by enabling automated control over water pumps, valves, or other electrical devices based on sensor inputs. This capability allowed us to implement precise and responsive watering schedules tailored to the specific moisture needs of different plants or soil conditions. By interfacing with sensors such as moisture or water level sensors, the relay sensor triggered actions like turning on or off water flow, adjusting irrigation cycles, or activating pumps only when necessary. This automation optimized water usage by delivering water directly to plants when and where it was needed, while also reducing the manual effort required for monitoring and maintenance. Moreover, the relay sensor enhanced system reliability by protecting against overwatering or underwatering, thereby promoting healthier plant growth and conserving water resources in both small-scale indoor gardens and larger agricultural operations. Its role in controlling electrical outputs based on sensor data ensured efficient and sustainable management of plant watering, contributing to improved plant health and overall system efficiency.

**Calibrating** a relay sensor involved setting up and adjusting its control parameters to ensure accurate and reliable operation in response to specific sensor inputs. We started by connecting the relay sensor to our microcontroller, ensuring all wiring was secure and properly configured. Then, we defined the desired thresholds or conditions under which the relay would activate or deactivate (turn on or off). This might have involved programming logic in the microcontroller that interpreted sensor readings (such as from moisture or temperature sensors) and triggered the relay accordingly. Once configured, we tested the relay sensor in real-world conditions to verify its responsiveness and accuracy. We modified the calibration by adjusting parameters such as activation thresholds, delay times, or hysteresis to optimize performance and ensure that the relay activated and deactivated precisely as required. We regularly monitored and re-calibrated the relay sensor as needed, especially if environmental conditions changed or to maintain consistent operation over time.

The **Keyes Relay Module** was a useful and commonly used component in electronics and automation projects, particularly with microcontroller platforms like Arduino or Raspberry Pi. It featured multiple channels, allowing it to control several high-power devices simultaneously. The module was designed with optocouplers for isolation, enhancing its reliability and protecting sensitive microcontroller circuitry from voltage spikes or feedback. Each channel on the Keyes Relay Module was capable of switching high currents, usually up to 10A at voltages suitable for a range of applications, from controlling motors and pumps to managing lighting systems or heaters. Its compact size and easy interface via digital pins made it popular among hobbyists and professionals alike for creating automated systems that required robust and reliable switching capabilities.

The **DF Robot Relay Shield** for Arduino was a useful module designed to simplify the integration of relay switching capabilities into Arduino projects. It featured multiple channels, typically offering four relay channels that could independently control high-current devices such as motors, lights, or pumps. Each relay channel was equipped with a dedicated LED indicator for visual feedback of its operational status, aiding in troubleshooting and monitoring. The shield was compatible with most Arduino boards, including the popular UNO and Mega models, and was easily stackable with other shields for expanded functionality. The DFRobot Relay Shield simplified the process of controlling electrical devices through Arduino, making it ideal for applications in home automation, robotics, and industrial automation where reliable switching and control of high-power devices were required. Its compact design and user-friendly interface made it accessible for hobbyists and professionals alike seeking robust relay capabilities integrated with Arduino microcontrollers.

**Final Selection**

The **SRD-05VDC-SL-C** relay is a reliable, compact, and cost-effective solution for controlling high-power devices. Operating at a low 5V DC input, it can switch up to 250V AC and handle 10A of current. Its small size makes it ideal for integration into tight spaces, and it is commonly used in microcontroller-based projects for automation tasks. These features make it a versatile choice for both hobbyists and professionals.

*Table 3.11: Relay sensor comparison*

| | Keyes Relay Modules | SRD-05VDC-SL-C | SainSmart 2-Channel |
|---|---|---|---|
| **Operating Voltage** | 220V | 5V | 5V |
| **Max AC Current** | 10A | 250A | 10A at 250V |
| **MAX AC Voltage** | 250V | 10V | 30V |
| **Dimensions** | 50mm x 40mm x 20mm | 19mm x 15.5mm | 80mm x 55mm x 25mm |
| **Price** | $6.12 | $4.59 | $7.99 |

***The relay sensor that we used was the* SRD-05VDC-SL-C.**


## 3.13 Voltage regulator

The voltage regulator held significant importance in our project, the "Plant Watering System," by ensuring stable and regulated voltage levels necessary for the reliable operation of electronic components and sensors. In this system, where various sensors (like moisture, temperature, and water level sensors) and actuators (such as pumps or valves) were controlled by microcontrollers or electronic circuits, consistent voltage was crucial to prevent damage to sensitive components and to ensure accurate data acquisition and control. The voltage regulator maintained a steady output voltage despite fluctuations in input voltage or load variations, thereby providing a reliable power supply to all components throughout the system's operation. This stability was essential for maintaining precise control over watering schedules, sensor readings, and actuator operations, ultimately contributing to efficient water management, enhanced plant health, and the overall reliability of the plant watering system.

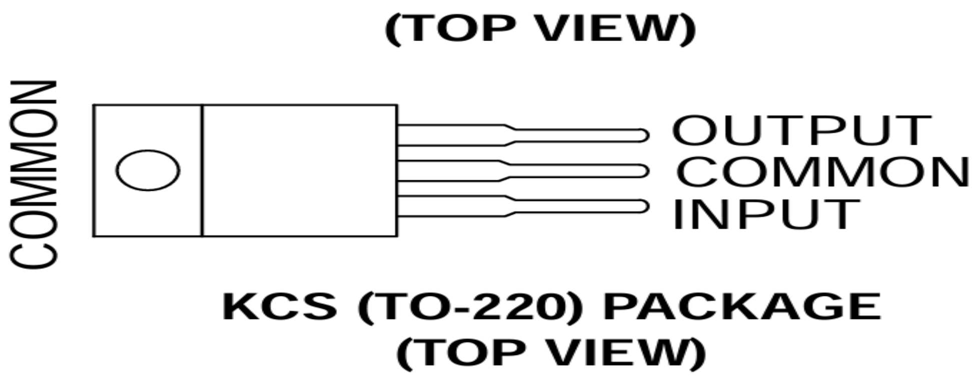
The **LM7805** was a widely used linear voltage regulator that provided a stable output voltage of 5 volts from a higher input voltage source. It was a member of the LM78xx series of voltage regulators manufactured by various semiconductor companies, known for their reliability and simplicity in design. The LM7805 handled input voltages typically ranging from 7V to 35V, making it versatile for a wide range of applications requiring a regulated 5V supply. Physically, the LM7805 came in a standard TO-220 package, which included a metal tab for heat dissipation when mounted on a heat sink. It consisted of internal circuitry that included a voltage reference, error amplifier, pass transistor, and current-limiting and thermal protection features. These features ensured stable operation under varying load conditions and protected the regulator from overheating or excessive current draw. In operation, the LM7805 required input and output capacitors to stabilize its operation and filter out noise. It provided a fixed 5V output voltage, which was suitable for powering microcontrollers like Arduino, sensors, LEDs, and other digital and analog circuits requiring a consistent power supply. Due to its straightforward operation and availability, the LM7805 was commonly used in hobbyist electronics projects, prototyping, and as a basic power supply component in various industrial and consumer electronics applications.

The **LM317** was a widely used adjustable voltage regulator integrated circuit (IC) capable of providing a stable output voltage that could be adjusted over a wide range. It operated as a linear regulator, meaning it regulated the output voltage by dissipating excess power as heat. The LM317 could accept an input voltage ranging from 3V to 40V, making it suitable for a variety of applications requiring a stable and adjustable output voltage. One of the key features of the LM317 was its ability to provide an adjustable output voltage that could be set using external resistors. By connecting a resistor network to its adjustment pin (ADJ), the output voltage could be precisely set to any desired value between 1.25V and 37V. This adjustability made the LM317 highly versatile, allowing it to meet the specific voltage requirements of different electronic circuits and components. The LM317 also included internal current limiting and thermal shutdown features to protect itself from overheating or current overload conditions, enhancing its reliability and longevity in operation. It was commonly used in various applications such as power supply circuits, battery chargers, voltage regulators for microcontroller projects, and as a general-purpose adjustable voltage regulator in both hobbyist and industrial electronics.

### Final Selection

*Table 3.12: Voltage Regulator comparison*

| | LM7805 | LM317 | AMS 117-05 |
|---|---|---|---|
| **Operating Voltage** | 35V | 37V | 5V |
| **Output Voltage** | 5V | 1.2V | 5V |
| **Operating Current** | 1.5A | 1.5A | 5mA |
| **Operating Temperature** | 0C – 125C | 0C – 125C | 0C - 125C |
| **Price** | $0.58 | $0.76 | $0.47 |




Choosing the **AMS117-05** as a voltage regulator provided several benefits, making it an ideal option for many applications. It featured a low dropout voltage of approximately 1.1V at full load, allowing it to maintain a stable 5V output even when the input voltage was close to the desired output, which was particularly useful in battery-powered devices. The regulator could supply up to 1A of output current, sufficient for low to moderate power applications, and came with integrated protection features such as thermal shutdown and current limiting, enhancing safety and reliability. Additionally, it operated across a wide temperature range (-40°C to +125°C) and had low noise operation, ensuring stable performance in sensitive applications like analog circuits.

# 4. Related Standards and Design Details

We delved into the standards and design details for our automatic plant watering system, ensuring that it effectively prioritized plant health, safety, and functionality. Design details included integrating various sensors, such as the Adafruit STEMMA for soil moisture and the GAOHOU PH0-14 for soil pH, with careful calibration to maintain accurate readings. The system also incorporated the A02YYUW ultrasonic sensor for water level monitoring, positioned within the gallon to accurately gauge refill needs. The DF Robot Relay Shield managed power distribution, supported by the LM7805 voltage regulator to provide stable voltage. Emphasis on robust enclosures protected electronic components from environmental damage, while thorough testing and regular maintenance ensured reliable operation. This comprehensive approach not only addressed technical requirements but also enhanced the overall functionality and safety of the system, making it a complete solution for maintaining plant health.

## 4.1 Standards

When we discussed standards for our automatic plant care system, we referred to the specific requirements and guidelines that ensured the system's reliability, safety, and effectiveness. Adhering to established standards was crucial not only for the system's performance but also for safeguarding the user and the environment. Safety was paramount, including compliance with various laws and regulations governing electrical and electronic devices. This included ensuring proper insulation, protection against electrical hazards, and adherence to electromagnetic compatibility requirements.

For our automatic plant care system, we aimed to simplify the user experience by developing a dedicated app. This app featured options that allowed users to monitor and manage the health of their plants easily. Users could track real-time data on soil moisture, pH levels, and water availability, and receive notifications for when to refill the water reservoir or adjust care routines. To ensure compliance with all applicable laws and regulations, we followed industry standards and best practices for electronic devices and environmental safety. This included adhering to RoHS for hazardous materials, ensuring electromagnetic compatibility (EMC) standards were met, and implementing data privacy measures for user information. By integrating these features and maintaining regulatory compliance, we aimed to create a user-friendly and effective system that promoted the long-term health and well-being of house plants.

## 4.2 Safety Standards

When discussing safety standards in the context of our automatic plant watering system, we addressed the sensors included in the design, such as the temperature and moisture sensors. Ensuring these sensors operated effectively and safely was essential for maintaining the system's overall integrity and reliability. Additionally, adherence to electromagnetic compatibility (EMC) standards prevented electromagnetic interference that could disrupt the functionality of the sensors or other nearby electronic devices.

In addition to ensuring sensor safety and functionality, we considered the environmental impact of our automatic plant watering system. By adhering to the Restriction of Hazardous Substances (RoHS) directive, we ensured our sensors and electronic components were free from harmful materials like lead, mercury, and cadmium, which could cause significant environmental damage if not properly managed. Furthermore, we used energy-efficient sensors and components to minimize the system's overall energy consumption, reducing its carbon footprint. We also prioritized using recyclable and biodegradable materials in the design and packaging of our system to lessen waste and support sustainable practices. By considering the environmental impact at every stage of development, we promoted the health of the plants and contributed to environmental conservation and sustainability.

After correctly connecting each sensor, we ensured that the entire system was well-protected through comprehensive circuit safety measures. This involved implementing robust insulation and shielding techniques to prevent accidental contact with live electrical components, reducing the risk of electric shock. Additionally, we used fuses and circuit breakers to protect the system from overcurrent conditions, which could damage the sensors or other components. Surge protectors safeguarded against voltage spikes resulting from external power disturbances, while proper grounding provided a controlled path for excess electricity to disperse safely. By integrating these circuit safety measures, we guaranteed not only the reliability and longevity of the sensors but also the safety and peace of mind for the user operating the automatic plant watering system.

## 4.3 Design Constraints

When designing an automatic plant watering system for house plants managed through an app, we encountered numerous design constraints that we had to navigate to meet the necessary standards. This process required balancing hardware and software integration and carefully considering various limitations. Economically, we ensured that the system was cost-effective, balancing quality and functionality with affordability to make it accessible to a wide range of users. This involved selecting sensors and components that offered the best value without compromising performance or safety.

The design of the system presented its own set of challenges. We created a compact and aesthetically pleasing device that seamlessly fit into a home environment, requiring meticulous planning to ensure that all components were housed efficiently and that the system remained user-friendly. Sensor selection and integration were critical aspects of this process, with each sensor chosen based on its accuracy, reliability, and compatibility with the overall system. We also ensured that the sensors were robust enough to operate in varying environmental conditions while maintaining their precision.

Throughout this process, we considered regulatory and safety standards to ensure the system complied with all relevant laws and guidelines, including standards for electronic safety, environmental impact, and data privacy. By remaining mindful of these constraints and rigorously testing our system, we aimed to deliver a reliable, efficient, and user-friendly automatic plant watering system that met the high standards required for modern smart home devices.

## 4.4 Economy

Creating an automatic plant watering system required balancing performance, accuracy, and cost-efficiency. As we developed this system, we recognized that the budget could quickly escalate, especially when aiming for high functionality and reliability. Our primary goal was to keep the system both efficient and affordable for users without compromising on quality. To achieve this, we set a budget of $400.

In our quest to achieve this, we meticulously selected the most accurate yet inexpensive sensors available. For example, when selecting temperature and moisture sensors, we ensured they offered precise readings at a reasonable cost, thereby enhancing the system’s overall value. However, finding an affordable pH sensor proved more challenging, as most pH sensors on the market were designed for liquid measurement and could be quite expensive, with prices often reaching up to $200.

To address this, we innovated by adapting a solution that allowed us to use a less expensive liquid pH sensor for soil measurement. By mixing soil samples with water, we effectively used the GAOHOU PH0-14 sensor, which cost around $14, to measure pH levels. This approach not only significantly reduced the cost but also maintained the accuracy and functionality required for effective plant care. By carefully managing our budget and making smart choices in component selection, we aimed to deliver a high-quality, cost-effective automatic plant watering system that met the needs of our users.

## 4.5 Design Constraints

When designing our automatic plant watering system, which incorporated both Arduino and the ESP-32-S microcontroller, we encountered a range of design constraints that required careful consideration and strategic management. Our design process involved navigating both hardware and software challenges to ensure a seamless and functional system.

A significant constraint was our decision to avoid using a breadboard, which necessitated finding alternative methods for creating secure and stable connections between components. This decision impacted our approach to hardware integration, requiring precise soldering techniques and potentially the design of custom PCBs (Printed Circuit Boards). Custom PCBs ensured reliable connections, improved durability, and reduced the risk of electrical faults that could compromise system performance.

Incorporating multiple sensors—such as pH, temperature, and moisture sensors—added complexity to our design. Each sensor had to be integrated with attention to power management, signal integrity, and data processing capabilities. The ESP-32-S microcontroller, known for its robust wireless communication capabilities, managed data transmission and connectivity. However, it required meticulous coding to handle sensor data efficiently, ensure real-time responsiveness, and maintain stable wireless communication.

On the Arduino side, we optimized the management of sensor inputs and the coordination of system actions, such as activating the water pump or adjusting lighting. This involved ensuring that the system operated reliably and efficiently, with careful calibration and testing to meet performance requirements.

The software development aspect introduced its own set of constraints. We created a user-friendly app interface that effectively communicated with the hardware components, ensuring compatibility across various devices and operating systems, as well as implementing robust security measures to protect against unauthorized access and data breaches. The app provided an intuitive experience while managing the system's operations and delivering real-time feedback to users.

Overall, addressing these constraints required a thoughtful approach to both hardware and software design. By carefully managing connections, optimizing sensor integration, and developing a secure and user-friendly app, we created a reliable and efficient automatic plant watering system that met all project objectives.

## 4.6 Time Constraints

Taking Senior Design I over the summer imposed significant time constraints on our project. The accelerated pace of the summer session meant we had limited time for thorough research and development, with deadlines rapidly approaching every two weeks. This condensed timeline required us to quickly conceptualize our idea, conduct initial research, and begin the design and prototyping process almost immediately.

One of the major time-related challenges was ordering and receiving parts. Sourcing components for our automatic plant watering system, such as sensors, microcontrollers, and other electronic components, could be time-consuming. Delays in shipping or availability disrupted our schedule, leaving limited time to assemble and test the system. Each delay demanded swift problem-solving and alternative planning to stay on track.

Moreover, the testing phase itself was time-intensive. Ensuring each component functioned correctly and integrated seamlessly with the rest of the system required meticulous testing and troubleshooting. Any issues that arose during testing had to be promptly addressed, which further compressed our already tight schedule. This iterative process of testing and refinement was critical for delivering a reliable and functional system, consuming a significant portion of our limited time.

In addition to technical challenges, the need to make crucial decisions quickly added another layer of complexity. With less time for deliberation and experimentation, we relied on efficient decision-making and problem-solving skills to navigate design constraints and unexpected obstacles. This often involved finding creative solutions to work around limitations, such as budget constraints or component availability, while still meeting project requirements and deadlines.

The time constraints of taking Senior Design I over the summer demanded a high level of efficiency and adaptability. The need to rapidly move from ideation to implementation, coupled with potential delays in parts procurement and the rigorous testing process, required us to be agile and resourceful. By staying organized and proactive, we overcame these challenges and successfully completed our automatic plant watering system project within the allotted timeframe.

## 4.7 Structure and environmental constraint

Our project consisted of various sensors, some of which were waterproof while others were not. This diversity in sensor types required a thoughtful approach to the structure of the design to ensure the protection of all components. Each sensor and electronic part were secured and isolated appropriately to prevent damage from soil or water exposure. For instance, waterproofing techniques were essential for components that might come into direct contact with water or damp soil, while other parts were encased or positioned in a way that kept them dry and functional.

The environmental constraints for our automatic plant watering system were relatively minimal, given that it was designed for houseplants. However, we still needed to address the requirement for adequate sunlight, which was crucial for plant health. In situations where the plant was placed away from natural light sources, such as windows, we implemented LED lighting. These LED lights provided the necessary light spectrum to promote healthy plant growth, ensuring that the plants received just the right amount of light. This approach helped us maintain plant health without relying solely on natural sunlight.

Additionally, our design accounted for the varying environmental conditions within a home setting. Factors such as room temperature and airflow could influence plant health and the performance of our system. By incorporating sensors to monitor these environmental variables, we adjusted our system's operation accordingly. For example, temperature sensors helped us fine-tune the watering schedule and lighting conditions to create an optimal environment for plant growth.

