

GreenSync: Engineering Innovation for Sustainable Agriculture



Group 12:

Roman Luknicki

Rohit Goeptar

Thomas Parker

Victor Arteaga Cuenca

Electrical Engineering

Computer Engineering

Electrical Engineering

Computer Engineering

Reviewers and Committee:

Dr. Mark Maddox

Dr. Mike Borowczak

Dr. Saleem Sahawneh

Dr. Justin Phelps

Mentor:

Dr. Lei Wei

Dr. Chung Yong Chan

Table of Contents

Chapter 1	Executive Summary	11
Chapter 2	Project Description	12
2.1	Project Background	12
2.2	Motivation	13
2.3	Existing Projects	14
2.3.1	Commercial Solutions	14
2.3.2	DIY and Open-Source Projects	15
2.3.3	Comparative Analysis	16
2.4	Goals and Objectives	16
2.5	Description of Features & Functionalities	21
2.6	Engineering Requirements	22
2.7	House of Quality	26
2.8	Hardware	27
2.8.1	Hardware Diagram	28
2.9	Software	28
2.9.1	Software Diagram	30
Chapter 3	Research	30
3.1	Processing Control, & Power	32
3.1.1	Processing Unit Technologies	32
3.1.1.1	Microcontroller Units	34
3.1.2.	Clock Source Options	37
3.2	Water Delivery & Moisture System	39
3.2.1	Water Delivery Methods	39
3.2.2	Water Pump	42

3.2.3	Moisture Sensing Technologies	46
3.2.4	Capacitive Soil Moisture Sensor	49
3.2.5	Release Valve	52
3.2.6	Water Reservoir	56
3.3	Lighting Systems	59
3.3.1	Light Distribution Systems	59
3.3.2	Grow Light Technologies	61
3.3.3	Light Sensing Technologies	64
3.3.4	LED Grow Light	67
3.4	Camera Integration	72
3.4.1	Camera Hardware Technologies	72
3.4.2	Camera Hardware Components	74
3.5	Communication & Integration	76
3.5.1	Wireless Communication Protocols	76
3.5.2	Sensor Integration Methods	78
3.6	Software Systems	81
3.6.1	Data Logging & Storage Options	81
3.6.2	Embedded Programming Environment	83
3.6.3	Programming Language	85
3.6.4	User Interface	87
3.6.4.1	Comparisons of App Development Technologies	89
3.6.4.2	Mobile App Storage	94
3.6.5	Custom PCB Design Software	95
3.7	Environmental Design Factors	97
3.7.1	Enclosure and Housing Design	97
3.7.2	Soil Type Considerations	99

3.8	Power Source	99
3.8.1	Power Supply	99
3.8.2	Wall Adapter	101
3.8.3	Voltage Regulation Modules	103
Chapter 4	Standards and Design Constraints	105
4.1	Standards	105
4.1.1	Safety and Health Concerns	105
4.1.1.1	Adherence to the Standards for Health and Safety	105
4.1.1.2	The Electrical Safety Standards	106
4.1.1.3	Standards for the Design of PCBs	106
4.1.1.3.1	PCB Standard IPC-2221	107
4.1.1.4	Standards for Wireless Communication and Connectivity	111
4.1.1.4.1	Wireless Comm Standard IEEE 802.11	111
4.1.1.4.2	ESP-NOW Wireless Communication	113
4.1.1.5	Standards for Data Security and Privacy	116
4.1.1.6	Standards for Smart Lighting Components	116
4.1.1.7	Standards for Sustainability and Energy Efficiency	117
4.1.1.8	Standards for Sensors and the Environment	117
4.2	Constraints	118
4.2.1	Constraints of Sustainability	118
4.2.1.1	Battery Longevity	118
4.2.2	Optimizing Battery Longevity	119
4.2.3	Reliability and Maintenance Constraints	119
4.2.3.1	The Process of Corrosion	119
4.2.3.1.1	Erosion Caused by Electrolysis	120
4.2.3.1.2	Corrosion Caused by Galvanic Current	121

4.2.3.1.3 Preventing Corrosion	122
4.2.3.2 The Presence of Salt	123
4.2.4 Constraints due to the Environment	124
4.2.5 Limitations on the Security	125
Chapter 5 Comparison of ChatGPT with Other AI	126
5.1 Navigating the Landscape of AI-Assisted Research	126
5.2 Case Studies in AI-Driven Component Analysis	127
5.2.1 Example 1: Water Pump Selection	127
5.2.2 Example 2: Sensor Battery Life Optimization	128
5.3 Triangulating Solutions with Competing AI Models	129
5.3.1 ChatGPT	129
5.3.2 DeepSeek	129
5.3.3 Claude	129
5.3.4 AI Comparison Synthesis and Conclusion	130
5.4 Critical Assessment of AI Tools in an Academic Setting	130
5.5 Concluding Analysis on Human-AI Collaboration	133
Chapter 6 Hardware	133
6.1 MCU PCB Schematic	133
6.1.1 ESP32	134
6.1.2 Regulators	134
6.1.3 I2C Multiplexer	136
6.1.4 Sensors	137
6.2 Water Pump Subsystem Block Diagram	138
6.3 Moisture Sensor	140
6.3.1 Voltage Regulation	142
6.4 Lighting Subsystem Block Diagram	143

6.5	Shelly RGBW2 Driver Schematics	145
6.6	Camera	146
Chapter 7	Software	148
7.1	Core System Software Design	148
7.1.1	Development Environment and Tools	151
7.1.2	Sensor Module Firmware	151
7.1.3	Central Control Unit Logic	152
7.2	Mobile Application Framework	155
7.2.1	Application Development Stack	156
7.2.2	User Interface and Experience	156
7.2.3	User Interface Mobile Application	157
7.3	System Calibration and Constants	159
7.3.1	Soil Moisture Calibration Constants	160
7.3.2	Network Communications Constants	160
Chapter 8	System Fabrication	152
8.1	Layout of the PCB	161
8.1.1	Board for the 3.3V Regulator	161
8.1.2	The PCB Prototype Layout	163
Chapter 9	System Testing and Evaluation	163
9.1	ESP32 Testing	164
9.2	Moisture Sensor Testing	165
9.3	Ambient Light Sensor Testing	165
9.4	Camera Module Testing	166
9.5	Plan for Senior Design 2	167
Chapter 10	Administrative Content	169
10.1	Component Comparison	169

10.2	Project Milestones	170
10.3	Workflow Distribution	171
Chapter 11	Conclusion	172
Appendix A	References and Data Sheets	173
Appendix B	Software Code	184
Appendix C	ChatGPT Prompts and Outcomes	190

List of Tables

Table 1	Specifications of Engineering Requirements	23
Table 2	Processing Unit Technologies	34
Table 3	Microprocessor Units	36
Table 4	Clock Source Options	38
Table 5	Water Delivery Methods	41
Table 6	Water Pumps	44
Table 7	Pump Components	45
Table 8	Moisture Sensor Technologies	49
Table 9	Capacitive Moisture Sensor	51
Table 10	Release Valves	54
Table 11	Valve Components	56
Table 12	Water Reservoir	58
Table 13	Light Distribution Methods	60
Table 14	Grow Light Technologies	63
Table 15	Light Sensing Technologies	65
Table 16	Light Sensor Component	67
Table 17	Grow Light Component	69