Finally, the structural and environmental constraints of our design require careful consideration and planning. By securing and protecting each component against potential damage from soil and water, and by addressing the plants' light needs through supplemental LED lighting, we created a robust and effective automatic plant watering system that ensured the well-being of houseplants in various home environments.

## 4.8 Power Supply Implementation

The power supply implementation was a critical aspect of our automatic plant watering system design, ensuring that all components operated reliably and efficiently. Given the various sensors and the ESP32-S microcontroller, a stable and sufficient power supply was essential to maintain system performance. A consistent power source was vital for the accurate functioning of the sensors that monitored soil moisture, pH levels, and water levels, as well as the microcontroller that processed this data and controlled the watering mechanisms. Without a reliable power supply, the system could fail to operate correctly, leading to inaccurate readings, missed watering schedules, and ultimately, potential damage to the plants. Therefore, careful consideration of power supply requirements, including voltage regulation and power distribution, was crucial to the success of our project.

The ESP32-S microcontroller and most sensors operated at specific voltage levels, often requiring 3.3V or 5V. To accommodate these requirements, we used voltage regulators like the LM7805, which stepped down the voltage from the main power source to the required levels. The LM7805 was known for its reliability and ability to provide a stable 5V output, which was crucial for maintaining the performance and longevity of the components. This ensured that each part of the system received the correct voltage, preventing overvoltage damage and ensuring optimal operation.

Efficient power distribution was essential for a well-functioning system. The power distribution board took the regulated voltage and distributed it to the various components of the system, including sensors, the microcontroller, and the water pump. By ensuring that each component received the appropriate voltage and current, we maintained the efficiency and effectiveness of the entire system. Additionally, using a centralized power distribution board simplified the wiring and helped manage the power flow more effectively.

Safety was a top priority in our power supply design. Overcurrent protection, short circuit protection, and thermal shutdown features were incorporated to prevent any electrical hazards. Fuses and current-limiting resistors protected the circuits from excessive current flow, while an insulated enclosure prevented accidental contact with live wires. These safety measures were essential to protect both the users and the system, ensuring safe and reliable operation.

## 4.9 I2C Protocol Standards

The Inter-Integrated Circuit (I2C) bus was a crucial communication protocol that underpinned the operation of numerous modern embedded systems, including our automatic plant watering system. Developed by Philips Semiconductor (now NXP Semiconductors), the I2C standard was designed for efficient, reliable, and simple communication between various electronic components, making it particularly well-suited for sensor and microcontroller interaction.

I2C was a serial communication protocol that employed two lines: the Serial Data Line (SDA) and the Serial Clock Line (SCL). These lines facilitated the bidirectional transfer of data between a master device (typically the microcontroller) and multiple slave devices (such as sensors and actuators). The master generated the clock signal on the SCL line, orchestrating the timing for data transfers, while the SDA line transmitted the actual data.

Each device on the I2C bus was assigned a unique address, allowing the master to communicate with individual devices without confusion. The addressing scheme in I2C could be either 7-bit or 10-bit, with 7-bit addressing being more common. The communication process began when the master device sent a start condition, followed by the 7-bit address of the target slave device and a read/write bit. The addressed slave acknowledged the master's request, and the data transfer could then proceed. This addressing and acknowledgment mechanism ensured that only the intended device responded, preventing data collisions and ensuring reliable communication.

Data transfer in I2C was byte-oriented, with each byte consisting of 8 bits followed by an acknowledgment (ACK) bit from the receiving device. The master device could initiate a read or write operation, depending on the state of the read/write bit sent during the addressing phase. After the transfer of each byte, the receiving device sent an ACK bit to confirm successful reception. This handshake ensured data integrity and allowed for error detection and correction. The communication could be terminated by the master sending a stop condition, indicating the end of the data transfer session.

I2C supported several clock speeds to accommodate different performance requirements: Standard Mode operated at up to 100 kHz, Fast Mode at up to 400 kHz, Fast Mode Plus at up to 1 MHz, and High-Speed Mode at up to 3.4 MHz. The choice of clock speed depended on the specific needs of the application and the capabilities of the devices on the bus. Higher speeds allowed for faster data transfers but required more stringent signal integrity considerations.

## 4.10 Voltage Regulator

The voltage regulator was a vital component in our automatic plant watering system, ensuring that all electronic parts received a stable and consistent voltage supply. In our design, we utilized the LM7805 voltage regulator, which was a linear regulator capable of providing a steady 5V output from a higher voltage input. This stability was crucial for the reliable operation of our sensors, microcontroller (ESP32-S), and other electronic components, which could be sensitive to voltage fluctuations.

The LM7805 voltage regulator operated by maintaining a fixed output voltage despite variations in the input voltage or changes in load conditions. It did this through a feedback mechanism that adjusted the internal resistance to keep the output voltage constant. The regulator's structure included a reference voltage, error amplifier, pass transistor, and feedback loop. When the input voltage fluctuated, the error amplifier detected the change and adjusted the pass transistor to stabilize the output. This design ensured that our system components received a consistent voltage, which was essential for accurate sensor readings and reliable microcontroller performance.

The structure of the voltage regulator also included protection features such as thermal shutdown and current limiting. Thermal shutdown protected the regulator and the system components from overheating by shutting down the regulator if it detected excessive temperature. Current limiting prevented damage caused by excessive current draw, which could occur during short circuits or other fault conditions. These protection features added an extra layer of safety to our system, ensuring that it could withstand various operational challenges without compromising performance.

Implementing the voltage regulator involved careful consideration of the input and output capacitor requirements. Capacitors stabilized the input and output voltages by filtering out noise and transient voltage spikes. Typically, an input capacitor was placed close to the regulator to smooth out the input voltage, while an output capacitor was placed close to the load to stabilize the output voltage. In our design, we ensured that these capacitors were correctly sized and positioned to maximize the effectiveness of the voltage regulation.

## 4.11 Wi-Fi

We used Wi-Fi connectivity for our microcontroller, which was a crucial aspect of our automatic plant watering system. By incorporating Wi-Fi, we enabled seamless communication between the system and the user's app, allowing for remote monitoring and control. This meant that users could manage their house plants from virtually anywhere, ensuring that their plants received the care they needed even when the user was away. With real-time updates and the ability to adjust settings from a distance, our system offered unparalleled convenience and flexibility. Users could check the status of their plants, monitor sensor data, and make necessary adjustments through an intuitive app interface. This connectivity not only enhanced the user experience but also ensured that the plants were consistently maintained in optimal conditions, promoting better plant health and growth.

The 802.11n standard, also known as Wi-Fi 4, operated on both the 2.4 GHz and 5 GHz frequency bands, offering data transfer rates up to 600 Mbps. This dual-band capability ensured robust performance in various environments, reducing interference and providing stable connections even in areas with many devices.

On the other hand, 802.11ac, known as Wi-Fi 5, operated predominantly on the 5 GHz band, delivering data rates up to several gigabits per second. This standard also supported a greater number of simultaneous connections through multi-user MIMO (Multiple Input, Multiple Output) technology, making it ideal for environments where numerous devices were connected to the network.

Implementing Wi-Fi in our microcontroller, specifically the ESP32-S, ensured that our system could handle the data-intensive tasks of monitoring multiple sensors, uploading data, and receiving commands from the user app efficiently. The Wi-Fi module was configured to maintain a secure and reliable connection, adhering to the latest security protocols such as WPA3, to protect user data and prevent unauthorized access.

## 4.12 Realistic Design Constraints

When we brainstormed for our automatic plant watering system project, we faced several realistic design constraints that influenced our decisions and approach. One of the primary constraints was the balance between accuracy and cost. High-accuracy sensors, particularly those that were waterproof, tended to be expensive and could easily exceed our budget. For instance, premium pH sensors designed for precise soil moisture and nutrient measurements were often priced prohibitively high. To address this, we sought to develop a cost-effective solution by exploring alternative sensor options and innovative methods to achieve similar results.

In our quest for an economical yet efficient design, we experimented with inexpensive sensors that, while not originally intended for our specific applications, could be adapted with creative engineering. For example, instead of using a high-cost soil pH sensor, we opted for a more affordable liquid pH sensor and devised a method to measure soil pH by mixing soil with water. This approach allowed us to maintain accuracy without breaching our financial limits, thus making our system accessible to a broader range of users.

Another critical aspect of our project was the development of an intuitive and comprehensive app to manage the watering system. While our vision included sophisticated features for real-time monitoring, data analysis, and remote control, we had to consider the constraints of our development timeline and technical resources. Developing an app with advanced functionalities required significant time, expertise, and testing, all of which had to be meticulously planned and executed within the scope of our project.

Additionally, the app needed to be user-friendly and compatible with various devices, adding another layer of complexity to our design process. We aimed to create a seamless user experience by integrating features such as a homepage for system overview, a data page for sensor readings, a camera page for visual monitoring, a water page for manual control, and a settings page for customization. Achieving this level of functionality while ensuring robust performance and security presented a substantial challenge that required careful planning and execution.

Moreover, the physical structure and environmental considerations of our design also imposed constraints. Ensuring that all components, particularly the sensors, were adequately protected from water and soil required thoughtful design and material selection. We needed to secure each sensor in a way that prevented damage while still allowing accurate readings, which involved designing custom enclosures and mounting solutions.

## 4.13 Constraints and Standards

Understanding and adhering to standards and constraints were pivotal in the design and implementation of our automatic plant watering system. Standards ensured that our system operated within the established guidelines for safety, functionality, and interoperability. By following relevant industry standards, such as those for electrical safety, wireless communication, and sensor accuracy, we guaranteed that our system was both reliable and compliant with regulations. This adherence not only minimized the risk of malfunctions but also enhanced the overall quality and user trust in our product.

Constraints, on the other hand, challenged us to optimize our design within defined limits such as budget, component availability, and technological capabilities. By addressing these constraints effectively, we developed innovative solutions that met our project goals while staying within our financial and technical boundaries. For example, selecting cost-effective sensors while maintaining performance standards ensured that our design remained affordable without compromising quality. Additionally, managing constraints related to time and resources helped us prioritize tasks and streamline the development process.

Incorporating both standards and constraints into our design process enabled us to create a well-balanced system that was both functional and feasible. It guided our decision-making, from component selection to system integration, ensuring that every aspect of our project aligned with practical and regulatory requirements. Ultimately, this approach enhanced the overall effectiveness and viability of our automatic plant watering system, making it a robust and dependable solution for users.

# 5. Comparison of ChatGPT with another Similar Platform

AI tools were a major focus in society recently with the advent of GPTs and other generative systems. These systems aimed to understand and answer questions or generate documents like a human would. Generative pre-trained transformers (GPTs), such as ChatGPT, were large language models (LLMs) that utilized a transformer structure. This made them appear as if they were able to understand and answer questions like a person could. This technology had been rapidly emerging in recent history, with the most well-known leader, OpenAI, being founded in December 2015. It had significant impacts on many industries, such as the business tool suite at Microsoft, which had added GPT to all its tools to summarize documents and auto-generate reports. It also had a large effect on education due to its ability to write essays and answer questions with convincing accuracy, as we would see in this chapter.

## 5.1 The Competition

Since the debut of ChatGPT, the landscape of AI-driven conversational tools evolved significantly with the introduction of various competitors. Initially pioneered by OpenAI, whose ChatGPT set the benchmark, several new players entered the market with distinctive offerings. Meta, for instance, launched LLama, positioning it as a lighter-weight, open-source alternative aimed at enhancing accessibility and customization for developers and users alike. LLama's approach emphasized efficiency and adaptability, catering to a diverse range of applications in AI interaction.

In contrast, Claude carved out a niche focusing on safety protocols to ensure more reliable and secure interactions. Its emphasis on safety measures aimed to enhance the quality and reliability of responses, addressing concerns about misinformation or inappropriate content in AI-generated interactions. This strategic focus on safety aligned with broader industry trends toward responsible AI deployment.

Meanwhile, Google's entry into the field with Gemini introduced a dynamic twist by leveraging online access to enhance the breadth and depth of information retrieval. Gemini's integration of real-time online searching capabilities promised a more dynamic and responsive user experience, catering to users' evolving needs and preferences in information retrieval.

These varied approaches reflected the growing diversification and specialization within the AI conversational tool market, each aiming to innovate and improve upon ChatGPT's foundational capabilities. As competition intensified, these advancements promised to drive forward new standards in AI-driven conversational interfaces, offering users a wider array of options tailored to specific use cases and preferences.

## 5.2 ChatGPT

ChatGPT was a GPT created by OpenAI and was released on November 30th, 2022. It was most notable for being the first major access to a GPT the world had seen. It had two variants: a free version powered by OpenAI's GPT-3.5 and a paid version powered by GPT-4. The estimated size of GPT-3.5 was 175 billion parameters, while GPT-4 was estimated to have 100 trillion. Recently, OpenAI had also been working on a GPT-4o that could handle text, audio, and visual inputs and outputs. OpenAI's GPT served as a major backbone for all of Microsoft's AI products as well (which is why their products were not being compared).

The pros of ChatGPT included its ability to generate text in a very convincing manner and to answer questions and create various text-based files with respectable accuracy. While doing this, it also supported many different languages, as well as programming languages. It could also adjust its writing style depending on the desired tone, allowing it to sound human or robotic as needed. Another major benefit of ChatGPT was its widespread usage, meaning there was no limit to the number of apps and UIs available for it if users preferred a different look. The official versions were accessible via website or phone app.

For all its impressive capabilities, it also had some glaring downsides. Firstly, it wasn't connected to the internet, meaning all its information was based on its last training date, which could be years behind depending on the version of GPT a user ran. Another issue was that it could exhibit bias based on the training data it used, which could lead it to seemingly support stereotypes. It could also engage in what was called "hallucinating," confidently providing incorrect information in a way that might seem knowledgeable on the subject.

## 5.3 LLama

LLama was a large language model (LLM) created by Meta (formerly Facebook). LLama 3, the most recent model, had 70 billion parameters. Despite the low parameter count, it was still capable of generating competitive outputs with other LLMs on the market. The biggest selling point for LLama 3 was that it was entirely open-source, allowing everyone to examine and run it on their own systems if they had sufficient processing power. This also meant users could adapt it for whatever specialization they desired.

The downsides to LLama 3 included the fact that, while it was competitive, it scored lower on tests than GPT-4. It also performed much worse at math and coding, making it less versatile. Another major downside was ease of use; LLama did not offer easy access via an app or website. Instead, the only way to access it was by downloading and running it on one's own computer or by renting a server that could run it. Additionally, it didn’t have constant access to the internet, so all information was based on the latest training data it used. Finally, like other LLMs, LLama suffered from bias and hallucinations.

## 5.4 Claude

Claude was a large language model (LLM) created by Anthropic AI. In many areas, such as general conversation and document writing, it was considered slightly worse than ChatGPT. However, in programming and ethics, it was regarded as the best option. Other benefits of Claude included its focus on safety, which was more pronounced than in its competitors. Its responses tended to be more generic but also more grounded compared to those of other LLMs.

The downsides to Claude were that versions competing with ChatGPT were not free and also lacked internet connectivity. This meant information was based on the latest training data available. While it supported multiple languages, its list was not as comprehensive as those of other competitors. Despite being safer, it remained susceptible to hallucinations and training data bias.

## 5.5 Google Gemini

Gemini was Google’s large language model (LLM). It was free with integration into Google and had a parameter count of around 7 billion. It garnered a lot of attention when it was first released due to giving harmful answers online to safety-critical questions. A benefit of Gemini was that it was integrated into the internet via Google, allowing it to utilize up-to-date information when answering questions. It also made searching Google easier by summarizing top results. Another benefit was that it integrated with all of Google’s product suite, enhancing productivity. This model was also much faster than other LLMs.

The downsides to Gemini were that the trade-off for its speed and smaller parameter size was that it was more inaccurate than its competition. This led to responses that didn’t make sense or were dangerous more often. While hallucination and training bias were issues for all LLMs, they were more prevalent in Gemini.

## 5.6 Impact on education in senior design

ChatGPT was a powerful tool, and like any tool, it could be used in both good and bad ways. In senior design, there were many ways ChatGPT could benefit or harm our experience and learning outcomes. The most important thing to mention was that ChatGPT couldn't make you learn something you didn't put effort into in the first place. A major way it could harm us in senior design was when people overutilized ChatGPT for everything. For example, if I wanted to create an app and just went to ChatGPT to ask it to generate the entire app for me without putting in the time and effort to understand any of it, there was a very high chance it wouldn't work, and some lines of code might be hallucinated. This was because LLMs struggled to handle massive workloads, and the larger the request, the higher the probability of error.

Let’s say I eventually managed to get the app to work just by asking an LLM. In senior design, we were there to learn how to design and showcase our skills to our professors and potential employers. If I generated a website without doing any of the design, I might have passed the class, but I would have cheated myself out of learning anything. This would have ultimately hurt me if I interviewed for a job and couldn’t answer questions because I wasn’t the one who did the actual work. This method of using LLMs was no different from mindlessly using an online calculator to plug and chug through a calculus course. There was also an issue if it was used for lower-level programming. Due to the nature of being closer to the hardware, there were small differences that ChatGPT wouldn't pick up on, especially if the user didn’t know about them and didn’t look out for them.

Another example of an LLM being harmful in senior design was if a student used ChatGPT to generate information personalized for their report or any substantial information. For example, if I just told it to research every common microcontroller unit (MCU) and pick the best one for me and write a report about it, this would generate a very convincing report that, upon first glance, seemed fine. However, upon inspection and research, it would most likely have been very flawed. This could have been due to the date of the model's last training. For instance, the most up-to-date training data for ChatGPT was from April 2023, which was now over a year old. If a new MCU had come out or had a refresh, the information would have been outdated. A real example of that was that many people did not recommend the ESP32-S3 in 2023 due to its lack of support at the time. But by the time I used it, the community had integrated it into the other ESP tools, so there was a chance ChatGPT would have given outdated information.

Another issue with blindly generating sections of a report was that ChatGPT didn’t understand every aspect of a project like a human did and didn’t ask clarifying questions. This could have led to hallucinations that made the report nonsensical because it couldn’t read the schematics of our design or fully understand our intent. A paragraph might have droned on about irrelevant topics instead of discussing the design we made and the specifications involved to make it work the way we intended. It could have also accidentally plagiarized sections without the user knowing. Finally, LLMs were trained on the work of others and required a lot of information to replicate something. This meant that the further you went from commonality, where there was a lot of data, the worse the results would have been, and the less it would have understood. This could have led to reports and programs being very repetitive and uncreative, which would have reflected poorly on students and been a detriment in senior design.

Despite the downsides of LLMs, they could have had an immense positive impact on productivity when used properly. They could significantly reduce repetitive sections of work that were easy to verify. For example, if you were writing a program and needed to implement a binary search algorithm, ChatGPT could generate the algorithm, and you could review it to ensure it was correct. This use case worked well because the function was a small program with a lot of training data examples available online. An important aspect was to make sure you reviewed that it worked, as even if it was correct 90% of the time, the last 10% could have caused a lot of debugging pain.

Another reason why this was helpful was that you could have used it for quick software prototyping. It could have been easier to direct ChatGPT to write up an outline of your code functions and header files. Because LLMs were not infallible, you didn't want to use them for critical tasks. In the drafting and outlining stages of a program, they could have been useful for quick writing that would undergo plenty of reviewing later.