Table 18	Camera Technologies	71
Table 19	Camera Component	72
Table 20	Wireless Protocols	75
Table 21	Sensor Integrations	77
Table 22	Storage	80
Table 23	Embedded Environment	82
Table 24	Programming Language	84
Table 25	Interface	86
Table 26	Basic App Development Technology Comparison	87
Table 27	Comprehensive Pros and Cons of Every Technology	87
Table 28	Basic App Storage Technology Comparison	92
Table 29	Flutter's Local Storage Options	92
Table 30	PCB Software	94
Table 31	Enclosure	96
Table 32	Soil Types	98
Table 33	Power Supply	100
Table 34	Comparison of Power Supply Adapters	102
Table 35	Voltage Regulators	104
Table 36	List of Categories and Descriptions for the IPC-2221 Specification	110
Table 37	Comparison of Electrochemical and Galvanic Deterioration	122
Table 38	Moisture Test	165
Table 39	Light Testing	166
Table 40	Bill of Materials	169
Table 41	Project Milestones Table	170
Table 42	Work Distribution Table	171

List of Figures

Figure 1	House of Quality	26
Figure 2	Hardware Diagram	28
Figure 3	Software Diagram	30
Figure 4	PCB Schematic	134
Figure 5	Regulator Schematic	136
Figure 6	I2C Multiplexer Schematic	137
Figure 7	AHT20 Sensor Connection	138
Figure 8	VEML770 Sensor Circuit	138
Figure 9	Water Pump Subsystem Diagram	140
Figure 10	Typical Moisture Sensor Schematic	142
Figure 11	Lighting Subsystem Block Diagram	144
Figure 12	Shelly RGBW2 Driver Schematic	146
Figure 13	ESP32 Camera Connection Schematic	148
Figure 14	Software State Block Diagram	150
Figure 15	GreenSync Sensor Module Operational Cycle	152
Figure 16	Central Control Unit Decision Logic Flowchart	153
Figure 17	System Initialization Process	154
Figure 18	Mobile App Flowchart	158
Figure 19	Users Mobile Interface	159
Figure 20	Regulators Designed Through Fusion360 (1)	162
Figure 21	Regulators Designed Through Fusion360 (2)	156

Chapter 1 - Executive Summary

GreenSync is an automated smart gardening system designed to simplify plant care through intelligent monitoring and control. The system addresses a

growing need for efficient, low-maintenance solutions for home and small-scale gardeners who face challenges such as inconsistent watering, inadequate lighting, and lack of plant health visibility. By combining environmental sensing, wireless communication, and automated actuation into a compact, affordable platform, GreenSync ensures plants receive the right conditions at the right time without constant user intervention.

At its core, GreenSync revolves around three primary subsystems: watering, lighting, and visual monitoring. A capacitive soil moisture sensor continuously monitors soil hydration levels. When moisture falls below a preset threshold, the microcontroller (ESP32) activates a miniature water pump to deliver water from a reservoir through controlled tubing. The lighting subsystem uses an ambient light sensor to detect illumination conditions and automatically powers an LED grow light when natural sunlight is insufficient. These two environmental control loops ensure that the plant's fundamental needs, water and light are met dynamically rather than through static schedules.

To complement these automated functions, GreenSync incorporates a camera module for real-time plant imaging, offering users a visual record of plant health and growth progress. The system also supports remote connectivity via Wi-Fi or Bluetooth Low Energy (BLE), allowing users to monitor conditions, adjust thresholds, and access data through a mobile application or web dashboard. This integration not only enhances user convenience but also aligns with modern IoT trends, where home automation extends to sustainable plant care.

While designed as a single-plant prototype, GreenSync is being developed with scalability in mind, ensuring that the same design principles can be applied to larger systems or multi-plant networks in the future. This flexibility reflects the project's long-term vision of merging cost-effectiveness with adaptability for broader applications.

The design of GreenSync was guided by extensive research documented in Chapter 3, covering technology alternatives, component trade-offs, and integration methods. From evaluating moisture sensing technologies to assessing grow light options, every decision balanced cost, reliability, and expandability. By combining affordability with adaptability, GreenSync demonstrates how practical engineering can merge with smart technology to create a product that not only works but sets a foundation for future innovation.

Chapter 2 - Project Description

2.1 Project Background

For thousands of years, agriculture has depended on the careful observation of natural elements. Early civilizations like those in Mesopotamia and along the Nile developed irrigation systems to channel water to their crops, relying heavily on environmental cues to guide planting and harvesting. The success of early farming was directly tied to how well people could monitor sunlight, soil conditions, and water availability, long before the invention of modern tools or automation. This deep-rooted relationship between humans and plants formed the basis of not only agricultural development but also societal growth.

As technology and infrastructure evolved, large-scale agriculture became more industrialized, reducing the need for individuals to grow their own food. However, interest in plants never disappeared, it simply shifted. In urban areas and residential spaces, people began caring for houseplants, small gardens, and community green spaces as a way to reconnect with nature, improve indoor air quality, and enhance living environments. In particular, the rise of indoor gardening and the popularity of houseplants in the 21st century has transformed plant care into both a creative hobby and a method of personal wellness.

This cultural shift has coincided with rapid advancements in consumer technology. The growing accessibility of sensors, microcontrollers, and open-source platforms has enabled individuals to explore how everyday tasks like watering a plant can be improved through automation and real-time monitoring. As interest in sustainability and smart living has grown, more people have turned to do-it-yourself (DIY) solutions to solve problems in their personal environments. Online communities of hobbyists and students now regularly build simple systems that integrate electronics with nature, learning engineering principles in the process.

These developments represent a fusion of traditional practices and modern innovation. While the fundamentals of plant care: light, water, and attention remain unchanged, the tools available to support those tasks have become far more advanced. The background of this project lies in that intersection: where ancient agricultural wisdom meets new technological possibilities, and where hands-on experimentation helps bring the future of home gardening to life.

2.2 Motivation

Houseplants have seen a significant rise in popularity, especially in recent years, as people seek to enhance their living environments with natural beauty, improved air quality, and a sense of calm. However, despite their benefits, many indoor plants are poorly maintained due to simple but

impactful mistakes. Studies and surveys have shown that a large percentage of houseplants are lost each year due to either neglect or improper care. Among the most common issues are **overwatering**, which can lead to root rot and the proliferation of fungus gnats, and **underwatering**, which stresses the plant and inhibits growth. Additionally, **improper sunlight exposure**, whether too much or too little, can result in scorched leaves, poor color, and even plant death. These challenges are not uncommon among casual plant owners and can be discouraging, especially when plants that are meant to bring joy become a source of stress.

This dilemma inspired our group of four electrical and computer engineering students to develop a smarter, more reliable approach to plant care. As engineering students, we're constantly looking for ways to apply our technical knowledge to solve real-world problems, and this project offered the perfect intersection of embedded systems, sensor integration, power management, and user interaction design. It allows us to practice and demonstrate skills in microcontroller programming, analog and digital signal processing, and system-level integration—skills we've developed throughout our coursework and labs. The Smart Garden System not only fits well within our academic field, but also gives us the freedom to be creative and work on something that's meaningful to us personally.