A non-strictly technical use of LLMs that made them useful for senior design students was for writing reports. This could have been problematic if overutilized. An example of using it in a good way was if you had written a paragraph but struggled with wording it in an easy-to-read manner. Running the paragraph through ChatGPT to find grammar mistakes or reword it was fine, as long as ChatGPT's involvement was cited and the information was properly proofread to ensure it didn't change the meaning of the paragraph. This could have allowed students to write much faster and be more productive with their time in senior design. It could also have helped students whose first language wasn't English and who weren't as comfortable with grammar, especially given the 120-page requirement. This could have also assisted with writer's block; whenever you got stuck, you could have seen what the LLM suggested, which could have helped you get back on track much faster.

Another aspect of using ChatGPT for senior design was during the early stages of brainstorming projects. Coming up with a project on the spot could have been difficult if you hadn't been thinking about it before class. Using ChatGPT to generate a variety of ideas and asking it different questions could have helped prime your thinking. The important thing to note was to build off the ideas and not just copy ChatGPT directly, ensuring the idea remained unique.

## 5.7 Our usage of LLMs

After everything we covered above, our team decided that generative AI was a beneficial tool to utilize during Senior Design 1. Its utility in report writing, when used correctly, greatly improved the quality of our documents. Our team used generative AI by first writing the content we intended to include in our paragraph, ensuring all important information was covered. We then copied our section of text into the AI and asked it to rewrite the paragraph in a more accessible way. We proofread everything the AI generated, ensuring the information matched our intentions, often making considerable changes. However, we felt this tool increased both the quality of our writing and the speed at which we moved through the writing process.

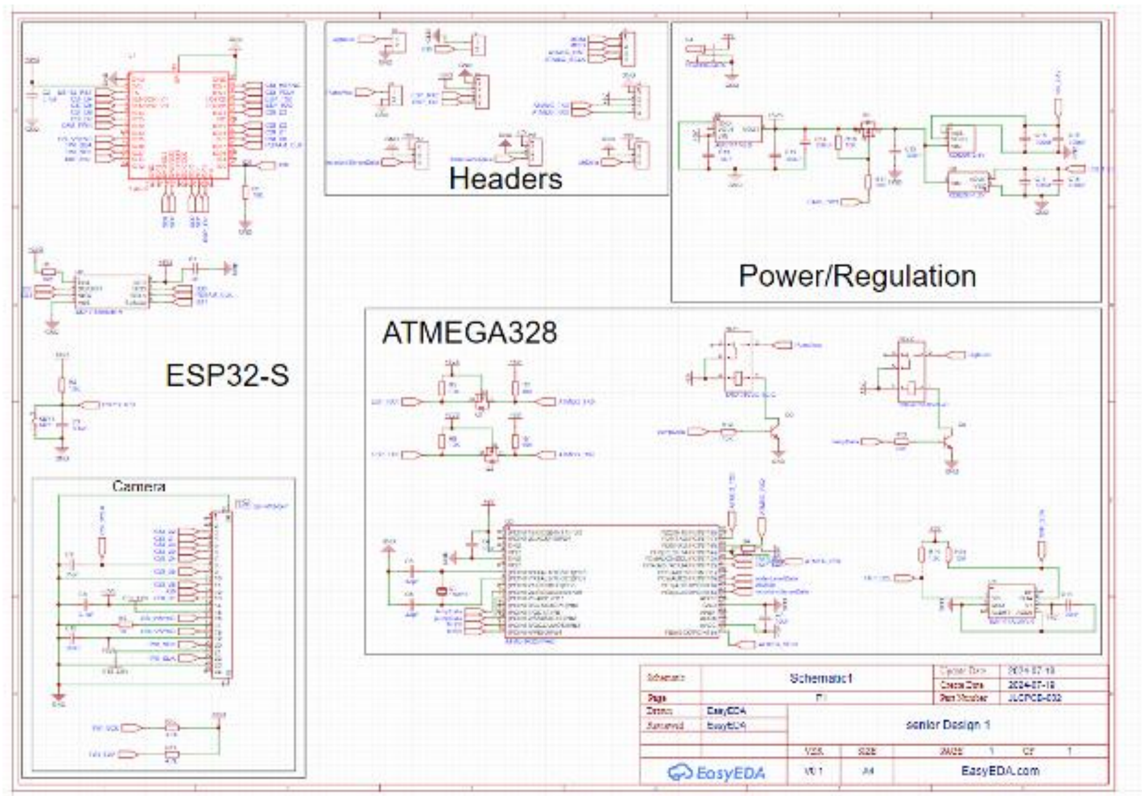
In choosing which generative AI to use, we opted for OpenAI's free version, ChatGPT. This decision stemmed from its ease of use, given its website access, and its high performance compared to the competition.

Reflecting on the factors discussed earlier, our team concluded that leveraging generative AI proved to be a valuable asset in our approach to Senior Design 1. Specifically, its application in report writing demonstrated significant potential to enhance the overall quality of our documentation. Our methodology involved initially drafting paragraphs with all essential information meticulously included. These drafts were then input into the generative AI tool, prompting it to rephrase the content into a more accessible and reader-friendly format. Subsequently, we undertook a thorough review and editing process to ensure the AI-generated output accurately conveyed our intended information, often necessitating substantial adjustments. Nevertheless, integrating generative AI into our workflow notably improved both the precision of our writing and the efficiency with which we navigated the writing process.

Looking ahead, the incorporation of generative AI into our Senior Design 1 activities represented a strategic enhancement in our project management and communication capabilities. By harnessing advanced AI technology like ChatGPT, we anticipated continued improvements in the clarity, accuracy, and efficiency of our written outputs. This evolution underscored our commitment to leveraging innovative tools to optimize our project outcomes and deliver impactful results within our academic and professional endeavors, aligning perfectly with the very goal of Senior Design 1.

# 6. Hardware Design

*Figure 6.1: Overall schematic*




Headers:

The success of our smart water planter hinges on the seamless integration of various electronic components, and the headers are central to this integration. Headers act as critical junctions that connect different parts of the system, enabling the transfer of power, data, and control signals between the microcontroller and various sensors, actuators, and peripheral devices. These headers ensure that the components work harmoniously together, facilitating a well-functioning and efficient smart water planter.

Header 1 is designed to connect the ESP8266 WiFi module to the microcontroller, which is a fundamental aspect of enabling remote control and monitoring of the smart water planter via WiFi. The ESP8266 is essential for providing the wireless connectivity that allows users to interact with the planter through a app or web interface. In this header, Pin1 (GND) and Pin2 (+3V) supply the necessary power to the ESP8266, ensuring it operates within its required voltage range. Pins 3 (ESP_RX0) and 4 (ESP_TX0) facilitate serial communication between the ESP8266 and the microcontroller. This serial connection is crucial for exchanging data such as sensor readings and control commands, enabling the system to operate based on user input and environmental conditions.

Header 2 provides additional control functionality for the ESP8266. Pin1 (GND) is used for grounding, while Pin2 (IO0) connects to a general-purpose input/output (GPIO) pin on the ESP8266. This GPIO pin can be utilized for various control purposes, such as resetting the ESP8266 or putting it into firmware upload mode. This flexibility ensures that the ESP8266 can be effectively managed and reprogrammed as needed for firmware updates or maintenance tasks, which is critical for maintaining the system's functionality and adaptability.

Header 3 connects the ATMEGA328 microcontroller to the SPI (Serial Peripheral Interface) bus, which is vital for high-speed communication between the microcontroller and other SPI-compatible devices. This header includes Pin1 (MOSI - Master Out Slave In), Pin2 (MISO - Master In Slave Out), Pin3 (ATMEG_RST - Reset), and Pin4 (ATMEG_SCLK - Serial Clock). The MOSI and MISO pins facilitate the transfer of data between the master (microcontroller) and slave devices, while the ATMEG_RST pin allows for resetting the microcontroller if needed. The ATMEG_SCLK pin provides the clock signal necessary for synchronizing data transfer. This SPI interface is crucial for efficient data exchange and system synchronization, enabling the microcontroller to communicate with various peripherals and sensors.

Header 4 is dedicated to UART (Universal Asynchronous Receiver/Transmitter) communication for the ATMEGA328 microcontroller. This header includes Pin1 (GND) and Pin2 (+5V) for power and grounding, while Pin3 (ATMEG_TX0) and Pin4 (ATMEG_RX0) handle serial data transmission and reception. UART communication is essential for interacting with devices such as the ESP8266 or a serial monitor, allowing for debugging, data logging, and control functions. This header ensures that the ATMEGA328 can effectively communicate with other system components and external devices, facilitating smooth operation and user interaction.

Header 5 connects the pH sensor to the microcontroller. This header includes Pin1 (+5V) for powering the sensor, Pin2 (GND) for grounding, and Pin3 (pH Data) for transmitting the sensor's output. The pH sensor is crucial for monitoring soil acidity, as different plants require specific pH levels to thrive. By integrating this header, the smart water planter can continuously monitor the pH levels and adjust watering or nutrient delivery accordingly. This ensures that the plants receive optimal growing conditions and helps maintain their health and productivity.

Header 6 is used for connecting the water level sensor to the microcontroller. Pin1 (+5V) supplies power to the sensor, Pin2 (GND) provides grounding, and Pin3 (Water Level Data) transmits the sensor's readings. The water level sensor is essential for monitoring the water reservoir and ensuring that the water pump operates efficiently. By integrating this header, the system can prevent the pump from running dry and maintain a consistent water supply for the plants. Accurate water level monitoring is critical for preventing water wastage and ensuring that the plants receive adequate hydration.

Header 7 connects the soil moisture sensor to the microcontroller. Pin1 (+5V) powers the sensor, Pin2 (GND) grounds it, and Pin3 (Moisture Sensor Data) carries the sensor's output. Soil moisture monitoring is a key feature of the smart water planter, as it helps ensure that plants receive the right amount of water without overwatering. By integrating this header, the system can collect precise moisture data and use it to control the water pump, maintaining optimal soil conditions and preventing water-related issues.

Header 8 is dedicated to controlling the LED lighting system. Pin1 (Light) connects to the control circuit for the LED lights, while Pin2 (GND) provides grounding. Proper lighting is essential for plant growth, and this header allows the microcontroller to regulate the intensity and duration of light exposure. By using this header, the LED lights can be adjusted based on the time of day or specific plant requirements, ensuring that plants receive the optimal light conditions for photosynthesis and growth.

Header 9 connects the water pump to the microcontroller. Pin1 (Pump) is used to control the operation of the water pump, and Pin2 (GND) grounds it. This header is crucial for automating the watering process, as the microcontroller can activate the pump based on signals from the moisture and water level sensors. This automation ensures that the plants receive precise amounts of water as needed, maintaining consistent hydration and preventing overwatering or under-watering.

In conclusion, the headers in our smart water planter project play a pivotal role in integrating and coordinating various electronic components. Each header is designed to fulfill specific functions, from power distribution and communication to sensor data collection and actuator control. By carefully designing and implementing these headers, we ensure that the smart water planter operates efficiently and effectively, providing users with a reliable and user-friendly gardening solution. The thoughtful integration of these components not only enhances the system's functionality but also contributes to the overall health and vitality of the plants, making modern indoor gardening more accessible and enjoyable.

**ESP32:**

The ESP32-S chip serves as the brain of the smart water planter project, providing essential processing power and connectivity to ensure the system's functionality and efficiency. This microcontroller is equipped with WiFi and Bluetooth capabilities, enabling remote monitoring and control of the planter through a dedicated application. The integration of the ESP32-S allows users to interact with their plants from anywhere, receiving real-time updates and making adjustments as needed. This connectivity is crucial for modern urban gardeners who may not have the time to constantly monitor their plants in person but still want to ensure their plants are thriving.

One of the primary roles of the ESP32-S is to manage data input from various sensors integrated into the smart planter. These sensors include those monitoring moisture levels, temperature, pH levels, and water levels. By continuously collecting and processing data from these sensors, the ESP32-S ensures that plants receive optimal care. For instance, if the soil moisture drops below a certain threshold, the ESP32-S can trigger the water pump to hydrate the plants automatically. This automatic response is vital for maintaining consistent growing conditions, particularly for plants that require precise care.

In addition to sensor management, the ESP32-S controls the lighting system of the smart planter. Using LED bulbs tailored for plant growth, the microcontroller adjusts the light intensity and duration based on the specific needs of the plants. This ensures that plants receive the right amount of light for photosynthesis, promoting healthy growth. The control over lighting can also be adjusted remotely via the app, allowing for flexibility in managing different plant species with varying light requirements. This feature is especially useful for gardeners growing a variety of plants with diverse lighting needs, as it allows for customized care settings for each plant type.

The ESP32-S also plays a crucial role in image processing for advanced plant health monitoring. By interfacing with a camera, the microcontroller captures images of the plants and analyzes them to detect signs of disease, pests, or nutrient deficiencies. This image analysis feature enables proactive plant care, allowing users to address potential issues before they become severe. The ability to detect and diagnose plant health issues early on can significantly improve the overall success rate of indoor gardening, as users can take corrective actions swiftly.

Furthermore, the ESP32-S facilitates communication between the various components of the smart planter. It coordinates the actions of the water pump, sensors, and lighting system, ensuring they work harmoniously to provide the best possible environment for plant growth. This integrated approach simplifies the user experience, making it easy to manage the planter through a single interface. The seamless interaction between components reduces the need for manual intervention, thus saving time and effort for the user.

The use of the ESP32-S also enhances the scalability of the project. With its powerful processing capabilities, additional features such as weed detection and plant type recognition can be implemented. These advanced features further enhance the functionality of the smart planter, making it a comprehensive solution for modern gardening. Weed detection can help maintain a clean and productive garden space by alerting users to unwanted plant growth, while plant type recognition allows the system to provide tailored care recommendations based on the specific needs of each plant species.

Moreover, the ESP32-S supports the implementation of a notification system through the app. Users can receive alerts about various conditions, such as low water levels, abnormal temperature fluctuations, or the detection of plant health issues. These notifications ensure that users are always informed about the status of their plants, enabling timely interventions when necessary. This feature is particularly beneficial for busy individuals who may not have the time to check the app frequently but still want to ensure their plants are well-cared for.

The ESP32-S also allows for the development of a scheduling system for automatic watering. By setting specific watering times, users can ensure that their plants receive consistent hydration without the need for manual intervention. This feature is especially useful for maintaining plant health during periods when the user is away from home, such as during vacations or business trips. The ability to automate watering schedules adds an extra layer of convenience and reliability to the smart planter system.

In summary, the ESP32-S is integral to the smart water planter project, providing the necessary processing power, connectivity, and control to manage all aspects of the system. Its ability to handle real-time data, automate plant care, and enable remote monitoring makes it an ideal choice for creating an efficient and user-friendly smart gardening solution. By leveraging the capabilities of the ESP32-S, the smart water planter can deliver optimal plant care, improving the health and productivity of indoor gardens. The chip's versatility and advanced features ensure that the smart planter can meet the needs of modern gardeners, making indoor gardening more accessible and effective for a wide range of users.

**ESP_PSRAM64H:**

The ESP PSRAM64H chip plays a critical role in the smart water planter project by providing additional memory resources necessary for handling complex tasks such as image processing and data storage. This external PSRAM (Pseudo Static RAM) significantly enhances the capabilities of the ESP32-S microcontroller, allowing it to manage more data and run more sophisticated algorithms without running into memory constraints. The integration of PSRAM is particularly important for advanced features like plant health monitoring and image recognition, which require substantial memory for processing and storing high-resolution images and sensor data.

One of the primary functions of the PSRAM64H is to support the image processing capabilities of the smart planter's camera system. As the camera captures images of the plants, the PSRAM provides the necessary memory space to store these images temporarily while the ESP32-S processes them. This is essential for performing real-time analysis to detect signs of disease, pests, or nutrient deficiencies. By having additional memory, the system can handle larger image files and perform more detailed analyses, improving the accuracy and reliability of the health monitoring feature.

The PSRAM64H also aids in managing sensor data collected from various sensors integrated into the planter. These sensors, which monitor moisture, temperature, pH levels, and water levels, generate a continuous stream of data that needs to be processed and stored. The additional memory provided by the PSRAM allows the ESP32-S to store more data for longer periods, enabling better trend analysis and more informed decision-making. This is particularly useful for implementing advanced features like automatic watering schedules based on historical data trends.

In terms of power management, the PSRAM64H is designed to operate efficiently with the ESP32-S. The chip operates on a 3V power supply, which is compatible with the ESP32-S and ensures low power consumption, a critical factor for battery-operated devices. The integration involves connecting the power pin to a 3V source in parallel with a capacitor, which helps to stabilize the power supply and filter out any noise, ensuring reliable operation of the memory chip.

Implementing the PSRAM64H into the smart water planter's PCB design involves careful consideration of the connection pins and their roles. For instance, the CE# (Chip Enable) pin is connected to a 10k ohm resistor to ground, which helps in controlling the activation of the PSRAM. The data lines, such as SD0, SD1, SD2, and SD3, facilitate the transfer of data between the PSRAM and the ESP32-S, enabling fast and efficient data handling. The SCLK (Serial Clock) pin ensures synchronized data communication between the two chips, while the ground (VSS) pin provides a common reference point for the circuit.

The use of external PSRAM, like the PSRAM64H, offers significant advantages over relying solely on the internal memory of the ESP32-S. One major benefit is the ability to handle more complex tasks and larger datasets without compromising system performance. This is crucial for a smart planter that aims to offer advanced features such as real-time plant health monitoring, automatic watering, and sophisticated data analysis. By offloading memory-intensive tasks to the PSRAM, the ESP32-S can focus on processing and decision-making, resulting in a more efficient and responsive system.

Another advantage of incorporating PSRAM is the flexibility it provides in expanding the system's capabilities. As new features and functionalities are developed, the additional memory ensures that the system can accommodate these enhancements without requiring significant hardware changes. This scalability is important for future-proofing the smart planter, allowing it to evolve and improve over time with software updates and new sensor integrations.

In summary, the ESP PSRAM64H is an essential component of the smart water planter project, providing the additional memory required to support advanced features and efficient data management. Its integration with the ESP32-S enhances the system's overall performance, enabling real-time image processing, comprehensive sensor data handling, and reliable power management. By incorporating PSRAM, the smart planter can deliver a more robust and feature-rich user experience, making indoor gardening more accessible and effective for modern urban gardeners.

**Camera:**

The camera module (FRC1/FPC0.52H-WS-24P) integrated into the smart water planter is essential for advanced features such as plant health monitoring, weed detection, and plant type recognition. This camera module provides the visual data necessary for these sophisticated functionalities, enabling the system to capture high-resolution images of the plants and analyze them for various indicators of plant health. The camera's integration is carefully designed to ensure reliable operation and efficient data transfer to the ESP32-S microcontroller.

One of the key roles of the camera module is to monitor plant health by capturing detailed images of the plants. These images can then be analyzed to detect signs of diseases, pests, or nutrient deficiencies. The camera's high-resolution capabilities allow for precise identification of issues that may not be visible to the naked eye. This proactive monitoring helps users take timely action to maintain plant health and prevent potential problems from escalating.

The camera module also plays a crucial role in weed detection. By using image recognition technology, the system can distinguish between desired plants and unwanted weeds. This feature helps maintain a clean and productive garden space, reducing the manual effort required for weed management. The camera captures images of the garden area, and the system analyzes these images to identify and alert users to the presence of weeds.

Another important application of the camera module is plant type detection. By recognizing different plant species, the system can provide tailored care recommendations based on the specific needs of each plant type. This personalized approach to plant care ensures that each plant receives the optimal conditions for growth, improving overall garden productivity and plant health.

The integration of the camera module into the PCB design involves several key components and connections. The camera's data lines, such as CSI_D0 through CSI_D7, are responsible for transferring image data to the ESP32-S for processing. The PCLK (Pixel Clock) line, connected in series with a 15pF capacitor, ensures synchronized data transfer, while the HSYNC (Horizontal Sync) and VSYNC (Vertical Sync) lines manage the timing of the image capture process.

Power management is critical for the reliable operation of the camera module. The camera requires multiple voltage levels, including 3V, 2.8V, and 1.2V. These voltages are supplied with appropriate capacitors to stabilize the power supply and filter out noise. For instance, the 3V line is connected in parallel with a 0.1uF capacitor to ensure a steady power supply, while the 2.8V line is similarly managed. The inclusion of ground pins and resistors helps to further stabilize the system and prevent electrical interference.

The TWI_SCK and TWI_SDA lines are used for the I2C communication interface, allowing the camera module to communicate with the ESP32-S microcontroller. This interface is crucial for configuring the camera settings and transferring control data. By using the I2C protocol, the system can easily manage the camera's operations and integrate it seamlessly with other sensors and components.

In summary, the camera module (FRC1/FPC0.52H-WS-24P) is a vital component of the smart water planter, enabling advanced features like plant health monitoring, weed detection, and plant type recognition. Its integration into the PCB design is carefully managed to ensure reliable power supply, efficient data transfer, and effective communication with the ESP32-S microcontroller. By capturing high-resolution images and providing detailed visual data, the camera module significantly enhances the functionality and user experience of the smart planter, making it a powerful tool for modern gardening.

A power regulation circuit is a crucial component in any electronic project, especially in a complex system like a smart water planter. Its primary purpose is to ensure that all components receive a stable and consistent voltage, which is vital for the reliable operation of the entire system. Power regulation helps to protect sensitive electronics from voltage fluctuations that could cause malfunction or damage.

In the context of our smart water planter project, the power regulation circuit performs several important functions. First, it converts the incoming power supply from an external source, such as a wall adapter or battery, to the specific voltage levels required by different components of the system. For instance, the ESP32-S microcontroller might require a 3.3V supply, while other sensors and the camera might need 5V or other voltage levels. The power regulation circuit ensures that each component receives the correct voltage, thereby optimizing performance and preventing damage.

One of the critical elements of the power regulation circuit is the voltage regulator. Voltage regulators can be linear or switching. Linear regulators are simple and provide a stable output voltage but can be inefficient as they dissipate excess power as heat. Switching regulators, on the other hand, are more efficient as they use an inductor, capacitor, and switch to convert one voltage to another with minimal loss. In our project, using a switching regulator would be advantageous due to its efficiency, especially if the system is powered by batteries where energy conservation is paramount.

The power regulation circuit also includes various capacitors and inductors that filter out noise and smooth the output voltage. These components help to eliminate voltage spikes and dips that could interfere with the operation of sensitive components like the microcontroller and sensors. By providing a clean and stable power supply, the power regulation circuit enhances the reliability and longevity of the entire system.

Another important aspect of the power regulation circuit is its ability to handle varying input voltages. For instance, if the smart water planter is powered by a solar panel, the input voltage can fluctuate based on the amount of sunlight. The power regulation circuit must be able to adapt to these changes and still provide a consistent output voltage. This capability is particularly important for outdoor or remote applications where the power supply might not be stable.

Moreover, the power regulation circuit plays a critical role in the safety of the system. Over-voltage and under-voltage conditions can be harmful to electronic components. The regulation circuit can include protection features such as over-voltage protection (OVP) and under-voltage lockout (UVLO) to ensure that the components operate within safe voltage ranges. This protection helps to prevent potential damage and ensures the longevity of the smart planter.

Integrating the power regulation circuit into the PCB design requires careful consideration of layout and component placement. Proper grounding and decoupling techniques are essential to minimize electrical noise and ensure stable operation. Placing capacitors close to the power pins of critical components can help to filter out high-frequency noise and provide a stable voltage supply.

In summary, the power regulation circuit is a fundamental component of the smart water planter, ensuring that all electronic components receive a stable and consistent voltage supply. It converts the input power to the required voltage levels, filters out noise, and provides protection against voltage fluctuations. By maintaining a reliable power supply, the power regulation circuit enhances the performance, safety, and longevity of the smart planter, making it a robust and efficient system for modern gardening applications.

.The ESP32-S microcontroller is the central hub of the smart water planter system, responsible for processing data and managing the various components integrated into the project. With its built-in WiFi and Bluetooth capabilities, the ESP32-S enables remote monitoring and control of the planter through a application or web interface. This microcontroller handles inputs from various sensors, including temperature, pH, moisture, and water level sensors, processing the data to make real-time decisions. It also controls output devices like the water pump and LED lights, adjusting their operation based on the sensor data. By managing these components, the ESP32-S ensures that the plants receive optimal care, maintaining consistent environmental conditions.

**PSRAM64H (External Pseudo Static RAM)**

The PSRAM64H chip provides additional memory resources to the ESP32-S, crucial for handling complex tasks such as image processing and extensive data management. This external memory supports the microcontroller in storing and processing high-resolution images from the camera module, as well as managing large datasets from multiple sensors. The increased memory capacity allows the ESP32-S to perform more sophisticated analyses and run advanced algorithms without being constrained by limited internal memory. This capability is essential for features like plant health monitoring and detailed environmental control, making the system more robust and capable of delivering precise care for the plants.

**Camera Module (FRC1/FPC0.52H-WS-24P)**

The camera module plays a pivotal role in the smart water planter by capturing high-resolution images of the plants. These images are analyzed to monitor plant health, detect signs of diseases, pests, or nutrient deficiencies, and identify weeds. The camera's integration with the ESP32-S allows for real-time image processing, which is essential for providing timely care and intervention. By interfacing with the PSRAM64H, the camera's data can be stored and processed efficiently, ensuring accurate and detailed monitoring of plant conditions. This visual data enhances the system's ability to manage plant health proactively and automate care routines based on visual cues.

**Temperature Sensor**

The temperature sensor measures the ambient temperature around the plants, providing critical data to maintain optimal growing conditions. By integrating this sensor with the ESP32-S, the system can monitor temperature fluctuations and adjust the environment accordingly. For example, if the temperature deviates from the ideal range, the microcontroller can trigger fans or heaters to regulate the temperature. This capability ensures that plants are kept within their preferred temperature range, promoting healthy growth and preventing stress-related issues. The temperature data is also used in conjunction with other sensors to provide a comprehensive environmental control system.

### Power Supply

The power supply is fundamental to the operation of the smart water planter, as it provides the necessary voltage and current to all electronic components. Depending on the design, it might consist of an external wall adapter, battery, or solar panel. The power supply ensures that each component, including the ESP32-S, sensors, and actuators, receives the appropriate voltage levels required for proper operation. A stable and reliable power supply is crucial for maintaining consistent performance and avoiding system failures. It directly impacts the efficiency of the power regulation circuit, which converts and stabilizes the input power to meet the needs of various components.

### Power Regulation Circuit

The power regulation circuit ensures that all components of the smart water planter receive a stable and consistent voltage. It converts the incoming power from the power supply to the specific voltage levels required by different parts of the system, such as the ESP32-S, sensors, and camera module. This circuit typically includes voltage regulators, capacitors, and inductors to filter out noise and stabilize the output. By providing a clean and regulated power supply, the power regulation circuit prevents damage to sensitive components and enhances the overall reliability of the system. It ensures that voltage fluctuations do not affect the operation of the smart planter, contributing to its long-term stability and performance.

### pH Sensor

The pH sensor measures the acidity level of the soil, which is a critical factor for plant health. By integrating the pH sensor with the ESP32-S, the system can continuously monitor soil conditions and ensure they remain within the optimal range for plant growth. The data from the pH sensor is used to adjust the watering or nutrient delivery system, helping to maintain balanced soil conditions. This continuous monitoring allows for proactive management of soil pH, which is essential for preventing nutrient deficiencies and promoting healthy plant development. The pH sensor's integration ensures that the smart water planter provides precise and tailored care for different plant species.

### Water Level Sensor

The water level sensor monitors the amount of water in the reservoir, ensuring that the system maintains an adequate water supply for the plants. By providing data on the water level to the ESP32-S, the sensor enables the microcontroller to control the water pump effectively. If the water level drops below a certain threshold, the ESP32-S can activate the pump to refill the reservoir, preventing the plants from running dry. This automated control helps maintain consistent hydration and prevents overflows or dry spells. The water level sensor is crucial for the efficient operation of the watering system, contributing to the overall health and stability of the plants.

### Soil Moisture Sensor

The soil moisture sensor measures the amount of moisture present in the soil, allowing the system to manage watering based on actual soil conditions. Integrated with the ESP32-S, the soil moisture sensor provides real-time data on soil hydration levels, enabling the microcontroller to regulate the water pump's operation. By ensuring that the soil is neither too dry nor too wet, the system prevents overwatering or underwatering, which can harm plant health. The soil moisture data helps in creating an optimal watering schedule, tailored to the specific needs of the plants. This precise control of soil moisture contributes to healthier plants and more efficient water usage.

### Water Pump

The water pump is a critical actuator in the smart water planter, responsible for automating the watering process. Controlled by the ESP32-S based on input from the soil moisture and water level sensors, the pump ensures that plants receive the correct amount of water. When the soil moisture falls below a predefined threshold or the water level in the reservoir is low, the microcontroller activates the pump to deliver water to the plants. This automation helps maintain consistent hydration, reduces the need for manual watering, and optimizes water usage. The water pump's integration with the sensor system ensures that plants are kept in optimal conditions with minimal user intervention.

# 7. Software Design

## 7.1 App Considerations

There were many considerations to take into account before beginning any type of implementation. Important factors such as accessibility, scalability, and necessity were carefully evaluated before realizing a final product. The goal of this project was to be able to automatically sustain a plant through occasional usage of an app. The aim was to create a hands-off environment that took away the difficulty of raising a plant, ensuring that users can maintain healthy plants with minimal effort.

To achieve this, the application needed to read and display sensor data as well as carry out the automatic tasks that sustained a plant. The tasks that were conducted were activated based on the conditions of the plant as determined by the sensors. The sensors used included a pH sensor, moisture sensor, water level sensor, and temperature sensor. Without an application, there would not have been a way for the user to utilize the information being tracked by the sensors.

The application had to ensure that the data from the sensors was accurately captured, processed, and displayed in real time to provide users with the most up-to-date information about their plants. This required a reliable and robust data handling mechanism that could efficiently manage the incoming data streams from multiple sensors simultaneously. The backend system needed to be capable of processing the sensor data, triggering necessary actions like watering or adjusting environmental controls, and sending notifications to the user.

Additionally, the user interface needed to be intuitive and user-friendly, allowing users to easily monitor their plant's health and status. This included clear visualizations of sensor data, such as graphs and charts, and easy-to-understand notifications about the plant’s needs. The application also needed to provide historical data tracking, enabling users to see trends and make informed decisions about their plant care routines.

Scalability was another possible consideration. The application had to be designed to potentially handle an increasing number of sensors without compromising performance. This meant choosing a technology stack that supported scalable architecture and could accommodate future growth. Ensuring that the app would integrate with various sensor types and models was essential for its versatility and adaptability to different plant care setups.

Finally, the necessity of the application was evaluated based on user needs. Thorough research was conducted to understand the pain points of plant care and how the application could address them, helping to create a product that truly added value to the users' lives. User feedback and testing were integral parts of the development cycle to refine the application and ensure it met the intended goals effectively.

## 7.2 App Interactions and Charts

The Plant Guardian app was designed to provide users with an intuitive and seamless experience for monitoring and maintaining the health of their plants. The app interface was organized into several key pages, each serving a specific function. Below is an overview of the main functionalities of the app.

### User Accessibility

User accessibility was a critical component in the design and development of the Plant Guardian app. The commitment was to ensure that the app was usable by all individuals, regardless of level of technical skills. To achieve this, the app incorporated several accessibility features aimed at enhancing the overall user experience.

### Text Color Contrast

One of the primary accessibility features of the Plant Guardian app was ensuring that text color contrast was sufficient for visibility. This was particularly important for users with visual impairments, including those with low vision or color blindness. The app used color contrast ratios that complied with accessibility standards, making text easily readable against its background. This included the use of dark text on light backgrounds and vice versa to ensure that all users could read information clearly without strain.

### Interface Clarity

The app's interface was designed for clarity and simplicity to accommodate users. This included clear labels and straightforward icons that helped users understand how to use the app effectively. The layout is clean and uncluttered, reducing cognitive load and making it easier for users to find the information and features they needed. The design prioritized ease of use, ensuring that users could navigate through the app.

### Intuitive Navigation

Navigating the Plant Guardian app was straightforward, with a simple press and no pop out menus. Users are able to move between the home screen, sensor data page, camera page, watering schedule, and settings without confusion. Each page has clear and distinct labels, making it easy for users to know where they were and how to access different functionalities.

### Visual Feedback

Visual feedback was integrated throughout the app to inform users of actions taken. For example, when a user manually waters the plant through the app, a update message appears, ensuring that the action was successful. This immediate feedback helped users understand the app's responses to their inputs, enhancing the overall user experience.

By incorporating these accessibility features, the Plant Guardian app ensured that it was inclusive and usable by a diverse range of users. This commitment to accessibility not only improved the user experience but also broadened the app's appeal and usability, making it a valuable tool for anyone interested in maintaining healthy plants with minimal effort.

### App Functionality

The app includes a navigation bar, user notifications, data logging, and manual control. Users could move to each app feature with a button click. From each page, they were able to view the sensor data or the plant through a camera feed. User notifications were routinely sent out while the app was open, depending on certain conditions of the plant. Notifications alerted users when the plant needed watering, and protection systems were in place to mitigate the risk of overwatering. The system tracked important information, such as the last time the plant was watered. The manual control functionality was designed to allow users to water the plant against the system's recommendations. Users could manually activate the watering system or toggle features such as lighting.

### Home Page

The Home page acted as the welcome screen of the app. It provided users with a friendly introduction and allowed them to choose which page to navigate to.

### Sensor Page

The Sensor page displayed real-time data, polled every thirty seconds, from various sensors monitoring the plant's environment. Key metrics shown on this page included current moisture, temperature, water level, and pH. These readings allowed users to quickly assess the condition of their plant and understand any immediate needs. The data was presented in a clear and concise manner, making it easy for users to interpret.

### Camera Page

The Camera page provided a visual check-in feature, allowing users to see their plant through an integrated camera. The camera feature was facilitated by the ESP32's camera, which sent information through the database. This page included a button to toggle lighting, helping users view the plant more clearly and assess its condition. This feature added an extra layer of monitoring by offering both quantitative sensor data and qualitative visual feedback.

### Water Page

The Water page was dedicated to managing the plant's watering schedule. It displayed the last time the plant was watered and provided a button for manual watering. It also included an immediate check to compare the plant's current condition to its preferred conditions. This functionality ensured that users could track watering intervals and intervene manually if necessary, offering flexibility and control over the plant's care routine.

### Settings Page

The Settings page allowed users to select a preset for their plant. Users could choose their plant type and apply recommended sensor levels accordingly. By configuring these settings, users customized the automated care routines to fit the specific needs of their plants. This page ensured that the application catered to a wide variety of plants with different requirements, enhancing the app's versatility and user satisfaction.

Each of these pages worked together to provide a comprehensive plant care system, combining real-time sensor data, visual monitoring, manual and automated watering controls, and customizable settings. The integration of these functionalities created a hands-off approach to plant care, making it easier for users to maintain healthy plants with minimal effort.

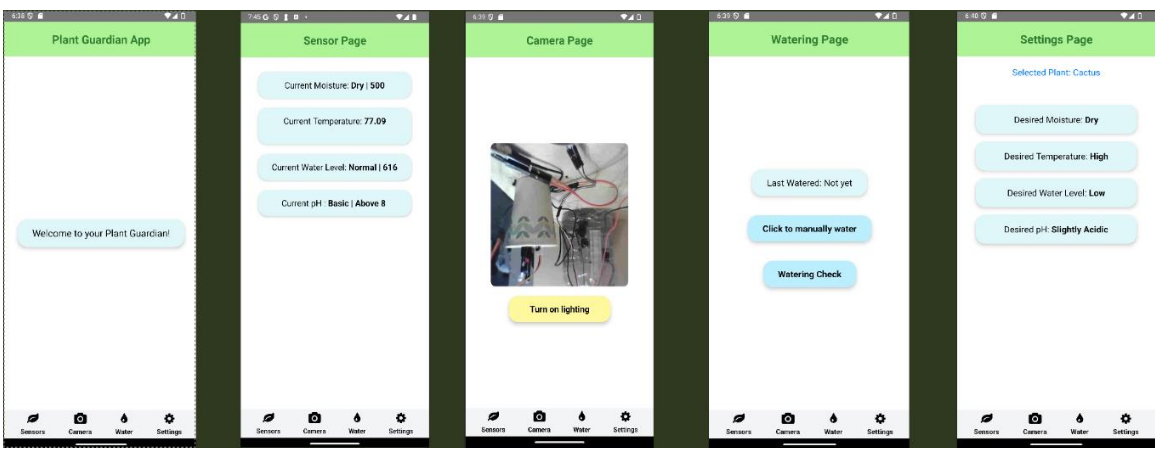



*Figure 7.1: App Interface*

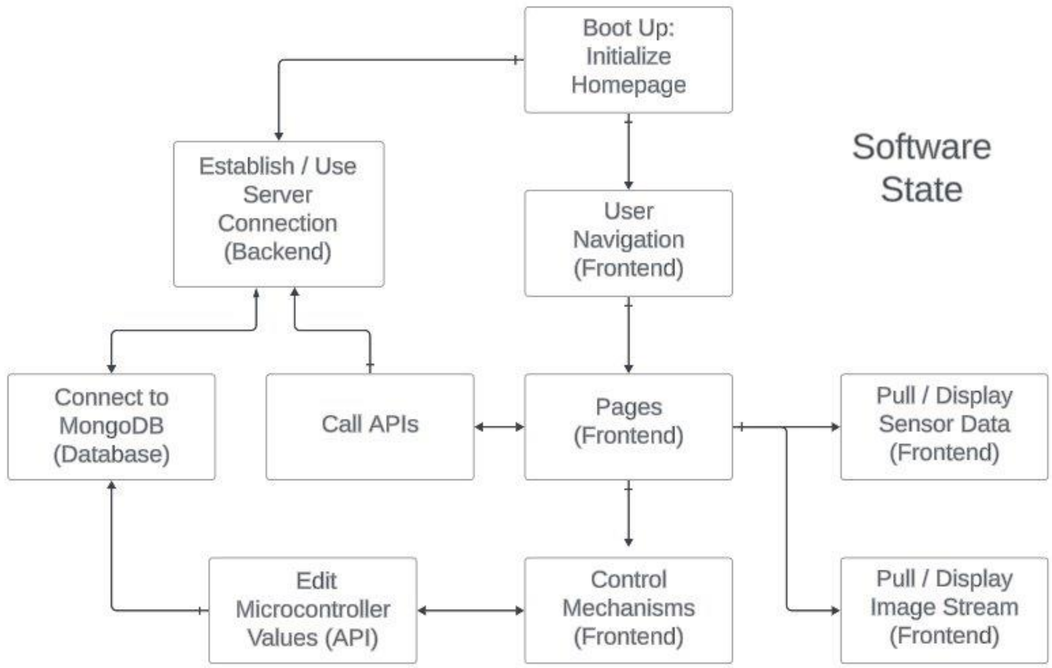



*Figure 7.2: Software/App Flowchart*

## 7.3 App/Sensor Integration

The Plant Guardian app had to communicate with the hardware being used. Each sensor played a critical role in maintaining the health of the plant by providing real-time data that informed the system's decisions. This information was made available to users to help them make better decisions about their plant's health. Below were the specific tasks performed by each sensor:

**Temperature Sensor**

The temperature sensor monitored the ambient temperature around the plant. If the temperature deviated from the optimal range for the plant species, the app notified the user to take action, such as moving the plant to a cooler or warmer location.