Each of us has, at some point, struggled to keep a plant healthy, whether due to a busy schedule, lack of knowledge, or inconsistent conditions in our homes. This personal connection to the problem strengthens our motivation, as we want to build something that we would use ourselves. There's something deeply rewarding about nurturing a plant and watching it grow, and this project gives us the opportunity to make that experience more accessible and successful for others.

Beyond our own interests, we recognize the broader value of plant care in households. Plants like the **snake plant** are known to filter airborne toxins such as formaldehyde, benzene, and xylene, improving indoor air quality and contributing to a healthier home. Studies have also shown that indoor plants can help reduce stress and improve mood—benefits that are especially important in today's fast-paced, often digitally overwhelming environments. Maintaining healthy plants isn't just a hobby; it's a lifestyle choice that supports wellness, sustainability, and mental clarity.

The **post-pandemic gardening boom** has also played a role in our decision. Many people began turning to indoor and balcony gardens during lockdowns as a way to stay grounded and productive. Even now, the trend of **natural living and homegrown food** continues to grow, with individuals and families looking for ways to bring a little more green into their homes. Our Smart

Garden System aligns perfectly with this shift and could eventually evolve into something scalable—perhaps a larger, multi-plant system, or even a community garden platform in the future.

Ultimately, this project represents a fusion of our technical training and our appreciation for the natural world. It challenges us to think critically about automation and sustainability, while offering a product that could make a real difference in people's lives. For us, it's more than just a senior design project—it's a chance to build something practical, meaningful, and potentially expandable far beyond its initial scope.

2.3 Existing Projects

The intersection of technology and horticulture has given rise to a variety of smart plant care solutions, ranging from commercial products to DIY projects. These innovations aim to simplify plant maintenance by automating tasks such as watering, lighting, and monitoring environmental conditions.

2.3.1 Commercial Solutions

LetPot Automatic Watering System 2.0: This system automates plant watering by delivering controlled water amounts to each plant's base through adjustable drippers, all managed via a smartphone app. It's suitable for different scenarios, including indoor and outdoor plants, and offers features like customizable watering schedules and real-time monitoring.

GrowCube Smart Watering System: GrowCube is an intelligent plant watering system based on four independent soil moisture sensors and water outlets. Using an exclusive app, it allows users to monitor soil moisture in real-time and control watering accordingly. The system supports up to four plants simultaneously and provides detailed data visualization for plant care.

Click and Grow Smart Garden: Click and Grow offers a range of indoor gardening products, including the Smart Garden series. These systems use self-irrigating technology and "plant pods" containing seeds, nutrients, and a growing medium. The Smart Garden 3 and 9 models are designed for home use, while the Wall Farm can grow up to 51 plants, catering to more extensive indoor gardening needs.

Gardyn Home System: Gardyn provides a high-tech indoor gardening solution that allows users to grow up to 30 plants, including vegetables, fruits, and herbs, indoors with minimal effort. The system features full-spectrum LED

lighting, AI-based plant care monitoring via the Gardyn app, and a sleek design suitable for modern living spaces.

2.3.2 DIY and Open-Source Projects

Arduino-Based Smart Gardens: Enthusiasts have developed various DIY smart garden projects using Arduino microcontrollers. These projects often include soil moisture sensors, temperature sensors, and light sensors to monitor plant conditions. For instance, one project integrates an Arduino Mega with a TFT touchscreen to display real-time data and control watering schedules.

Raspberry Pi IoT Garden: Leveraging the power of the Internet of Things (IoT), some DIY projects utilize Raspberry Pi to create smart gardens. These systems can monitor real-time status of plants, including soil moisture and temperature, and allow users to access data and control watering through a smartphone app.

OpenAg Personal Food Computer: Developed as an open-source platform, the Personal Food Computer is designed for controlled-environment agriculture. It allows users to monitor and control environmental variables such as temperature, humidity, and light, providing a comprehensive system for plant growth research and education.

2.3.3 Comparative Analysis

While commercial systems like LetPot and GrowCube offer user-friendly interfaces and plug-and-play functionality, they may come with limitations in terms of customization and scalability. On the other hand, DIY projects provide flexibility and a deeper understanding of the underlying technology but require a higher level of technical expertise and time investment.

Our proposed Smart Garden System aims to bridge the gap between these two approaches by offering a customizable, scalable, and user-friendly solution. By integrating affordable sensors, microcontrollers like the ESP32, and optional features such as solar power and mobile app integration, our system aspires to provide a comprehensive plant care solution suitable for both novice and experienced users.

2.4 Goals and Objectives

Goals:

This project aims to provide an efficient and reliable solution for monitoring and maintaining the health of indoor plants using automation, environmental sensing, and user-centered design. The Smart Garden System is intended to reduce the burden of daily plant care while promoting sustainability and increased user engagement. The platform combines hardware and software subsystems to create a modular, extensible system capable of adapting to a wide variety of plant types and environments. The goals listed below represent the broad, high-level vision for the system, with each goal followed by specific technical objectives that define the key deliverables needed to fulfill it.

Goal 1: Build an optimized plant environment for health and growth

To ensure plant longevity and promote consistent development, this goal focuses on sensing and maintaining the plant's physical environment primarily hydration and sunlight exposure through intelligent, automated responses.

- Develop a soil moisture sensing system capable of continuously monitoring the plant's hydration level.
- Implement a light intensity sensing system to measure and log the amount of ambient sunlight the plant receives throughout the day.
- Record real-time environmental data from the soil moisture and light sensors at regular intervals for analysis and feedback.
- Establish a structured data storage system to archive all recorded sensor data for long-term tracking and comparison.
- Include plant care profiles with ideal moisture and light ranges for various species, allowing the system to interpret sensor data accordingly.
- Define automated threshold logic to trigger watering or light supplementation when environmental readings fall outside optimal ranges.
- Design a stable power delivery system with support for future expansion to solar or battery options.
- Calibrate all sensors against known values to ensure accurate readings for reliable feedback control.

Goal 2: Develop a Camera System for Daily Growth Tracking

To visualize and document plant growth over time, the system will include a time-lapse camera module. This feature not only enhances user engagement but also serves as a useful tool for analyzing environmental effects on plant

development. Captured images will be stored with timestamps, allowing for playback of the plant's progress over days or weeks.

- Integrate a compact, low-power camera module capable of capturing periodic images (e.g., once per day).
- Automate the image capture process using the microcontroller, triggered by a daily timer.
- Store images locally with metadata (date/time) for future retrieval and time-lapse generation
- Ensure power-efficient operation of the camera system to support long-term deployment.
- Design system architecture to allow later integration with cloud or mobile app viewing features.

Goal 3: Create a Functional Mobile Application Interface

The mobile application will serve as the primary interface for users to access their plant's environmental data in a clean and readable format. The goal is to provide users with an easy-to-understand interface for tracking their plant's condition over time without requiring complex interaction or real-time processing. This will help support user awareness and engagement while maintaining system simplicity.

- Develop a mobile app interface that displays current moisture and light readings in an organized and user-friendly format.
- Enable the app to store and display past sensor readings over time, allowing users to view environmental changes and identify patterns.
- Configure the app to receive and display data transmitted from the system's microcontroller at fixed time intervals.
- Integrate a basic plant database within the app that allows users to select their plant type and apply pre-programmed optimized care levels.

Goal 4: Create a Modular and Expandable System Design

This goal focuses on ensuring the project is built with long-term scalability in mind. By designing the hardware and software architecture to be modular, future additions such as more sensors, plant zones, or new features can be integrated with minimal rework. This also allows other developers or users to customize and adapt the system to suit different use cases or plant types beyond the initial prototype.

- Design the system architecture to support multiple sensor inputs and actuator outputs, allowing additional plant zones to be added in the future.
- Organize the hardware layout, including the custom PCB, to allow for additional modules such as temperature sensors, humidity monitors, or alternative lighting.
- Ensure software components, including data logging and mobile interface functions, are structured to accommodate future expansion without major redesign.
- Enable clear documentation and code structure to support future teams in continuing or building upon the project after the initial implementation.

Objectives:

Project Objectives

Water Distribution System Objectives

- Deliver water to the plant using a compact pump or solenoid valve system controlled by the microcontroller
- Respond to moisture sensor data and initiate watering when hydration levels fall below a pre-set threshold
- Regulate water flow duration for each watering cycle to avoid overwatering
- Allow for simple calibration and tuning of moisture thresholds during testing and refinement

Light Management System Objectives

- Use an LED grow light to provide supplemental lighting in environments with low natural sunlight
- Activate the light module based on light sensor input and defined exposure thresholds
- Support scheduled or triggered lighting routines that simulate a healthy day-night cycle
- Ensure efficient power usage to minimize energy consumption and heat buildup

App and User Interface Objectives

- Develop a mobile application that displays soil moisture and light readings from the system

- Visualize past sensor readings through a basic history log or time stamped record system
- Allow the user to select a plant profile from a preset database with optimized care values
- Receive and display sensor data via Bluetooth communication from the microcontroller
- Create a clean and user-friendly layout tailored for casual plant owners

Camera System Objectives

- Integrate a small camera module to capture daily images of the plant
- Store images locally or forward them to the app for later viewing
- Associate each image with a timestamp to enable time-lapse creation
- Operate with low power consumption between image captures

Environmental Sensor Objectives

- Accurately measure soil moisture and ambient light conditions using reliable sensors
- Collect and log data at consistent intervals to track environmental trends
- Ensure sensor accuracy through calibration and validation during integration
- Reduce power usage by placing sensors into low-power modes between active readings

Microcontroller and System Integration Objectives

- Manage all system functions including data collection, actuator control, and communication
- Coordinate timing and logic to run the watering, lighting, and camera subsystems
- Package sensor and system data for Bluetooth transmission to the app
- Interface with a custom PCB for streamlined hardware integration and power routing

Stretch Goals / Future Additions

Notification and Alerts

- Add a mobile alert system to notify users of low moisture, poor lighting, or other plant stress indicators
- Include customizable thresholds so users can set personalized alert conditions



Environmental Expansion

- Integrate temperature and humidity sensors to monitor additional environmental factors
- Create alerts when values fall outside the optimal range for a selected plant type



Enclosed Growth Environment

- Design a mini greenhouse enclosure with passive or active climate control features
- Maintain a stable microclimate for sensitive plants or exotic species



Edible Plant Support

- Expand system profiles to support fruits, vegetables, and herbs
- Include care parameters for different growth phases like seedling, flowering, and fruiting



Modular Scalability

- Support multi-plant setups with separate sensor-actuator modules per plant
- Enable easy hardware expansion through plug-and-play connectors or addressable nodes



AI Plant Identification

- Use the onboard camera and an integrated AI model to recognize plant species
- Auto-load matching care profiles based on plant type for optimal automation



Cloud Integration

- Sync data to the cloud for backup, multi-device access, and remote monitoring
- Enable long-term trend analysis and personalized plant insights via web or mobile dashboard



Growth Analytics

- Analyze image and sensor data to estimate plant growth trends and detect early signs of stress
- Provide predictive care suggestions based on historical patterns

2.5 Description of Features & Functionalities

The smart plant care system is designed to automate and optimize the maintenance of houseplants through a user-friendly, scalable platform. Its general functionality revolves around monitoring environmental variables, delivering targeted care, and providing meaningful user feedback. This system is intended to serve both novice and experienced plant owners, offering reliable and informative features to support healthier plant growth.

A core feature of the system is its modular scalability. The initial prototype is built to support a single plant, but it is designed to expand easily to accommodate multi-plant configurations. Each module includes its own sensor suite and actuator components to ensure individual plant needs are met without interference. This allows the system to grow alongside a user's plant collection, creating a flexible long-term solution.

The system also integrates a basic user interface designed for accessibility. The UI provides real-time sensor data, historical trends, and options to configure care parameters such as light thresholds or watering schedules. For enhanced interaction, the project envisions a mobile application interface that may eventually support remote monitoring and manual overrides.

Another general capability of the system is the ability to function semi-independently when unattended. The goal is to ensure consistent plant care even while users are traveling or otherwise unavailable. Timed watering, light triggering, and data logging are all features intended to reduce the amount of manual upkeep required.

Additionally, the project is built with educational and sustainability-oriented goals. Users can learn more about their plant's needs over time and may consult a built-in database for optimal growth conditions. Power options may include standard DC supply or, in future revisions, solar panel integration for low-energy off-grid setups. Overall, the general system behavior emphasizes reliability, user empowerment, and extensibility.

2.6 Engineering Requirements:

Table 1 outlines the key engineering specifications defined for the Smart Garden System. Several of these specifications have been highlighted as testable benchmarks for the final demonstration in SD2. These values represent critical system requirements tied to core features such as automated watering, light activation, and camera capture. Each specification

was selected to reflect important design priorities including power efficiency, sensor responsiveness, system scalability, and integration with typical indoor plant environments. By focusing on these essential parameters, the team can validate the system’s reliability and effectiveness during development. These specifications also provide a clear framework for testing, helping identify performance gaps early and ensuring the final implementation aligns with user needs. Ultimately, this structured approach supports both a successful prototype and future system expansion.

Table 1: Specifications of Engineering Requirements

CATEGORY	PARAMETER	REQUIREMENT
Water System	Soil Moisture Polling Rate	At least once per minute
	Watering Activation Delay	Less than 10 seconds after threshold is crossed
	Water Delivery Accuracy	±10% of target watering duration
	Water Pump Flow Rate	At least 1 Liter per Minute
	Water Pump Response Time	Less than 3 seconds
	Water Reservoir Capacity	At least 500 mL
	Moisture Sensor Lifetime	At least 6 months (under standard use)
	Moisture Sensor Accuracy	±5% volumetric water content
	Water Pump Power Consumption	Less than 75 Watts
Light System	Light Sensor Polling Rate	At least once every 2 minutes
	Grow Light Power Consumption	Less than 50 Watts
	Grow Light Response Time	Less than 5 seconds

Camera System	Image Capture Frequency	Once per day
	Image Resolution	Minimum 640×480 pixels
	Image Storage Capacity	At least 30 days of daily images (locally)
	Timestamp Accuracy	Within ±1 minute of scheduled capture
	Camera Wake & Capture Time	Less than 10 seconds
User Interface	Data Sync Interval	At least once per hour
	Supported Plant Profiles	Minimum of 5 preloaded species
	Sensor Data Display Delay	Less than 15 seconds after MCU transmits
System Power	System Power Source	12V DC adapter (baseline configuration)
	Battery Life (Moisture Sensor)	At least 180 days
	Operating Temperature Range	10°C to 40°C
	PCB Expansion Support	Minimum 1 additional sensor or actuator port
	System Footprint (Single Unit)	No larger than 6" × 6" × 8"

System Demonstration Overview:

To verify the integrated performance and engineering integrity of the GreenSync smart gardening system, our live demonstration will simulate a complete automated plant care cycle. This demonstration will emphasize seamless interaction between environmental inputs and system outputs, validating both core functionality and compliance with our engineering requirements. The system consists of three major subsystems: the **Watering System**, **Lighting System**, and **Camera Monitoring Module**, each rigorously tested and implemented using embedded microcontroller logic and real-time sensor feedback.