**Water Level Sensor**

The water level sensor ensured that the water reservoir had sufficient water. If the water level dropped below a certain point, the app alerted the user to refill the reservoir to prevent the system from running dry.

**Moisture Sensor**

The moisture sensor measured the soil moisture levels. When the soil became too dry, the system activated the watering mechanism to replenish moisture and maintain optimal growing conditions.

**pH Sensor**

The pH sensor tracked the soil’s pH level to ensure it stayed within a range suitable for the plant. If the pH level became too acidic or alkaline, the app notified the user to adjust the soil conditions appropriately.

## 7.3.1 Database for Sensors

We chose MongoDB as our database of choice for conducting sensor integration. MongoDB is part of the MERN stack that is being used.

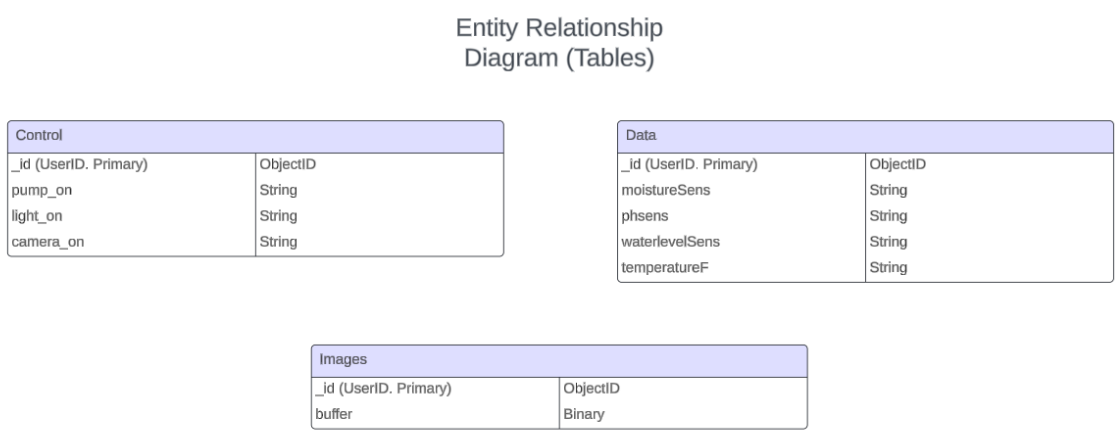



*Figure 7.3: MongoDB ERD*

## 7.4 Stack Considerations

When developing the Plant Guardian app, various technology stacks were evaluated to ensure the final product met all functional and non-functional requirements. These considerations involved selecting the appropriate stack, which significantly influenced the app's performance, scalability, maintainability, and overall user experience. The choice of technology stack was driven by several factors, including the need for real-time data processing from multiple sensors, seamless user interface interactions, robust backend infrastructure, and scalability to accommodate a growing user base. Below is an overview of the potential technology stacks that were considered, each offering unique advantages and challenges, to identify the most suitable stack for the project.

**Potential Technology Stacks**

The choice of technology stack was critical as it determined the efficiency and effectiveness of the development process and the end product. Several popular technology stacks were considered, each with its strengths and weaknesses, to find the best fit for the automatic plant watering device checker app. The main contenders included the LAMP stack, MEAN stack, Django stack, and MERN stack. Each stack was evaluated based on its components, advantages, and challenges, with a particular focus on how well it supported real-time data handling, scalability, and ease of development and maintenance.

In the following sections, each technology stack is explored in detail, examining its components and how it could be applied to the project. This comprehensive analysis ensured that an informed decision was made, aligning the chosen stack with the project's goals and requirements.

**LAMP Stack (Linux, Apache, MySQL, PHP)**

The LAMP stack consisted of Linux as the operating system, Apache as the web server, MySQL as the database, and PHP as the server-side scripting language. Frontend development involved using HTML, CSS, and JavaScript. This stack offered proven stability and had extensive documentation and a large support community. It was ideal for traditional web applications that did not require real-time updates. However, it faced challenges in scalability compared to modern stacks and was less efficient for developing dynamic single-page applications.

**MEAN Stack (MongoDB, Express.js, Angular, Node.js)**

The MEAN stack utilized Angular for creating dynamic SPAs on the frontend, Node.js with Express.js for the backend, and MongoDB for the database. This stack allowed for full-stack JavaScript development, enabling efficient sharing of code between the client and server. Angular provided a robust framework for building SPAs, making the application highly interactive and responsive. The main challenge with the MEAN stack was Angular’s complexity and steep learning curve, which could pose difficulties for our team.

**Django Stack (Python, Django, PostgreSQL)**

The Django stack involved using Python and the Django framework for backend development, PostgreSQL as the database, and typically HTML, CSS, and JavaScript for the frontend. Modern frontend frameworks like React or Vue could also be integrated for a more dynamic interface. Django offered a high level of abstraction and promoted rapid development, with Python being known for its simplicity and readability. The challenges included the need to learn Python and Django if the development team was not already familiar and the additional setup required to integrate with modern frontend frameworks.

**MERN Stack (MongoDB, Express.js, React, Node.js)**

The MERN stack consisted of React for the frontend, Node.js with Express.js for the backend, and MongoDB for the database. This stack enabled full-stack JavaScript development, allowing code to be shared efficiently between the client and server. React facilitated the creation of dynamic and responsive user interfaces, while Node.js and Express.js handled the server logic and API endpoints. MongoDB provided a flexible, scalable solution for data storage. The MERN stack was particularly suitable for applications requiring real-time updates and a seamless user experience.

In particular, the MERN stack's ability to handle asynchronous operations and real-time data updates was crucial for the automatic plant watering device checker app. The application needed to process data from multiple sensors continuously and trigger actions based on specific thresholds. React provided a smooth and interactive user interface where users could monitor their plants' status in real-time. Node.js, with its non-blocking I/O operations, managed multiple sensor inputs efficiently, ensuring that the backend processed data quickly and accurately. MongoDB's flexible schema allowed storage of various types of sensor data without predefined structures, making it easier to adapt to changes in sensor types or data formats.

The MERN stack was chosen for this project due to its cohesive use of JavaScript across the entire stack, simplifying development and maintenance. React's component-based architecture allowed for efficient UI development and reusability. Node.js and Express.js provided a powerful backend environment capable of handling concurrent connections with high performance. MongoDB offered a NoSQL database solution that was scalable and flexible, fitting well with the dynamic nature of the application. Additionally, the extensive community support and comprehensive documentation available for the MERN stack contributed to its appropriateness for this project. By leveraging the MERN stack, a robust, scalable, and efficient solution could be ensured to meet the project's goals.

In conclusion, after evaluating various technology stacks, the MERN stack emerged as the most suitable choice for the automatic plant watering device checker app. Its consistent use of JavaScript, robust handling of real-time data, flexibility, and scalability aligned perfectly with the project requirements, ensuring a successful and efficient implementation.

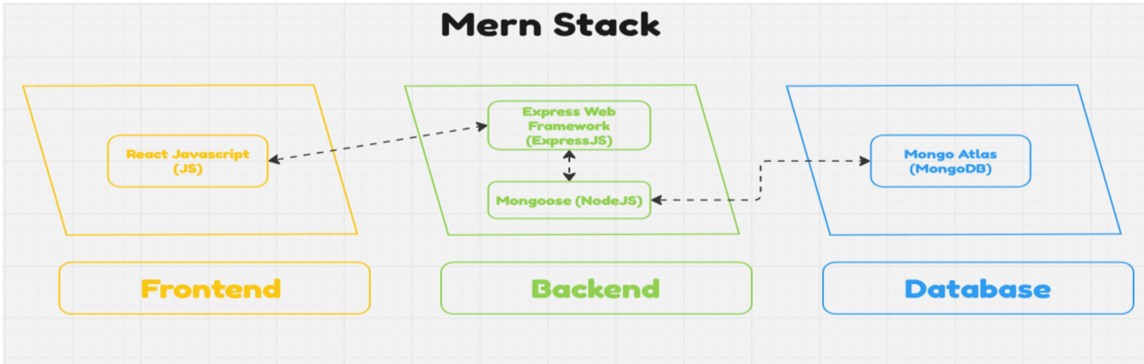


*Figure 7.4: Mern Stack Breakdown*

## 7.5 Platform Considerations

The next consideration was the various mediums that users could potentially use. Each option was carefully evaluated regarding its accessibility, impact, and overall value to the user experience. As a system designed with user notifications in mind, the decision was made to operate with phone applications rather than a web application. A person generally always carries their phone and has access to important notifications from their everyday applications. The app notifies users about their plant's health based on sensor information, including updates if the plant needed watering or changes in its environment. This was determined to be a better alternative to using a web app with an inbox of updates, as users might feel negligent in taking extra time to access a web application, akin to sifting through an email account.

### Platform Choices

When developing the Plant Guardian app, several platform options were considered to ensure the best user experience, accessibility, and functionality. The main platforms evaluated were web applications, Android applications, iOS applications, and hybrid applications. Each platform offered its own set of advantages and challenges, and the decision on which to use depended on factors such as user accessibility, development complexity, and the ability to provide and access timely notifications and real-time updates.

### Web App

A web application was a potential option for hosting the implementation of this project. It would have been built using Visual Studio Code with a React framework for the frontend, Node.js for server-side scripting, Express.js for the backend framework, and MongoDB as the database. This combination was part of the MERN stack, enabling efficient and scalable development. React offered a dynamic and responsive user interface, while Node.js and Express.js handled the server logic and API endpoints, and MongoDB managed data storage and retrieval.

While a web app would have functioned similarly to a phone app, it required a browser for access rather than being a dedicated application. Users could view updates about their plants with timestamps via the web. However, it was determined that users typically did not consistently access the web for such tasks. As a result, the team chose not to create a web app, believing it would not be suitable for receiving and displaying daily notifications.

Instead, a React Native application was chosen for its ability to provide a native app experience across both Android and iOS platforms. This decision ensured timely notifications, seamless user interactions, and real-time updates, enhancing the overall user experience.

### Android App

An Android app was chosen for this implementation. Developing the Android app involved using React Native for cross-platform mobile development, allowing code to be reused for both Android and iOS platforms. For backend services, Node.js with Express.js was used to manage server-side operations, and MongoDB handled data storage. Visual Studio Code served as the development environment, while the final app was tested and built using Android Studio and a physical Android phone.

This approach integrated the MERN stack's benefits—such as efficient server management and scalable database solutions—into the Android app development process. A burner Android phone was planned for purchase to simplify the development and testing process, avoiding the complexities associated with the iOS App Store publisher process.

### iOS App

An iOS app was considered for the implementation of this project, given the prevalence of iPhones and other Apple devices among users. Developing an iOS application would have involved using React Native for the frontend, which is compatible with iOS development. The backend would have been powered by Node.js and Express.js, with MongoDB as the database, leveraging the full MERN stack. Development would have been conducted using Visual Studio Code, and the final app would have been tested and built using Xcode.

This setup would have allowed the project to utilize the robust server-side capabilities of Node.js and the flexibility of React Native to create a seamless user experience. However, logistical challenges and the complexity of the App Store publisher process led to the decision to favor the Android app development route. Additionally, the team lacked access to a personal computer capable of running Xcode, which is unavailable on Windows and Linux devices, further influencing the decision.

## 7.6 Software Considerations

It was essential to carefully evaluate and select each individual technology that made up the stack. Successfully integrating these technologies allowed for the creation of a functioning app. The goal was to select the most accessible and user-friendly technologies that best supported team collaboration. These technologies related to the frontend, backend, database, or other aspects of software design.

### Frontend

### Flutter

Flutter is an open-source UI software development kit created by Google. It allows developers to build cross-platform applications for iOS, Android, web, and desktop from a single codebase. Flutter's primary advantage is its ability to create apps for multiple platforms with a single codebase, saving time and resources. It offers a rich set of pre-designed widgets that are customizable and facilitate building visually appealing UIs. The hot reload feature allows developers to see changes in real-time without restarting the entire app, speeding up the development process. Flutter apps are compiled directly to native ARM code, resulting in high-performance applications.

Although Flutter was considered by our group due to its capabilities, it was not chosen for this project. Its features, such as rich UI components and cross-platform compatibility, would have helped in creating an intuitive and engaging user interface, including dashboards, sensor data views, and control screens. However, the team ultimately opted for a different stack that better aligned with the specific requirements of our project.

### React

React is a JavaScript library for building user interfaces, maintained by Facebook and a community of developers. It is particularly known for its efficiency in creating dynamic and responsive web applications. React's component-based architecture allows developers to build encapsulated components that manage their own state, making the code more reusable and easier to maintain.

React uses a virtual DOM to improve performance by minimizing direct manipulation of the actual DOM. Its extensive community and ecosystem of libraries and tools were leveraged by our project to enhance the development process.

For our project, React was used to create a responsive and interactive user interface for the plant watering system. Its dynamic features allowed us to build intuitive dashboards where users could monitor sensor data, view updates, and control various aspects of their plant care system. By utilizing React, the group ensured a seamless and user-friendly experience tailored to the needs of the application.

### Angular

Angular is a platform and framework for building single-page client applications using HTML and TypeScript. It is developed and maintained by Google. Angular provides a robust framework for building dynamic SPAs, making it suitable for complex applications with extensive functionality. It offers a rich set of features, including data binding, dependency injection, and comprehensive testing tools. Its modular structure makes it scalable and maintainable, though it has a steep learning curve for beginners.

While Angular is part of the MEAN stack and was considered for our project, it was not chosen due to its complexity and the group’s preference for a simpler, more accessible framework. React was selected instead to align with our project’s requirements for creating a dynamic, user-friendly interface while maintaining ease of development. Angular remains a strong choice for enterprise-level applications requiring extensive functionality and scalability.

### Backend

### Node.js

Node.js is an open-source, cross-platform JavaScript runtime environment that executes JavaScript code outside a web browser. It is designed for building scalable network applications. Node.js uses a non-blocking, event-driven architecture, which makes it lightweight and efficient—perfect for data-intensive real-time applications. Using JavaScript for both the frontend and backend streamlines the development process, making it easier for full-stack developers to collaborate effectively on the project. Node.js also has a rich ecosystem of libraries and modules available through npm (Node Package Manager), allowing developers to easily extend their applications' functionality.

In our project, Node.js was used to build the server-side application that handled API requests, managed communication between the app and the sensors, and processed data. It ensured real-time data synchronization and efficiently handled concurrent connections, allowing the system to provide timely updates and a seamless user experience.

### Express.js

Express.js is a minimal and flexible Node.js web application framework that provides a robust set of features for building web applications. It offers a middleware system that allows developers to handle HTTP requests, responses, and routing efficiently. Its minimalistic design makes it easy to learn and use, yet it remains flexible enough to support complex applications. With widespread usage and extensive documentation, Express.js benefits from a large community that contributes tutorials, plugins, and modules, making it a versatile tool for backend development.

In our project, Express.js was used to set up the server and API endpoints. It handled routing, managed HTTP requests, and served as the backbone of the backend infrastructure. This ensured smooth communication between the frontend application and the sensors, allowing real-time data processing and efficient integration of the system components.

### PHP

PHP is a popular general-purpose scripting language especially suited for web development. It is fast, flexible, and pragmatic, powering a wide range of websites, from small blogs to some of the most popular sites in the world. PHP is a key component of the LAMP stack and is used primarily for server-side scripting. It is known for its ease of use and wide adoption, making it an accessible choice for developers familiar with web development.

PHP can be embedded into HTML, which simplifies development for those building traditional server-rendered web applications. However, PHP is less efficient for handling real-time data processing compared to more modern back-end technologies like Node.js, which are better suited for applications requiring high performance and scalability in real-time scenarios.

### Django (Python)

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It handles many of the complexities of web development, allowing developers to focus on building their app without needing to reinvent the wheel. Django is part of the Django stack and is well-regarded for its high-level abstractions, built-in admin interface, and robust security features.

Django is particularly suitable for building robust and scalable web applications, offering tools that streamline development and promote best practices. However, it may present a learning curve for developers who are not familiar with Python or its framework-specific conventions.

### Database

### MongoDB

MongoDB is a NoSQL database known for its flexibility, scalability, and performance. It stores data in JSON-like documents, making it easy to use for modern applications. MongoDB’s document-oriented storage is more adaptable and scalable than traditional relational databases, making it ideal for handling diverse and evolving data structures. It efficiently manages large volumes of data and scales horizontally by distributing data across multiple servers.

MongoDB supports a rich, expressive query language that enables developers to filter, sort, and aggregate data with ease. Its free tier is particularly suitable for student projects and small-scale applications, offering a cost-effective solution for development needs. MongoDB was used to store user data, sensor readings, and control logs in our project, providing a flexible and scalable solution for managing the diverse data generated by the plant watering system.

### MySQL

MySQL is an open-source relational database management system and a central component of the LAMP stack. It is widely recognized for its reliability, performance, and ease of use. MySQL uses structured query language (SQL) for database access and management, a language that is industry-standard and familiar to many developers.

It is well-suited for applications requiring structured data storage and complex queries. MySQL is also highly scalable, capable of handling large volumes of transactions, which makes it a dependable choice for web applications needing a robust relational database.

### PostgreSQL

PostgreSQL is a powerful, open-source object-relational database system with a strong reputation for reliability, feature robustness, and performance. It supports both SQL for relational queries and JSON for non-relational queries, providing versatility in handling diverse data needs.

PostgreSQL is a key component of the Django stack and is known for its advanced features, including support for complex queries, foreign keys, triggers, and updatable views. It is particularly suitable for applications requiring intricate data relationships and operations. Its extensive support for data integrity and concurrency control makes it an excellent choice for high-availability and mission-critical applications.

## Development Tools

### Visual Studio Code

Visual Studio Code (VSCode) is a free, open-source code editor developed by Microsoft, renowned for its versatility, powerful features, and extensive plugin ecosystem. It offers a rich marketplace of extensions that enhance its functionality, including plugins for frameworks and tools such as React, Node.js, and more.

VSCode includes an integrated terminal, making it convenient to run command-line tools and scripts directly within the editor. It also features powerful debugging tools, supporting breakpoints, call stacks, and an interactive console, which streamline the debugging process. Its seamless integration with Git enables easy version control and collaboration without leaving the editor.

Visual Studio Code was the primary code editor used in our project for writing, debugging, and managing the codebase for both the frontend and backend components. Its extensive library of extensions enhanced productivity, while its debugging and Git integration simplified development workflows.