Watering System

The Watering System is powered by a capacitive soil moisture sensor calibrated to recognize thresholds corresponding to healthy hydration levels across common indoor soil types. During the demonstration, simulated dry conditions will trigger automatic watering within a **Watering Activation Delay** of **less than 10 seconds**, confirming that our system reacts quickly enough to prevent plant stress. The **Soil Moisture Polling Rate**, set to a minimum of **once per minute**, ensures that the system can detect changes in real-time

without excessive power consumption. Water delivery will be managed using a mini peristaltic pump and solenoid valve, controlled through GPIO-driven relays, allowing users to visually confirm both sensor input and actuator response.

Lighting System

To simulate daily light cycles and low ambient light scenarios, the demonstration will include manual occlusion of the VEML7700 ambient light sensor. The system will respond automatically by activating the grow light, with a **Grow Light Response Time of under 5 seconds** from detection to illumination. This verifies both sensor responsiveness and software control logic. The lighting subsystem is designed for energy efficiency, drawing **less than 50 Watts** total power, and remains suitable for small home gardens or desktop horticulture. The lighting schedule and override conditions can be manually configured through the mobile UI, allowing real-time testing of both automatic and user-triggered control.

Camera Monitoring Module

The ESP32-CAM module will demonstrate autonomous image capture with onboard timestamping and local storage. A captured image will be displayed on-screen or retrieved via serial or Wi-Fi, showing resolution at or above **640×480 pixels**. The timestamp, derived from either RTC or NTP synchronization (depending on Wi-Fi availability), will maintain accuracy **within ±1 minute** of the scheduled snapshot time, validating that the system's internal clock and scheduling logic are properly synchronized. This module supports long-term progress tracking and remote plant monitoring, with data saved to a microSD card or optionally sent to a cloud service.

This comprehensive demonstration not only verifies each subsystem's intended behavior, but also affirms that the GreenSync prototype meets the performance criteria outlined in our formal **Engineering Requirements Table**. It showcases the practical viability of the system in real-world usage, reinforcing our design's readiness for scalable deployment or future expansion.

2.7 House of Quality

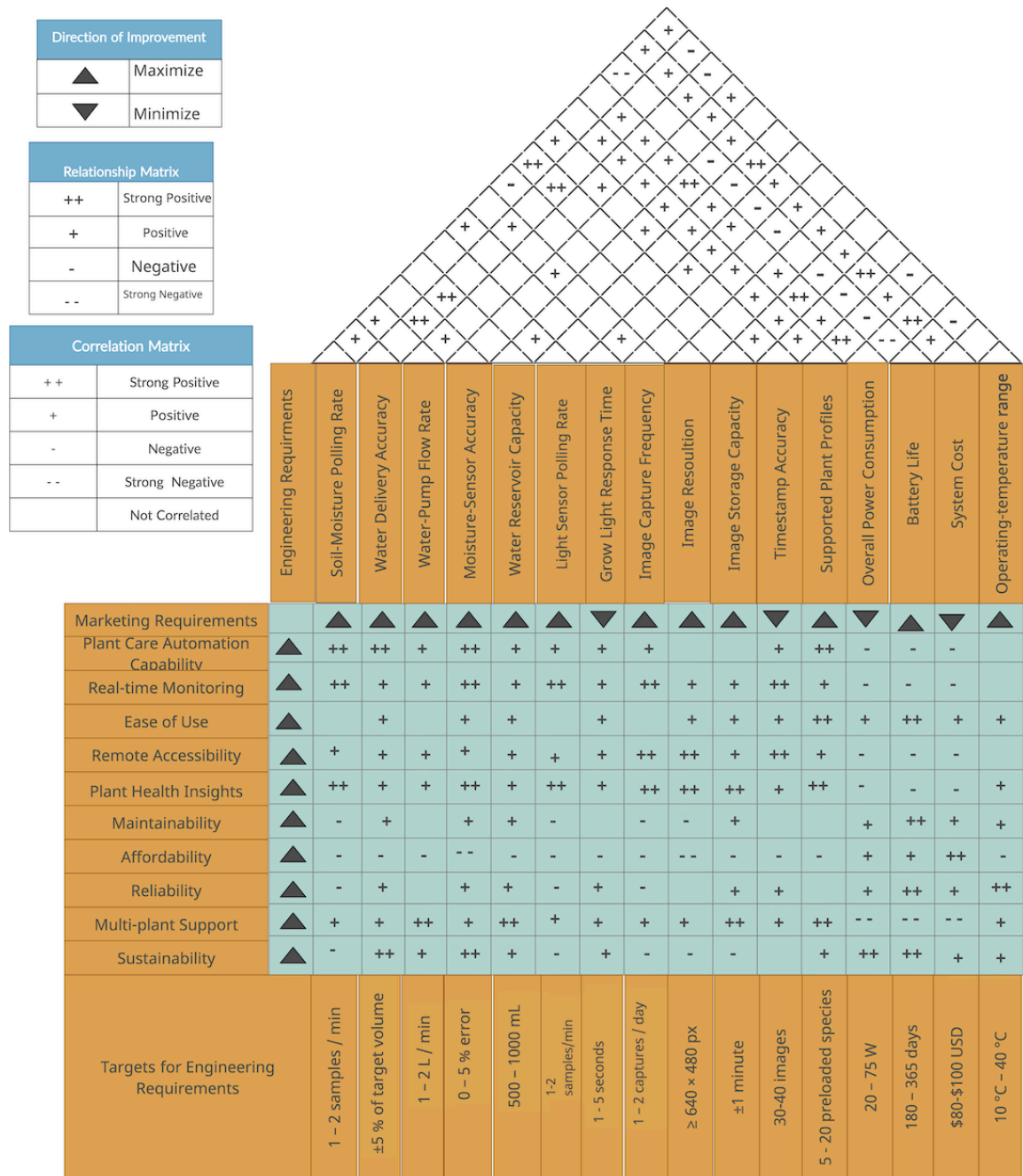


Figure 1: House of Quality

2.8 Hardware

The hardware implementation is structured around three primary subsystems: environmental sensing, plant care delivery, and growth monitoring. Each subsystem is composed of key components selected to achieve precise functionality and reliable integration.

Environmental Sensing Subsystem: This includes the soil moisture sensor and the light sensor. The soil moisture sensor is embedded in the plant's soil and is responsible for collecting data on current hydration levels, which are then transmitted to the microcontroller. The light sensor, positioned to detect ambient conditions, captures sunlight exposure levels. Together, these components allow the system to continuously monitor plant care variables and determine whether intervention is needed.

Plant Care Delivery Subsystem: This subsystem handles active plant maintenance tasks. It includes a water pump or solenoid valve system that is triggered based on sensor thresholds or time logic. A grow light module may also be included for indoor environments with insufficient natural light. These elements are connected to a custom PCB, which organizes power distribution and circuit integration. The microcontroller manages all actuator commands, ensuring that water and light are delivered in the correct amount and at the right times.

Growth Monitoring Subsystem: To add a visual element to plant progress tracking, the system integrates a compact camera module. This module captures daily images that are logged with a timestamp for future time-lapse generation. The camera's operation is controlled by the central microcontroller and configured to function at low power to accommodate continuous deployment.

Additional Considerations: The system includes a power management unit that supports either wall outlet DC input or, in extended applications, solar panel integration. All sensors, actuators, and communication modules connect through the custom-designed PCB, which also supports power regulation and expansion ports for future scalability.

While the database and user interface reside on the software side, the microcontroller and PCB enable the bridge between hardware readings and digital logging. This ensures sensor data is routed to storage and the user interface seamlessly, maintaining synchronization between physical measurements and software feedback.

2.8.1 Hardware Diagram

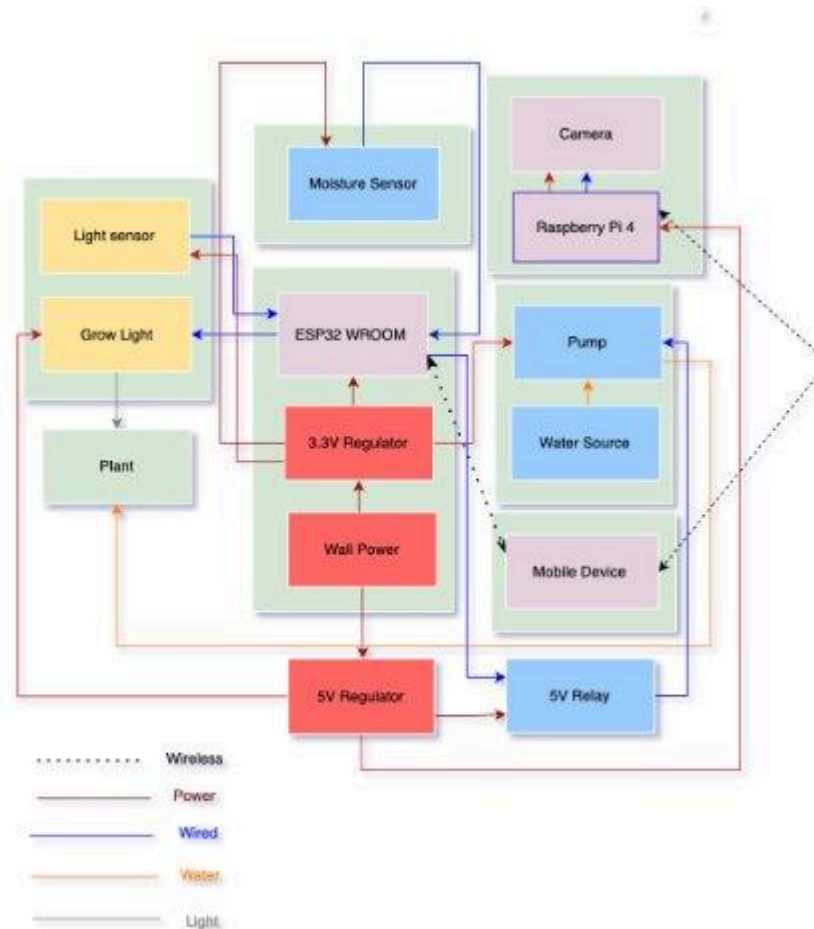


Figure 2: Hardware Diagram

2.9 Software

The software architecture of the smart plant care system is responsible for processing sensor data, executing logic for automated actions, managing user interaction, and maintaining data logs. It is divided into several core modules corresponding to the system's primary functions: sensing and data processing, care logic and actuation, image logging, and user interface management.

Environmental Data Processing:

Sensor readings from the soil moisture probe and light sensor are sampled at predefined intervals by the microcontroller's firmware. The raw analog or digital values are filtered and normalized for consistency. Threshold-based

comparisons are then applied to determine if the soil is too dry or if the light level is inadequate. These results are stored locally in memory and periodically written to a database for historical tracking.

Automated Care Logic:

Based on the processed data, the software determines whether to activate the water pump and/or grow light. Watering logic can operate on two modes: threshold-based (triggered when soil is dry) or time-based (watering at set intervals). Similarly, the light module can be toggled if natural light falls below a configured lux threshold. Logic routines also include cooldown timers to avoid repeated or excessive actions. These automated decisions are coordinated by a state machine programmed on the microcontroller.

Image Capture and Logging:

The time-lapse feature is managed by a scheduled software task that triggers the camera once daily. Each image is stamped with a date and time, then saved to a local or cloud-connected storage directory. This enables long-term visual analysis of plant growth, which may be displayed in the user interface or exported externally. Storage management includes naming conventions and retention policies to avoid overflow.

User Interface and Database Integration:

All sensor readings, watering events, light activation logs, and image data are synchronized with a centralized database. This allows the mobile application to retrieve and display real-time and historical plant data. The UI provides users with tools to configure watering and lighting thresholds, view graphs of moisture/sunlight trends, browse time-lapse photos, and manually override system actions. Notifications and alerts are also generated here, informing users of critical plant conditions (e.g., low moisture or excessive heat).

Communication and Expandability:

The software includes serial, Bluetooth, or Wi-Fi communication protocols, depending on final platform selection. These channels support communication between the microcontroller and mobile application, as well as future modules such as multi-plant expansions. The codebase is modular to support scalability and eventual updates such as additional plant profiles or sensor types.