**GitHub**

GitHub is a web-based platform for version control and collaboration that uses Git. It offers a wide range of features for managing and sharing code, tracking changes, and collaborating with other developers.

GitHub provides robust version control capabilities, allowing developers to track changes, revert to previous versions, and manage multiple branches of code. Its collaboration tools, such as pull requests, code reviews, and issue tracking, streamline teamwork and improve communication. Additionally, GitHub integrates with continuous integration and continuous deployment (CI/CD) tools, automating build, test, and deployment processes for smoother development workflows.

In our project, GitHub was used for version control and collaboration. It enabled the development team to work efficiently, track changes, and maintain a stable and up-to-date codebase throughout the development process.

**Design**

**Figma**

Figma is a collaborative interface design tool widely used for web and app development. It allows multiple designers to work together in real-time, making it an ideal tool for collaborative projects.

Figma's real-time collaboration feature enables team members to work on a design simultaneously, improving teamwork and accelerating the design process. As a cloud-based tool, it ensures that everyone has access to the latest version of the design files from anywhere. Figma also includes robust prototyping tools, allowing designers to create interactive prototypes and share them with stakeholders for feedback. Its support for design system management helps teams maintain consistency across designs by using shared styles and components.

In our project, Figma was used to design the user interface of the app, enabling the team to create and refine design concepts collaboratively. The prototyping features allowed us to test the user flow and gather feedback early in the development process, ensuring that the app's design was both user-friendly and visually appealing.

## 7.7 Detailed Use Case Scenarios

Understanding how different users will interact with the Plant Guardian app helps in designing a user-friendly and effective solution. Here, we outline several detailed use case scenarios to illustrate the app's practical applications.

### Scenario 1: The Busy Professional

A busy professional enjoys having plants in their apartment but often forgets to water them due to a hectic schedule. With the Plant Guardian app, they set up their plants with the necessary sensors and configure the app to automatically water the plants based on moisture levels. The app sends notifications when additional care is needed, such as adjusting the pH level or temperature. They can monitor their plants' health in real-time through the app, ensuring the plants remain healthy and vibrant without requiring constant attention.

### Scenario 2: The Commercial Greenhouse Manager

A commercial greenhouse manager oversees a variety of plants requiring precise water and nutrient management. Using the Plant Guardian app, they install sensors throughout the greenhouse to monitor moisture, temperature, pH levels, and water levels. The app automates the watering process, ensuring each plant receives the optimal amount of water. They receive detailed reports on greenhouse conditions, enabling data-driven decisions to enhance productivity and resource efficiency. The app’s automation features reduce labor costs and improve overall plant health.

### Scenario 3: The Urban Community Garden Organizer

An urban community garden organizer relies on volunteers for maintenance. The Plant Guardian app helps coordinate watering schedules and monitor plant health remotely. They set up the garden with sensors and configure the app to notify volunteers when specific tasks need attention. Real-time data and notifications ensure the garden is well-maintained even when volunteers have varying availability. This fosters community engagement and ensures the garden remains a thriving green space in the urban environment.

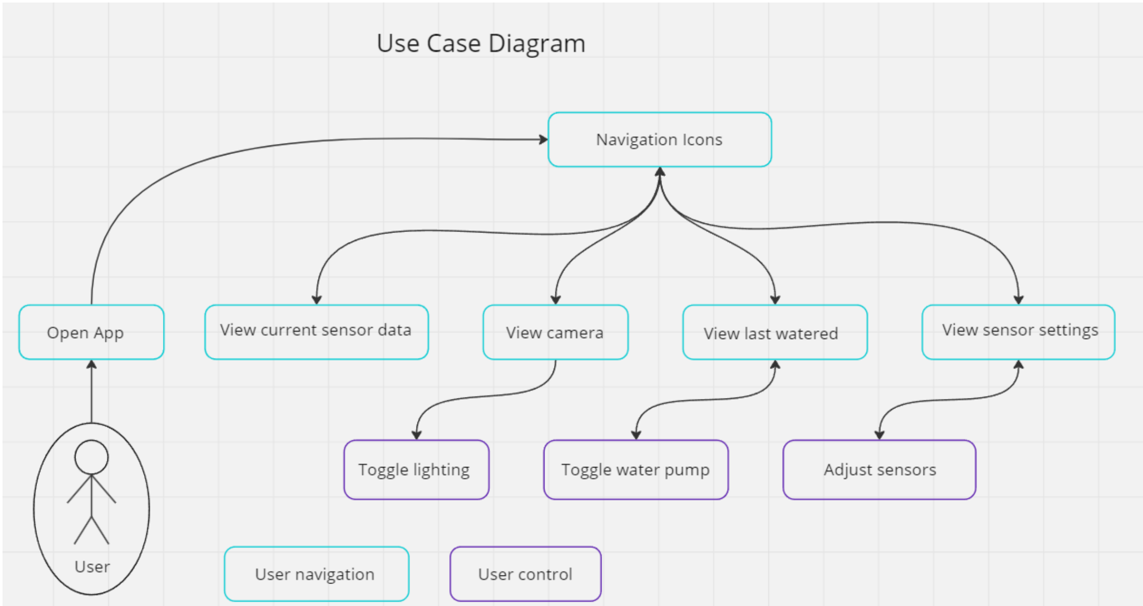



*Figure 7.5: Use Case Diagram*

## 7.8 App Impact

The automatic plant watering system we developed has the potential to make significant impacts across various areas, including individual households, commercial agriculture, and the overall environment. In this section, we explored the possible benefits.

### Individual Households

This system directly offers convenience, saving time and effort for those who need it. Users can automate the care of their plants, ensuring that they are watered appropriately even when they are busy or away from home. This reduces the chances of overwatering or underwatering, both of which can be detrimental to plant health.

Plant health will be improved, as even well-taken-care-of plants can receive optimal treatment through the system's sensors and data. This will lead to lush and vibrant plants that provide an aesthetically healthy environment. Healthy plants can enhance the ambiance of a home, contributing to a pleasant living space. Additionally, this system would foster engagement between users and their plants. Users would learn about their plant's needs as well as gain insight into raising other similar plants. This educational aspect can increase a user's confidence in plant care, encouraging them to expand their home gardens.

Furthermore, the system can serve as an educational tool for families, teaching children about the importance of plant care, water conservation, and environmental stewardship. This interaction can instill a sense of responsibility and appreciation for nature in younger generations.

**Commercial Agriculture**

Although commercial agriculture is already very systematic, this automatic plant watering system offers an opportunity to gain resource efficiency. Limited resources such as water can be efficiently managed and distributed rather than routinely consumed. Often, various households and greenhouses may use sprinkler systems that over-apply water more than necessary. This system helps in reducing water waste and ensuring that plants receive the right amount of water at the right time.

In commercial settings, optimizing water usage can lead to significant cost savings and increased sustainability. The system’s sensors provide precise data that allows for targeted watering, reducing runoff and evaporation losses. This precision can lead to higher crop yields and better quality produce, as plants receive consistent and optimal care.

Moreover, the system can be integrated with other agricultural technologies such as soil nutrient monitors and climate control systems. This integration creates a comprehensive smart farming solution that enhances overall productivity and resource management. By adopting such technologies, commercial farms can move towards more sustainable practices, reducing their environmental footprint and contributing to global food security.

**Overall Environment**

A lush environment is proven to have a positive impact on society. Rather than focusing on just individual and commercial viewpoints, there is a benefit in having urban greenery. The creation and maintenance of community gardens play a role in improving air quality and reducing urban heat islands. Many cities suffer from structures that retain heat compared to rural areas that have greenery. Communities can also share a sense of unity in the symbolic structures that are created. This sense of shared responsibility in the local environment can improve the mental health and well-being of the area.

The environmental benefits extend beyond individual gardens and farms. By promoting water conservation, the system contributes to the sustainable use of natural resources. Water is a precious resource, and its efficient use is crucial in the face of growing global water scarcity. Reducing water waste through automated watering systems can significantly lessen the strain on local water supplies.

Additionally, the system supports the growth of urban green spaces, which play a critical role in mitigating climate change. Plants absorb carbon dioxide and release oxygen, improving air quality and reducing the urban heat island effect.

Green spaces also provide habitat for urban wildlife, promoting biodiversity in city environments.

This system has the potential to make significant multifaceted impacts. By enhancing plant health, conserving resources, and promoting sustainable practices, our system can benefit individuals, commercial agriculture, and the broader environment. Additionally, its contributions to urban green spaces, community engagement, and mental well-being highlight its value as a tool for fostering a healthier society. Through careful design and thoughtful implementation, we strive to create a solution that addresses the needs of users while promoting environmental responsibility and social well-being.

The automatic plant watering system is a step towards more sustainable living, encouraging individuals and communities to adopt practices that support environmental conservation and resource efficiency. By making plant care more accessible and efficient, we hope to inspire a broader appreciation for nature and foster a more sustainable relationship with our environment. This system is not just a tool for plant care, but a catalyst for positive environmental change.

# 8 System Fabrication/Prototype Construction
## PCB

### Version 1

Our first version of the PCB was designed during senior design 1, We understood that due to the complexity of integrating two MCU's and a camera which has a lot of data signals that our PCB could prove to be a major challenge. So we decided to turn our schematic into a first draft of our PCB. For our PCB we used EasyEDA. The main reason for using EasyEDA was due to their easily accessible library of components which allowed for trying out different components during the designing phase. Starting with the overall layout of the PCB. We decided to put our headers for our sensors and power in one location for ease of wiring into the plant watering system case. Next, due to the ATMEGA328's large number of connections to other parts of the circuit, we put it in the middle in order to reduce the number of VIA's we would need. Similarly, we put the ESP32-S and it's PSRAM and camera port on the right side of the board due to their high density of connections between each other. We also included the 3.3v, 1.2v, and 2.8v power regulators on the right side because they are used for the ESP32-S and camera port. We then located the level shifter in between the ESP32-S and the ATMEGA328. Finally we put our relays on the far left side of the board. We did this due to concern for high voltage interfering with the camera traces. While the PCB design isn't required for Senior Design 1, we anticipated challenges given the complexity of integrating two MCUs and a high-data camera. Understanding this, we opted to translate our schematic into an initial PCB draft ahead of schedule. For our PCB software, utilizing EasyEDA was an obvious choice due to its extensive component library, which facilitated experimentation with various components during the design phase. Our approach to the PCB layout began with strategic decisions. Headers for sensors and power were consolidated in one area to simplify integration with the plant watering system enclosure. Recognizing the ATMEGA328's extensive interconnections within the circuit, we positioned it centrally to minimize the need for vias. Similarly, the ESP32-S, alongside its PSRAM and camera interface, was placed on the right side of the board due to their dense connectivity with each other. To support the ESP32-S and camera functions effectively, we incorporated 3.3V, 1.2V, and 2.8V power regulators on the same side. The placement of these components was intentional, aligning them with their respective power demands and minimizing trace lengths. A level shifter was strategically positioned between the ESP32-S and ATMEGA328 to manage voltage compatibility between these critical components. Furthermore, the placement of relays on the far left side of the board was driven by concerns regarding potential interference from high-voltage components, particularly safeguarding against disruptions to the sensitive camera traces.

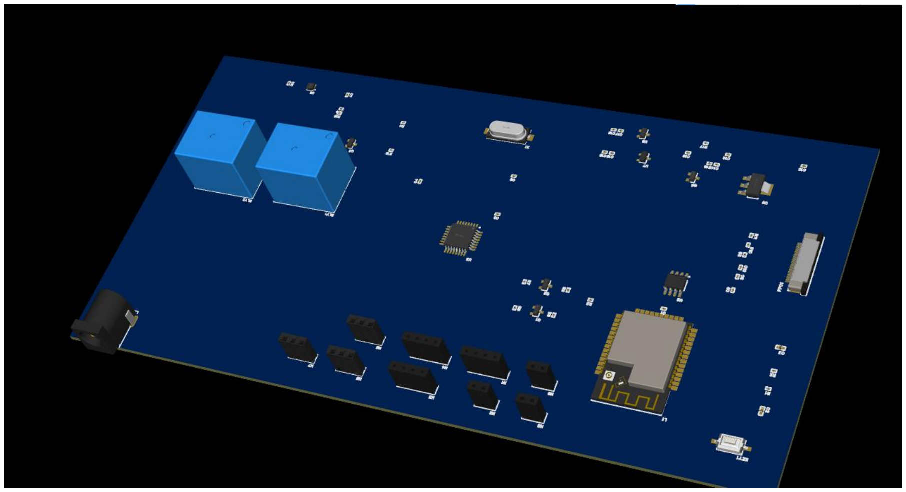


*Figure 8.1: The 3d render of the PCB*

Because of the extensive pin requirements for the camera port, we ultimately opted for a 4-layer PCB configuration. The top layer primarily accommodates power and ground traces, ensuring stable and reliable electrical distribution across the board. In contrast, the two inner layers, and to a lesser extent the bottom layer, are dedicated predominantly to data lines, facilitating efficient communication between components. However, the decision to utilize additional layers does come with its drawbacks, notably an increase in overall PCB manufacturing costs. This increase is primarily due to the complexity involved in manufacturing and aligning multiple layers during production. Despite this downside, the benefits of routing flexibility outweighed the cost considerations in our design process.

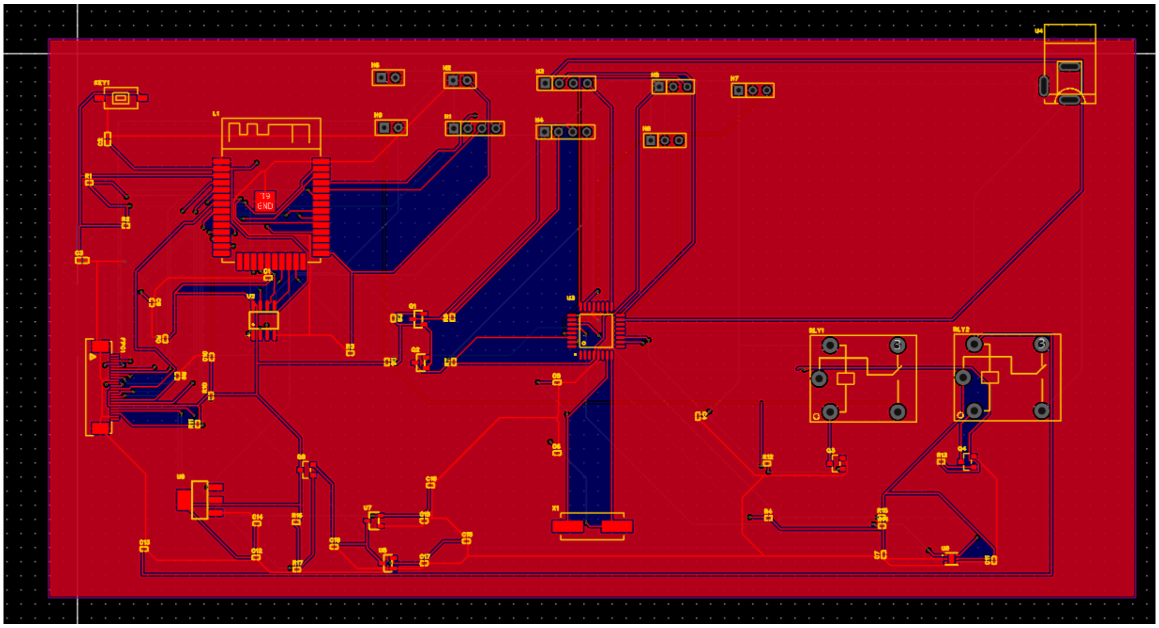


*Figure 8.2: The top and bottom traces*

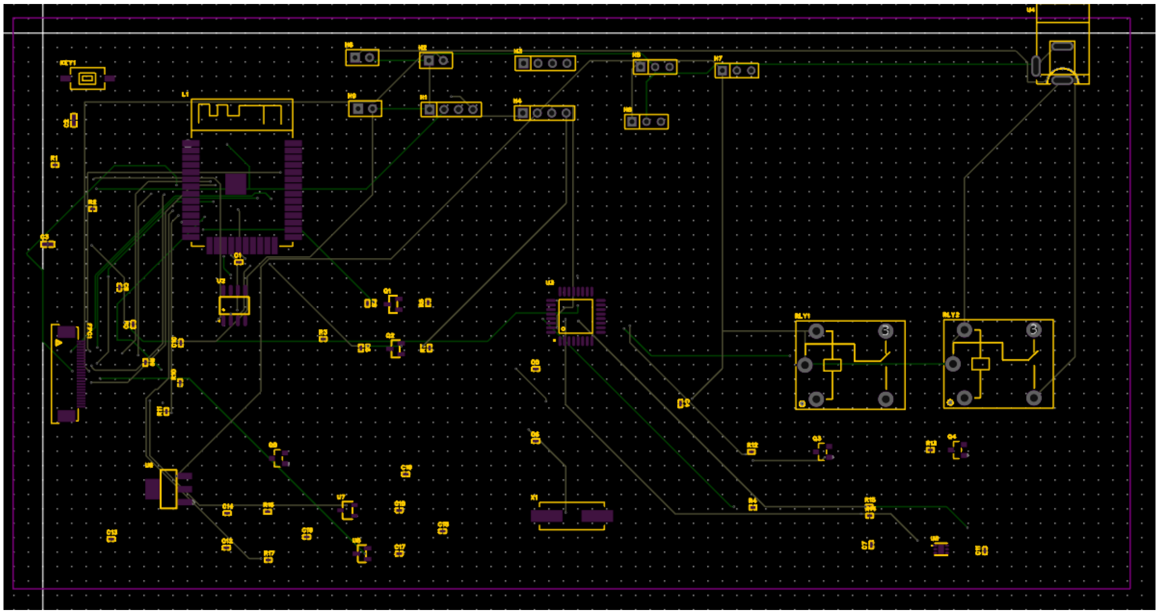


*Figure 8.3: The two inner PCB layer traces*

## Version 2

After speaking with Dr.Weeks and going through our first iteration of PCB. We found due to limitations in the tools we have we needed to simplify the PCB. The biggest issue is we needed to solder a ribbon cable connector for a camera that was too small for our oven to handle. So we moved the ESP32 system onto a dev board that includes a camera connector. We also found in our initial design a few issues with the circuit. First the ESP32 was mounted the wrong way so that the camera was on the wrong side. This meant that it needed to be flipped to the other side. There was also an issue with a ground pin being connected to the reset pin causing infinite looping. This was also resolved. For the sensors the temperature sensor needed to have a filter circuit to reduce interference as well as a bypass capacitor to ensure consistent readings. Then for the light circuit was designed for 5v but we had to switch to a 12v light so we changed the relay to include the light power supply. Finally we made the overall board 6x6 inches from 8x8 inches as well as added mounting holes.

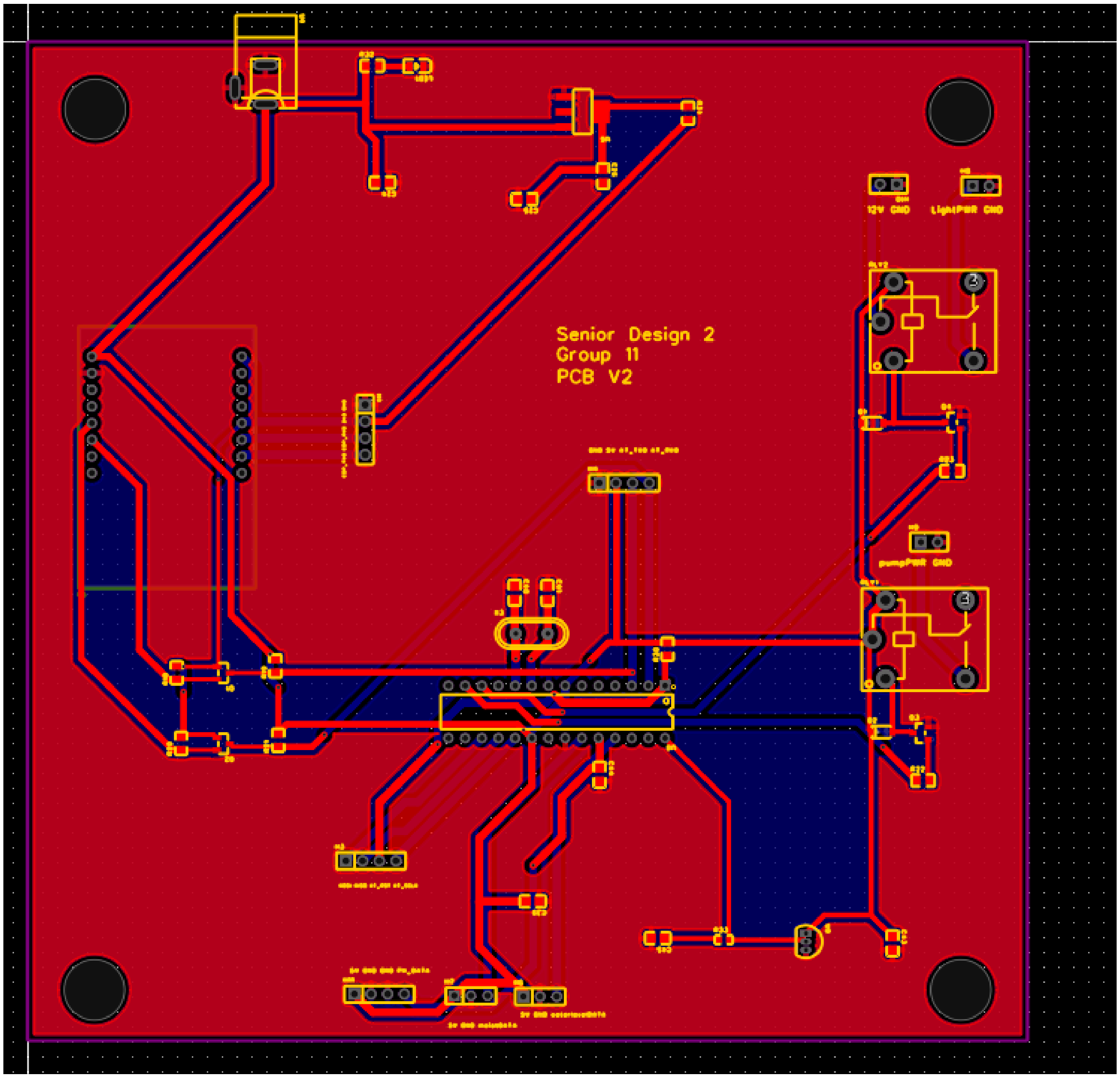



*Figure 8.3: Version 2 of our PCB*

## Exterior Fabrication

Given our limited expertise in mechanical engineering, our approach to the structural design of our system emphasizes simplicity, practicality, and resourcefulness. To achieve this, we devised a straightforward and cost-effective plan that effectively supports our system's requirements.

For the reservoir, we repurposed a 3-liter container, which we modified by cutting a hole to accommodate the water level sensor. We also installed a pump inside the container to enable automated irrigation. For the plant setup, we used a pot already owned by one of our team members, which saved costs and streamlined our design process.

To house the components and provide a stable structure, we constructed a basic stand using a wood shipping crate. We assembled the crate with one side removed, creating a functional open-box design. This allowed us to mount the PCB on the interior wall of the crate, attach a light fixture to the ceiling, and secure the camera in an optimal position. For the PCB, we took extra precautions by adding foam padding at each corner before taping it to the top of the box. This measure ensured the PCB remained elevated, preventing physical damage and reducing the risk of short circuits.

For the irrigation system, we embedded the soil sensor directly into the plant's soil and connected the pump's output to the soil using a spike and tubing. This configuration allowed for precise delivery of water to the plant while integrating seamlessly with our automated control system.

By repurposing existing resources and opting for readily available materials, we significantly reduced costs and simplified the assembly process. This resource-conscious approach allowed us to focus our efforts on our team's strengths in electronics and software development. As a result, we successfully met the core objectives of our project: efficient plant monitoring and reliable automated irrigation. This practical yet effective design highlights our ability to adapt and innovate within the constraints of our expertise.

# 9. System Testing and Evaluation

System testing and metrics for evaluation are crucial components in the development lifecycle of any complex system, just as critical as having a well-defined design and functional components to begin with. They play a pivotal role in ensuring that each subsystem operates correctly on its own and integrates seamlessly into the larger system framework. By conducting thorough testing and measurement of performance metrics, we not only verify that each individual component like the ESP32-S, ATMEGA328P microcontroller, Soil Moisture Sensor, Water Pump, Grow Lamp, pH Sensor, and Temperature Sensor functions as intended but also guarantee their cohesive operation within the entire system architecture.

The primary objective of system testing is twofold: firstly, to confirm the proper functioning of each subsystem according to its specifications and requirements, and secondly, to identify and rectify any potential issues or discrepancies early in the development cycle. This proactive approach significantly reduces the likelihood of encountering critical flaws or failures during later stages of deployment or operation.

Among the various components listed, the ESP32-S holds particular significance as it serves as the central interface between our system's sensors, actuators, and the internet. Its proper functionality ensures reliable communication and data transfer, enabling real-time monitoring and control capabilities essential for the system's overall performance and efficiency.

## 9.1 MCU

Our microcontroller unit (MCU) setup has evolved from our original plan due to a discovery during testing of the ESP32-S. We realized that when utilizing WiFi, the ESP32-S restricts the number of available IO ports more than we initially anticipated. To address this limitation, we opted to integrate an ATMEGA328 microcontroller into our system architecture. This decision allows the ATMEGA328 to manage the majority of our sensors and relays, while the ESP32-S focuses on handling WiFi connectivity and higher-level tasks. Communication between these microcontrollers is achieved through UART, ensuring efficient data exchange and coordination.

Given the challenge of prototyping with SMD (Surface-Mount Device) chips, we selected the ESP-CAM development board for testing purposes. This board features an ESP32-S, 4MB of PSRAM, and a dedicated camera slot, providing a robust platform for integrating and testing our components. For the ATMEGA328, we employed an Arduino Uno, which also supports USB flashing for convenient programming.

To verify the functionality of the ESP32-S, we uploaded a C++ program via the Arduino IDE to the ESP-CAM. This program enabled serial communication, allowing us to confirm the ESP32-S responsiveness and validate our UART implementation. Subsequently, we replicated this testing process on the Arduino Uno, ensuring consistent success across both platforms using the same code and development environment.

By adopting this dual-microcontroller approach and leveraging suitable development boards, we ensure comprehensive testing and seamless integration of our MCU components. This method not only addresses initial design challenges but also establishes a solid foundation for further development and optimization of our agricultural automation system.

## 9.2 Camera and Wifi

Testing the OV2640 camera module necessitates evaluating not only its functionality but also the WiFi capabilities of the ESP32-S. This approach allows us to validate both aspects simultaneously by streaming images to a web server hosted on the ESP32-S. Initially, we connected the OV2640 to the camera slot on the ESP-CAM development board. Utilizing the camera demo example code provided by Expressif under the GNU open source license, we verified the integration of the camera and WiFi components.

The demo code effectively initializes the camera and establishes a web server that connects to our specific WiFi network. It provides a user interface through which we can adjust various parameters for the video stream, ensuring flexibility in capturing and transmitting images. Upon successfully connecting to the hosted website, we confirmed that the OV2640 camera was indeed streaming. We further experimented with different video resolutions to optimize image quality while balancing frame rate performance.

This comprehensive testing process not only validated the basic functionality of the OV2640 and ESP32-S WiFi capabilities but also allowed us to fine-tune settings for optimal performance. By systematically adjusting parameters and monitoring results via the web interface, we found what the highest quality to FPS we could achieve. This unfortunately ended up being lower than we expected causing us to increase our buffer on the camera refresh rate specification.

## 9.3 Soil Moisture Sensor

Our moisture sensor, the Adafruit Capacitive Soil Moisture Sensor v1.2, operates with a voltage input ranging from 3.3V to 5V and requires a ground connection. It outputs an analog signal. To integrate it into our system, we connected its VCC to the 5V pin on the Arduino Uno and the ground to the Arduino's ground. The analog output of the sensor was connected to analog pin A0 on the ATMEGA328 microcontroller, which houses the ADC (Analog-to-Digital Converter).

We proceeded by uploading Arduino code that reads the analog value from pin A0 and outputs it to the serial monitor. This allowed us to observe the sensor's readings in real-time. To validate its functionality, we conducted tests with a cup of water, measuring the sensor's output both in the air and submerged in water. This comparison enabled us to confirm that the sensor responded appropriately to changes in moisture levels, demonstrating a measurable difference in output.

Following these tests, we consulted the datasheet provided by DH for the moisture sensor to ensure that our observed results aligned with the expected behavior documented in the specifications. This verification step validated the accuracy of our sensor readings and affirmed its suitability for accurately detecting soil moisture levels in our agricultural automation project.

By systematically testing and verifying the sensor's performance under controlled conditions, we ensured its reliability and effectiveness in providing crucial data for our application. This meticulous approach not only validates our hardware setup but also lays a solid foundation for further integration and development within our automated agricultural system.

## 9.4 Water Pump

Our water pump operates on a 5V DC supply and functions as a straightforward DC motor. Upon receiving the pump, our initial step was to apply a 5V DC current directly to it to verify its operational functionality. Once confirmed, our next objective was to integrate the pump into our control system using the ATMEGA328 microcontroller.

To safely control the 5V pump with the ATMEGA328, we employed a relay. Although the ATMEGA328 can output 5V DC on its digital pins, using a relay is preferred to prevent excessive current draw that could potentially damage the microcontroller. The relay serves as a switch controlled by the ATMEGA328, ensuring safe operation of the pump.

Here's how we set it up: We connected the control signal (data line) of the relay module to digital pin 7 on the Arduino Uno. The power for the relay module (5V) was sourced from the Arduino's 5V rail, and we connected the VCC for the pump to the relay's output as well. Ground connections were made to ensure proper circuit completion.

Programming the Arduino Uno involved setting up serial communication to receive commands. When the Arduino receives a command (e.g., '1' for ON, '2' for OFF) via serial input, it toggles digital pin 7 accordingly. Setting pin 7 to HIGH activates the relay, powering the pump, while setting it to LOW turns off the relay and consequently stops the pump.

After implementing this setup and testing, we successfully demonstrated our ability to control the water pump remotely through commands sent via serial communication. This method ensures both the safety of our microcontroller and the reliable operation of the pump within our agricultural automation system.

## 9.5 Overall Integration

To integrate our various components into our system, we began by connecting the water pump and moisture sensor to the Arduino Uno. Simultaneously, we connected the camera to the ESP-CAM development board. Ensuring proper communication between the Arduino Uno and ESP32-S, we connected the Arduino's RX and TX pins to GPIO 15 and GPIO 14 on the ESP32-S, respectively.

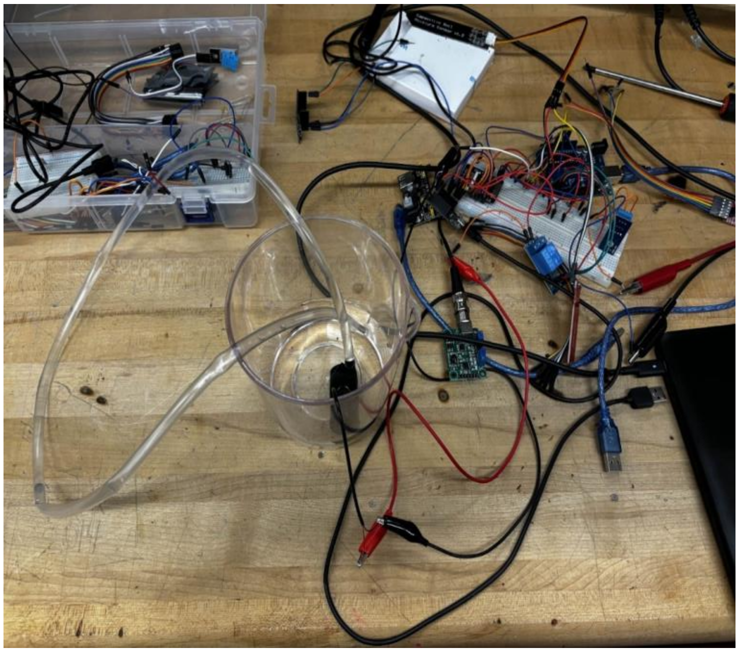


*Figure 9.1: Full breadboard integration*

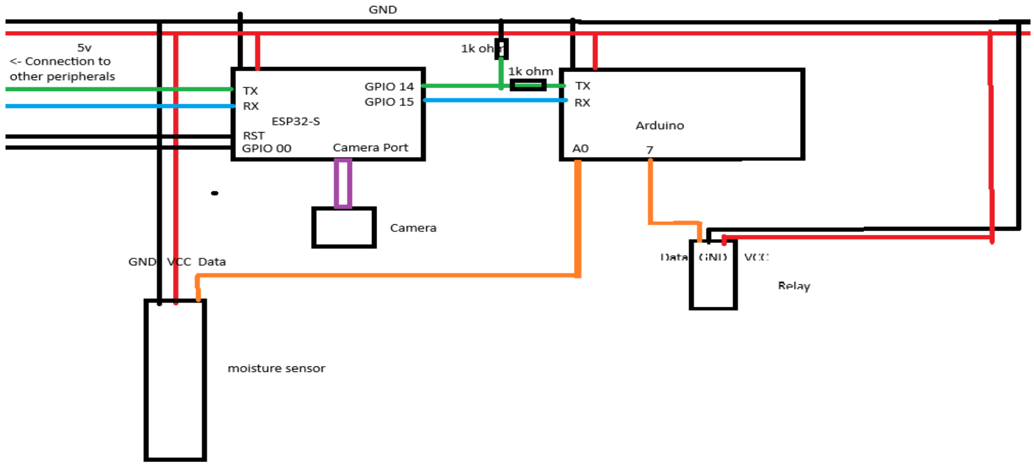



*Figure 9.2: Breadboard full test schematic*

A critical consideration in this integration was the voltage disparity between the ESP32-S (operating at 3.3V logic) and the ATMEGA328 (operating at 5V logic) UART connections. To mitigate potential issues, we implemented a voltage divider to safely interface the TX and RX lines between these microcontrollers. Although effective as a temporary solution, we plan to replace this with a dedicated level shifter circuit on the final PCB design to improve noise reduction and reliability.

During testing, a significant challenge arose with the UART connection on the ESP32-S. While it has two hardware UART chips, one is reserved for WiFi operations, necessitating the use of the SoftwareSerial UART library for communication with the ATMEGA328. Additionally, since the ATMEGA328 has only one UART port used for both programming and communication, we managed this by disconnecting the Arduino Uno's RX and TX pins when not in use. Future iterations of our circuit design will incorporate headers to enable or disable these connections more conveniently.

To ensure seamless data flow between sensors, the ATMEGA328, and ESP32-S, we modified the ESP32-S code to relay any received serial data to the ATMEGA328 and vice versa. We also programmed the ESP32-S to transmit sensor data to our terminal for monitoring. The ATMEGA328 code was enhanced to manage multiple sensors on different ports and transmit data at regular intervals. It also monitored UART communications to toggle the operation of the pump or light based on received commands.

Through these integrative efforts, we successfully established uniform control over all system components and achieved simultaneous data monitoring. This comprehensive approach assures us of the reliability and functionality of our sensor array, reinforcing our confidence as we progress with our agricultural automation project.

## 9.6 Evaluation

Overall, our testing phase proved to be largely successful, although it did uncover several key insights that prompted adjustments to our design. One significant finding was the limitation of the ESP32-S, which cannot fully utilize its ports when its wireless chip is active, leading us to explore the need for an additional MCU to manage these functions effectively. Additionally, our evaluation highlighted the necessity to adjust our specifications regarding image update speed and resolution tolerance to better align with practical capabilities. During camera testing, we identified that the FPS performance achievable by the ESP32-S was somewhat slower than initially anticipated. Despite this, our pump, moisture sensor, and primary MCU functioned reliably both independently and when fully integrated. These successful operational tests have bolstered our confidence in the overall efficacy of our design as we advance into Senior Design 2.

## 9.7 Plan for senior design 2

In Senior Design 2, our team is set to achieve significant milestones as we advance from the foundational work laid out in Senior Design 1. Central to our objectives is the finalization of our PCB schematic, a critical task following comprehensive testing of all hardware components undertaken in the previous phase. With insights gained from Senior Design 1, where prototypes were rigorously assessed and refined, we are poised to translate our refined designs into a fully detailed PCB layout. This stage demands meticulous attention to detail to ensure all connections, components, and functionalities align seamlessly within the printed circuit board.

Anticipating the complexities inherent in PCB fabrication, we recognize that challenges and iterations are inevitable. Addressing potential issues proactively, we are prepared to iterate on our designs as needed, leveraging our collective expertise to optimize performance and reliability. This iterative process is essential for achieving a PCB that not only meets technical specifications but also aligns with our project's overarching goals of functionality and efficiency.

Furthermore, a pivotal aspect of our project advancement in Senior Design 2 involves the full integration of our hardware with our dedicated application. This integration represents the convergence of hardware engineering and software development efforts, ensuring that our technological solution operates seamlessly across these platforms. By synchronizing hardware functionalities with the user-centric features of our app, we aim to deliver an intuitive and responsive user experience. Collaboration between our hardware and software teams will be key as we navigate the intricacies of compatibility testing and user interface optimization, striving towards a cohesive end product.

## 9.8 Post Senior Design 2 Evaluation

In Senior Design 2, as we anticipated, challenges arose during the integration and testing phases, but our earlier efforts in testing proved invaluable in addressing these issues. One significant problem emerged when the temperature sensor was integrated onto the PCB. Unlike in the testbed environment, the sensor exhibited unexpected behavior, which required us to troubleshoot and implement fixes. To resolve the issue, we added a filter and a bypass capacitor to stabilize the sensor's performance.

After making these modifications, we re-tested the PCB thoroughly, applying the same rigorous testing methods we used on the testbed during Senior Design 1. However, instead of solely relying on the sensor's datasheet for validation, we took an additional step by calibrating and verifying our sensor measurements against pre-calibrated commercial sensors. We purchased an external moisture sensor, thermometer, and pH sensor to serve as reference points, ensuring the accuracy and reliability of our readings.

With the completion of both hardware and software integration, we also developed a comprehensive testing framework to verify seamless interaction between the PCB, the server, and the mobile application. This systematic approach allowed us to identify and resolve any integration issues efficiently.