2.9.1 Software Diagram

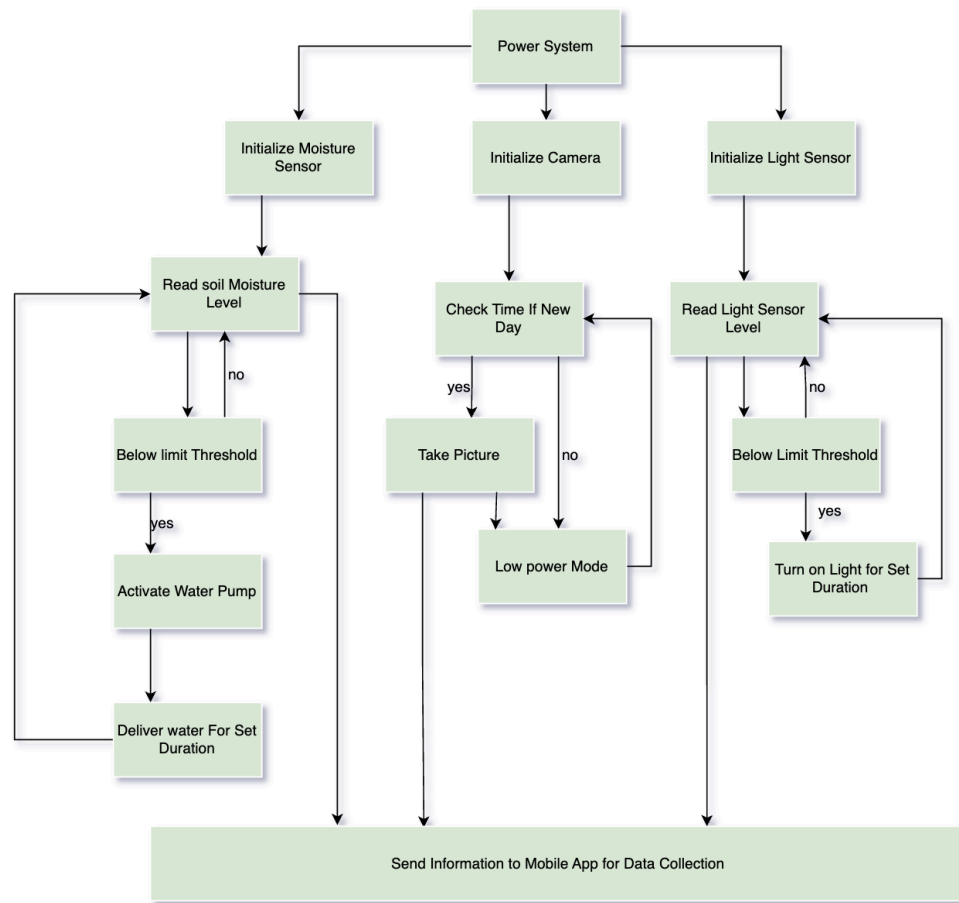


Figure 3: Software Diagram

Chapter 3 - Research

3.0 Chapter Breakdown

Before diving into the actual build of GreenSync, it's essential to outline the research that guided every major decision in our design process. This chapter serves as the foundation of our system, detailing the alternative solutions we explored, the reasoning behind each decision, and the final choices that shaped the GreenSync prototype. Since our goal was to create an automated smart garden system that delivers real-world reliability, we needed to look beyond picking parts at random—we had to compare entire technologies,

weigh their strengths and weaknesses, and then select components that aligned with our goals for efficiency, scalability, and ease of integration.

To keep everything organized and clear, this chapter follows a structured approach for every major subsystem. For each category, we begin by examining the **technology level**, looking at different methods or design strategies available in the field. For example, in the watering system section, we compare options like drip irrigation versus pump-based delivery, while in the lighting section, we analyze various grow light technologies. After discussing these alternatives, we summarize the findings in a **technology comparison table** for quick reference.

From there, we transition into **specific hardware components**, identifying real-world parts that implement those technologies. Each component comparison is also followed by a summary table that highlights key parameters such as cost, power requirements, ease of integration, and performance. Finally, we conclude each section with a **selection paragraph**, explaining why we chose the final component for our system based on the research, trade-offs, and project constraints.

This process is repeated across all major subsystems of GreenSync, starting with **processing and control hardware** (microcontroller and power systems), followed by **watering and moisture sensing systems, lighting systems, camera integration, communication and sensor integration methods**, and finally the **software and interface layer**. We also include considerations for environmental factors, such as enclosure design and soil type, to ensure the system is practical and adaptable for real-world conditions.

By presenting our research in this structured format, we demonstrate that GreenSync isn't just a random collection of parts—it's the result of deliberate design thinking backed by technical analysis and informed decisions. This ensures that the final system is not only functional but also optimized for reliability, energy efficiency, and future scalability.

3.1 Processing, Control & Power

3.1.1 Processing Unit Technologies

When designing an automated gardening system like GreenSync, the choice of processing unit plays a critical role in balancing performance, power

efficiency, cost, and ease of integration. This component acts as the system's "brain," coordinating sensor inputs, executing control logic, and handling communication with user interfaces. Embedded systems offer a variety of architectures, each with unique advantages and trade-offs. Below, we compare five major processing unit technologies commonly used in IoT and automation applications.

1. Microcontrollers (MCUs)

Microcontrollers are self-contained processing units that integrate a CPU, memory, and essential peripherals on a single chip. Examples include the ESP32, STM32, and Arduino boards. MCUs are optimized for real-time control, low power consumption, and straightforward interfacing with sensors and actuators. Their simplicity makes them ideal for systems like GreenSync, where tasks include monitoring soil moisture, controlling pumps, and toggling grow lights. Most MCUs feature built-in ADCs, PWM controllers, and wireless connectivity options, further reducing hardware complexity. However, they are limited in computational performance compared to higher-end processors, which restricts advanced features like machine learning or video analytics.[1]

2. Microprocessors (MPUs)

Microprocessors differ from MCUs in that they typically require external components such as RAM, storage, and peripheral interfaces to function. MPUs like those used in Raspberry Pi boards run full operating systems (e.g., Linux), making them suitable for complex tasks such as image processing, data visualization, and cloud integration.[2] For GreenSync, an MPU could enable advanced features like real-time image analysis or AI-driven plant diagnostics. However, this comes at a cost—MPUs are significantly more power-hungry, physically larger, and require more complex software environments. These characteristics make them less practical for a compact, energy-conscious system intended for everyday users.

3. System-on-Chip (SoC)

A System-on-Chip combines the features of an MPU with integrated wireless connectivity and peripheral control, offering a hybrid between an MCU and a small computer. SoCs like the ESP32 (used in GreenSync) are specifically designed for IoT applications, incorporating dual-core processing, built-in Wi-Fi/Bluetooth, and moderate computational capabilities in a compact footprint.[3] This architecture minimizes external components while enabling wireless communication and multitasking. SoCs strike an ideal balance for

projects like GreenSync, as they support both control functions and basic communication without requiring a full operating system. Their only limitation is scalability for computationally heavy tasks like deep learning or high-resolution video processing.

4. Field-Programmable Gate Arrays (FPGAs)

FPGAs represent the most customizable approach to embedded processing, allowing hardware logic to be programmed and reconfigured for specialized tasks. Their ability to execute operations in parallel makes them exceptionally fast for real-time signal processing and AI inference. [4] However, FPGAs are costly, power-intensive, and require advanced programming skills using hardware description languages like VHDL or Verilog. While FPGAs excel in high-performance applications such as industrial automation and robotics, their complexity far exceeds the needs of a consumer-oriented smart gardening system. Implementing GreenSync on FPGA hardware would add unnecessary expense and development time without practical benefits for its intended feature set.