Our most significant challenge during Senior Design 2 was staying within budget. Unexpected complications, such as the need to replace components due to performance or compatibility issues, led to higher-than-expected costs. These changes often forced us to purchase alternative components, some of which were more expensive than originally planned, contributing to the budget overrun. Despite this, the project benefitted from our adaptability and commitment to maintaining quality across all aspects of the design.

# 10. Administration

## 10.1 Work Distribution

Our team consisted of three Electrical Engineering students and one Computer Engineering student, each bringing specialized skills to the project. Keven and Luz, with their strong focus on hardware expertise, were responsible for the meticulous design and implementation of all peripheral devices and power circuits essential to the system's operation. Their role ensured robust and reliable hardware integration, which was crucial for the project's functionality. They worked diligently to select, test, and incorporate sensors, voltage regulators, and other components, ensuring that each part functioned optimally within the overall system. Their attention to detail and knowledge of circuit design helped prevent potential issues related to power supply and signal integrity, thus enhancing the system's reliability and performance.

Tony, leveraging his extensive experience in software development, took charge of crafting the application interface. His proficiency in software design and user interface development guaranteed an intuitive and efficient app that enhanced user interaction and system control. Tony focused on creating an app that was not only user-friendly but also powerful enough to provide real-time monitoring, data visualization, and remote control of the watering system. He ensured that the app seamlessly integrated with the hardware, providing users with an easy-to-navigate platform that simplified plant care and management.

Steven, with a background in embedded programming and hardware systems, undertook the challenging task of bridging the gap between hardware components and the application layer. Steven's responsibilities encompassed managing the microcontroller unit (MCU) and developing firmware that optimized communication and functionality between the hardware elements and the user interface. His work involved writing code that ensured the sensors and actuators responded accurately to commands from the app, as well as implementing protocols for data transmission and device control. Steven's expertise ensured that the system operated smoothly and efficiently, with all components working together harmoniously.

This collaborative division of expertise ensured that our team not only met but exceeded project requirements by effectively leveraging each member's strengths. By combining our skills in hardware design, software development, and embedded systems programming, we aimed to deliver a cohesive and innovative solution that met the project's objectives with precision and efficiency. Our integrated approach allowed us to address challenges comprehensively, ensuring that both the hardware and software components were optimized for the best possible performance. This synergy between team members formed the foundation of our project's success, enabling us to create an automatic plant watering system that was both reliable and user-friendly.

Table 10.1: Sub System Breakdown

| Section | Primary | Secondary |
|---|---|---|
| App Interface | Tony Chau | Steven Keller |
| App Backend | Tony Chau | Steven Keller |
| MCU firmware | Steven Keller | Tony Chau |
| MCU | Steven Keller | Tony Chau |
| Wifi | Steven Keller | Tony Chau |
| Camera | Luz Romero | Steven Keller |
| Power Regulation | Keven Hyppolite | Luz Romero |
| Growth Lamp | Keven Hyppolite | Luz Romero |
| Water Pump | Keven Hyppolite | Luz Romero |
| Moisture Sensor | Luz Romero | Keven Hyppolite |
| Water Level Sensor | Luz Romero | Keven Hyppolite |
| PH Sensor | Luz Romero | Keven Hyppolite |
| Temperature Sensor | Luz Romero | Keven Hyppolite |

Table 10.2: Report Page Breakdown

| Section | Primary | Secondary |
|---|---|---|
| Ch1 Executive Summary | Luz Romero | Keven Hyppolite |

| | | |
|---|---|---|
| Ch2 Project Description | Everyone contributed 2.5 pages as part of the initial DNC document | |
| Ch 3 Research and Investigation | Everyone researched their own parts for their section and wrote pages | |
| Ch4 Standards and Design Constraints | Luz Romero | Tony Chau |
| Ch5 Comparisons of ChatGPT | Steven Keller | Tony Chau |
| Ch6 Hardware Design | Keven Hyppolite | Steven Keller |
| Ch7 Software Design | Tony Chau | Steven Keller |
| Ch 8 System Fabrication | Steven Keller | Keven Hyppolite |
| Chapter 9 System Testing | Steven Keller | Luz Romero |
| Ch 10 Admin content | Everyone contributed to part of it during the DNC document | Steven Keller |

## 10.2 Budget and Financing

During our research, we meticulously sought out the most cost-effective items that enabled us to achieve our project goals without compromising on quality. Our goal was to design an efficient and reliable smart system that ensured optimal care for plant health. To manage our resources effectively, we set a budget limit of $400.00. This budget covered essential components such as sensors, a microcontroller, a water pump, power supplies, and any additional hardware required for system integration. The table below provides a detailed estimate of the expenses we anticipated for the necessary components identified thus far, ensuring transparency and careful financial planning throughout the project's development. We were also incentivized to reduce costs, as we were entirely self-funded. Our budget was split four ways among each group member.

*Table 10.3: Budget table*

| Item | Estimated Cost | Actual Cost |
|---|---|---|
| **Relay Module** | $10 | $6.12 |
| **pH sensor** | $35 | $30.99 |
| **Temperature sensor** | $15 | $2.03 |
| **Water level Sensor** | $9 | $59.95 |

| | | |
|---|---|---|
| **Moisture Sensor** | $12 | $7.50 |
| **LED** | $7.19 | $15 |
| **MCU (ESP32)** | $10 | $10 |
| **ATEMGA328** | $10 | $9 |
| **DC Water Pump** | $13 | $13 |
| **Phone** | $70 | $50 |
| **PCB** | $30 | $120 |
| **Wood Box** | $100 | $131 |

## 10.3 Project Milestone

Initially, our project focused on the design and selection phase, where we finalized the choice of components, such as sensors, actuators, and the microcontroller. This was followed by the prototyping phase, during which we assembled a working prototype of the system, integrating hardware and developing preliminary firmware. Our next milestone provided Senior Design I and II with achievable goals:

*Table 10.4: Senior Design I & II schedule table*

| Task | Duration | Status |
|---|---|---|
| **Senior Design I** | **5/13 – 8/02** | **Current** |
| Brainstorming / Project selection / Group Formation | 5/14 – 5/21 | Done |
| Divide and Conquer 10-page Document | 5/21 – 5/31 | Done |
| Research | 5/30 – 6/6 | Done |
| Materials research | 5/30 – 6/6 | Done |
| Paper (25 pages) | 6/3 – 6/20 | Done |
| Paper (40 pages) | 6/20 – 6/27 | Done |
| Paper (60 pages) | 6/27 – 7/5 | Done |
| Demo | 7/5 – 7/12 | Done |
| 120 Page Final | 712 – 7/23 | Done |

| | | |
|---|---|---|
| Document / Demo | | |
| Order Parts for PCB | Early August | Done |
| **Senior Design II** | **8/19 – 12/07** | |
| PCB Assembly | Late August | Done |
| Testing and Redesign | September | Done |
| PCB Assembly | October | Done |
| Testing | November | Done |
| Final Presentation | November | Done |

# 11. Conclusion

In conclusion, our automatic plant watering system project was designed to revolutionize home gardening by integrating advanced technology with user-friendly features. This innovative system employed a range of sensors—including those for soil pH, moisture, and water levels—alongside the ESP32-S microcontroller to provide precise and automated care for houseplants. The system was managed through a dedicated app, offering users the ability to monitor and control their plant care from anywhere, thus ensuring optimal plant health with minimal effort.

Our approach balanced performance and cost-efficiency by incorporating both high-quality and economical sensors, such as the GAOHOU PH0-14 sensor for soil pH measurement. By addressing constraints related to hardware selection, environmental factors, and budget considerations, we aimed to deliver a reliable and affordable solution.

Our team, composed of three Electrical Engineers and one Computer Engineer, was in the summer semester of Senior Design 1. During this period, we focused on the development of our printed circuit board (PCB). In the forthcoming fall semester, we advanced to the construction and full-scale implementation of the system. This phased approach, supported by our team's specialized skills and collaborative effort, positioned us to effectively achieve our project goals and deliver a sophisticated automatic plant watering solution.

This document offered a thorough overview of our automatic plant watering system project, outlining the design process, technical details, and implementation plan. It highlighted our approach to integrating hardware and software, justified our design choices, and mapped out the project milestones. By providing insights into system integration, sensor selection, and user interface development, this document helped stakeholders grasp the complexity of our design and the efforts invested in ensuring its success. It served as a guide for the development process and a reference for assessing the system's functionality and overall effectiveness.

## Appendix A: References

[ChatGPT]: We used ChatGPT to proofread our paragraphs

3.4.2

"Digikey." *ESP32-CAM Development Board*, media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0602_Web.pdf. Accessed 23 July 2024.

ESP32 series datasheet. (n.d.). https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

3.4.3

Arduino® Giga R1 WIFI. (n.d.-a). https://docs.arduino.cc/resources/datasheets/ABX00063-datasheet.pdf

Datasheet - STM32H747xI/G - dual 32-bit ARM®cortex. (n.d.-b). https://www.st.com/resource/en/datasheet/stm32h747ai.pdf

3.4.4

Raspberry pi documentation - raspberry pi hardware. (n.d.-d). https://www.raspberrypi.com/documentation/computers/raspberry-pi.html

3.4.5

MSP430G2553 PDF. (n.d.-d). https://www.ti.com/lit/ds/symlink/msp430g2553.pdf

3.4.8

ATMEGA328P. (n.d.-b). https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

3.4.9

I.MX RT1060 crossover processors for consumer products. (n.d.-e). https://www.nxp.com/docs/en/nxp/data-sheets/IMXRT1060CEC.pdf

3.4.10

RP2040 Datasheet. (n.d.-h). https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf

3.5.1

Notes, E. (n.d.). *IEEE 802.11n Wlan Standard*. Electronics Notes. https://www.electronics-notes.com/articles/connectivity/wifi-ieee-802-11/802-11n.php

3.5.2 and 3.5.3

*Core specification 5.4*. Bluetooth® Technology Website. (2024, July 17). https://www.bluetooth.com/specifications/core54-html/

*Things you should know about bluetooth range*. Welcome to our blogs. (n.d.). https://blog.nordicsemi.com/getconnected/things-you-should-know-about-bluetooth-range

3.5.4

Lora® and LoRaWAN®. (n.d.-f). https://www.semtech.com/uploads/technology/LoRa/lora-and-lorawan.pdf

3.6.1

OV2640DS.pdf. (n.d.-h). https://www.uctronics.com/download/cam_module/OV2640DS.pdf

3.6.2

Datasheet OV5640. (n.d.-d). https://cdn.sparkfun.com/datasheets/Sensors/LightImaging/OV5640_datasheet.pdf

3.6.3

OV7725.pdf. (n.d.-j). https://cdn.sparkfun.com/datasheets/Sensors/LightImaging/OV7725.pdf

5.2

Introducing chatgpt. (n.d.-g).

https://openai.com/index/chatgpt

Raghunath, A. (2024, June 18). *GPT-4 parameters explained: How many parameters in GPT-4 & more*. HIX.AI. https://hix.ai/hub/chatgpt/gpt-4-parameters

5.3

*Introducing Meta Llama 3: The most capable openly available LLM to date*. AI at Meta. (n.d.). https://ai.meta.com/blog/meta-llama-3/

Davis, J. J. (2024, July 5). *Meta Llama vs CHATGPT: A detailed comparison*. Medium. https://medium.com/illumination/meta-llama-vs-chatgpt-a-detailed-comparison-9794ccedd41c

5.4

*Claude vs CHATGPT: Which AI assistant suits you best in 2024?*. Semrush. (n.d.). https://www.semrush.com/goodcontent/content-marketing-blog/claude-vs-chatgpt/

5.5

Google. (n.d.). *What Gemini apps can do and other frequently asked questions*. Google. https://gemini.google.com/faq

*Gemini*. Google DeepMind. (2024, May 14). https://deepmind.google/technologies/gemini/#:~:text=Gemini%20models%20are%20built%20from,images%2C%20audio%2C%20and%20video.

9.2

Espressif. (n.d.). *Arduino-ESP32/license.md at master · espressif/arduino-ESP32*. GitHub. https://github.com/espressif/arduino-esp32/blob/master/LICENSE.md

*Copyrights and Licenses*. ESP. (n.d.). https://docs.espressif.com/projects/esp-idf/en/stable/esp32/COPYRIGHT.html

Licensing for products based on Arduino. (n.d.-h). https://support.arduino.cc/hc/en-us/articles/4415094490770-Licensing-for-products-based-on-Arduino

9.3

Digikey. (n.d.-e). https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/SEN0193_Web.pdf

## Appendix B: Copywrite

Espressif. (n.d.). *Arduino-ESP32/license.md at master · espressif/arduino-ESP32*. GitHub. https://github.com/espressif/arduino-esp32/blob/master/LICENSE.md

*Copyrights and Licenses*. ESP. (n.d.). https://docs.espressif.com/projects/esp-idf/en/stable/esp32/COPYRIGHT.html

Licensing for products based on Arduino. (n.d.-h). https://support.arduino.cc/hc/en-us/articles/4415094490770-Licensing-for-products-based-on-Arduino

## Appendix C: Technical

Digikey. (n.d.-e). https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/SEN0193_Web.pdf

OV2640DS.pdf. (n.d.-h). https://www.uctronics.com/download/cam_module/OV2640DS.pdf

RS485 & 4-20mA current PH sensor user manual. (n.d.). https://files.seeedstudio.com/products/101990666/res/RS485 & 4-20mA Current pH Sensor User Manual-S-pH-01.pdf

Datasheet - STM32H747xI/G - dual 32-bit ARM®cortex. (n.d.-b). https://www.st.com/resource/en/datasheet/stm32h747ai.pdf

ATMEGA328P. (n.d.-b). https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

"Digikey." *ESP32-CAM Development Board*, media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0602_Web.pdf. Accessed 23 July 2024.

ESP32 series datasheet. (n.d.). https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

Ti. (n.d.). https://www.ti.com/lit/gpn/tmp117

GAOHO PH0-14

https://www.scribd.com/document/609479894/ph

S-PH-01-PH

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://files.seeedstudio.com/products/101990666/res/RS485%20&%200-2V%20pH%20Sensor%20(S-pH-01)%20-%20User%20Guide%20v2.0.pdf

TMP117AIDRVR

https://www.ti.com/product/TMP117/part-details/TMP117AIDRVR

DS18B20

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.analog.com/media/en/technical-documentation/data-sheets/ds18b20.pdf

5'' eTape Liquid Level Sensor with Plastic Casing

https://www.adafruit.com/product/3828?gad_source=1&gclid=CjwKCAjwqf20BhBwEiwAt7dtdUM5f521pA5eWZ81Miy452qrnQgek3WCSSg0jQl8sYGqHLY4TBxphRoCGnQQAvD_BwE

HC-SR04

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

Grove-Moisture Sensor

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.mouser.com/datasheet/2/744/Seeed_101020008-1217463.pdf

adafruit STEMMA

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://cdn-learn.adafruit.com/downloads/pdf/adafruit-stemma-soil-sensor-i2c-capacitive-moisture-sensor.pdf

Keyes Relay Module

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://roboticafacil.es/datasheets/ky-019.pdf

df robot rely shield

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/2357/DFR0144_Web.pdf

LM7805

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.sparkfun.com/datasheets/Components/LM7805.pdf

Amazon. (n.d.). ESP32-CAM-MB, a development board with camera module OV2640. Amazon. https://www.amazon.com/dp/B09SF1C9DQ?ref=ppx_yo2ov_dt_b_product_details&th=1

Amazon. (n.d.). HLG 65 V2 4000K Horticulture Lighting Group Quantum Board LED Grow Lamp Veg & Bloom. Amazon. https://www.amazon.com/Horticulture-Lighting-Group-USA-Quantum/dp/B07C57J7XX/ref=sr_1_2_sspa?crid=3JFJZ8DCO32IV&keywords=vipar+v100&qid=1721749072&s=hi&sprefix=vipar+v100%2Ctools%2C276&sr=1-2-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&psc=1

Amazon. (n.d.). Spider Farmer SF-1000 LED Grow Light with Samsung LM301B Diodes & Dimmable MeanWell Driver Sunlike Full Spectrum for Indoor Plants Veg and Flower. Amazon. https://www.amazon.com/Spider-Farmer-Dimmable-MeanWell-Spectrum/dp/B07TS82HWB/ref=sr_1_3_sspa?crid=3JFJZ8DCO32IV&keywords=vipar+v100&qid=1721749072&s=hi&sprefix=vipar+v100%2Ctools%2C276&sr=1-3-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&psc=1

Amazon. (n.d.). Phlizon Newest 1200W LED Plant Grow Light Four Head Full Spectrum with Samsung LM301B LEDs. Amazon. https://www.amazon.com/Phlizon-Dimmable-Spectrum-Waterproof-Hydroponics/dp/B083SDVKTQ/ref=sr_1_4_sspa?crid=3JFJZ8DCO32IV&keywords=vipar+v100&qid=1721749072&s=hi&sprefix=vipar+v100%2Ctools%2C276&sr=1-4-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9tdGY&psc=1

Amazon. (n.d.). AGLEX 2000W LED Grow Light Dimmable Full Spectrum Plant Grow Lights with Daisy Chain Function. Amazon.

https://www.amazon.com/AGLEX-Dimmable-Waterproof-Hydroponic-Greenhouse/dp/B089CR9N5X/ref=sr_1_1_sspa?dib=eyJ2IjoiMSJ9.Dz_eRxlbZiRAOmfR9GFl6tz3oy7zxtqOnHf4iyQaD0ACnLPpzAyN37sSuMBl0FjyHo-m2sAQmTg4xm8u58xt6dq0jbwy9z9T9pwWKraO1fKi_VAyPHfSGilklqI-VAr1Gqxq1_b_TXsS2FCpNew0mFfnlDY7GGajQ9pMaQ7tLTfImUGbkX5NRoMEMA3B5nhS6GLYm_y_AI3YJuijpRYVoTMh9OGvg_mopKxez8QErv4.MOq2SRN547DY3QVE7qTZ9nyvnr-RT60xvtVTeRC66Ng&dib_tag=se&keywords=aglex&qid=1721749514&sr=8-1-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&th=1

Amazon. (n.d.). Linkind A19 LED Light Bulbs, Non-Dimmable, 60W Equivalent Soft White 2700K Standard Replacement. Amazon. https://www.amazon.com/Linkind-Non-Dimmable-Equivalent-Standard-Replacement/dp/B08L37F4BW/ref=sr_1_2_sspa?crid=DBBDF9I44C2P&dib=eyJ2IjoiMSJ9.5UgrWPo80mK2fidCBkQD_8XLyQllfm5W_NIpONbEpStp2RmEt7KkqnpFq2ccZAvN2YaazAixM1SljT5GlaEPw6Xx3ES4G52NOkg_4lXx6z_G9J6Wshr-hgVAzN_tkOyLTVjymvt2b4wANss13iu6tDyq_MtivgIkK1m1SPPTpVXA5O_l55GC8ml4uFL8GvFpuAtye1JVB4jX3dyqm_5IpmQJZZVJs3Z_QzGKCXbUyyiTeMbDG7tVZp3zlFGjsJnO8FQ5PZGdXokKS5fgzrstDiirVAhyFpkg2lE2J9pq9_g.js2VU1UAqioiNqbsebAUiej85WyYEcIGFnOhpMpi4qs&dib_tag=se&keywords=light+bulbs+60+watt+soft+white&qid=1721749559&sprefix=light+bulbs%2Caps%2C285&sr=8-2-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&psc=1