5. Central Processing Units (CPUs)

CPUs, in the context of general-purpose computing, are the backbone of personal computers and servers. They offer unparalleled flexibility and raw computational power but rely heavily on external memory, storage, and power systems. While CPUs can theoretically control a smart garden system, doing so would require a full computer platform, making it inefficient and impractical. High energy consumption, heat generation, and cost render CPUs unsuitable for embedded designs where simplicity and portability are paramount. Although CPUs remain essential in backend systems or for running complex analytics, they are not viable as the main processing unit for GreenSync's compact architecture. [5]

Table 2: Processing Unit Technologies

TECHNOLOGY	PERFORMANCE & USE CASE	POWER CONSUMPTION	INTEGRATION NEEDS	IDEAL FOR	NOT IDEAL FOR
Microcontroller (MCU)	Moderate (real-time control) Tasks like moisture sensing	Very Low	Minimal (all-in-one chip)	Sensor control, pump toggling	Advanced computing, ML or video processing
Microprocessor (MPU)	High (OS, image processing) Linux-capable	High	Requires RAM, Flash & OS setup	Complex features, AI, cloud access	Compact or low-power embedded systems
System-on-Chip (SoC)	Moderate-High (dual-core, Wi-Fi/BLE on board)	Moderate	Built-in wireless & peripherals	IoT applications, multitasking	Heavy AI/video tasks with high memory demand
FPGA	Very High (custom logic, parallel execution)	High	Complex dev tools, HDL programming	Parallel processing, high-speed control	Consumer IoT products due to cost/complexity
CPU	Extremely High (PC/server-grade processing)	Very High	Needs full computer system	Backend analytics, advanced compute	On-device control, portability

Selection:

The microcontroller (MCU) was selected as the core processing unit for GreenSync due to its superior balance of simplicity, efficiency, and functionality for an IoT-based gardening system. MCUs, such as the ESP32, provide integrated peripherals like ADCs, PWM controllers, and built-in wireless communication (Wi-Fi/BLE), reducing external hardware requirements and overall system complexity. Their compact size and low power consumption make them ideal for continuous operation in resource-constrained environments without generating excess heat or draining energy. Additionally, MCUs are cost-effective compared to MPUs, FPGAs, or CPUs, ensuring affordability for end-users while delivering sufficient computational capability for GreenSync's primary functions—sensor monitoring, pump and light control, and data transmission. This combination of affordability, integrated features, and energy efficiency makes the MCU the most practical and scalable choice for the system's control architecture.

3.1.1.1 Microcontroller Units

In any smart system like GreenSync, the microcontroller serves as the brain of the operation, taking in sensor data, making decisions, and sending commands to actuators like pumps or lights. Choosing the right MCU is critical for balancing performance, connectivity, cost, and power consumption. For GreenSync, we ultimately selected a **dual-microcontroller setup**: the **ESP32** to manage sensors, water, and lighting, and the **Raspberry Pi 4 Model B** to handle higher resolution camera streaming. This section reviews several popular microcontrollers we considered and explains why the ESP32 and Raspberry Pi 4 were chosen.

1. Arduino Uno (ATmega328P)

The Arduino Uno is a reliable entry-level MCU with broad community support and a simple development environment. It offers 14 digital I/O pins, 6 analog inputs, and a 16 MHz clock speed. While useful for prototyping and basic automation, it lacks built-in wireless capabilities and has limited processing power, making it unsuitable for GreenSync's wireless IoT and multitasking requirements.

2. ESP8266

The ESP8266 is a cost-effective IoT solution with built-in Wi-Fi and adequate performance for lightweight applications. Running at 80–160 MHz, it is significantly stronger than the Arduino Uno. However, its limited GPIO count, single ADC channel, and lack of Bluetooth support reduce flexibility when managing multiple sensors and actuators. For GreenSync, this made the ESP8266 less ideal compared to more robust options.

3. STM32 "Blue Pill" (STM32F103C8T6)

The STM32 Blue Pill offers solid performance with a 72 MHz ARM Cortex-M3 core, 64 KB flash, and 20 KB SRAM. It supports a wide variety of communication protocols and has strong analog capabilities. However, it lacks built-in wireless connectivity and requires a more complex development setup, increasing system overhead compared to boards with integrated Wi-Fi and Bluetooth.

4. Raspberry Pi 4 Model B

The Raspberry Pi 4 is a powerful single-board computer rather than a traditional MCU. It features a quad-core Cortex-A72 CPU up to 1.5 GHz, up to 8 GB RAM, and strong multimedia support. Critically, it supports high-resolution camera streaming and advanced networking, which makes it ideal for GreenSync's live imaging and video tasks. However, it consumes more power and is better suited for higher-level functions rather than low-power real-time control.

5. ESP32

The ESP32 is a dual-core 32-bit MCU with integrated Wi-Fi and Bluetooth. Running up to 240 MHz with 520 KB SRAM, it supports multiple ADC channels, PWM outputs, capacitive touch inputs, and low-power modes. It is

highly efficient for sensor data collection, environmental control loops, and wireless communication. Additionally, it pairs seamlessly with the Raspberry Pi 4 to offload high-performance tasks while retaining cost-effective embedded control.

Table 3: Microprocessor Units

MCU	CORE & SPEED	WIRELESS CONNECTIVITY	GPIO & ADC	EASE OF DEVELOPMENT	KEY PROS & CONS
Arduino Uno (ATmega328P)	8-bit @ 16 MHz	None	14 GPIO, 6 PWM, 6 ADC	Very High (beginner-friendly, Arduino IDE)	Great community support, but no Wi-Fi; limited speed and RAM
Raspberry Pi Pico (RP2040)	Dual Cortex-M0+ @ 133 MHz	None (external module needed for Wi-Fi/BLE)	Rich I/O (26 GPIO multiple I2C/SPI)	High (Python, C/C++)	Great performance for price, but no native wireless adds complexity
ESP8266	32-bit @ 80–160 MHz (Tensilica L106)	Wi-Fi only	Fewer GPIO, 1 ADC	Medium (Arduino & ESP-IDF)	Good for simple IoT, but lacks Bluetooth and limited multitasking
STM32 Blue Pill (STM32F103C8T6)	32-bit @ 72 MHz (Cortex-M3)	None	37 GPIO, 10 ADC channels	Medium-Low (STM32CubeIDE, more complex setup)	Powerful, flexible, but no built-in wireless and higher dev complexity
ESP32	Dual-core 32-bit @ up to 240 MHz	Wi-Fi + Bluetooth	Many GPIO, ADCs, PWM, touch, etc.	High (Arduino, ESP-IDF, camera & sensor support)	All-in-one solution, fast, wireless-ready, multitasking, great for GreenSync

Selection: For GreenSync, we selected a dual-board architecture using both the ESP32 and the Raspberry Pi 4 Model B. The ESP32 was chosen to manage the soil moisture sensor, ambient light sensor, water pump, and grow light due to its low power consumption, integrated Wi-Fi and Bluetooth, and strong support for real-time control tasks. However, the ESP32 alone could not handle live high-resolution camera streaming, which is a key feature of our system. To address this, we incorporated the Raspberry Pi 4 Model B, which provides significantly greater processing power, memory, and multimedia capability, making it ideal for image capture, video streaming, and future computer vision features. Together, this pairing allows the ESP32 to efficiently handle low-level sensor control and actuation, while the Raspberry Pi 4 manages advanced imaging and networking tasks. This combination balances efficiency, cost, and functionality, ensuring GreenSync meets both its current requirements and future scalability goals.

3.1.2 🕒 Clock Source Options

The clock source in an embedded system determines the timing accuracy of all operations, from sensor polling to communication protocols. Selecting an appropriate clock source is essential for balancing precision, cost, and power efficiency. Below are five common options considered for GreenSync:

1. Crystal (XTAL)

Crystals are highly accurate frequency references made from quartz, providing stable oscillations over a wide temperature range. They're ideal for applications requiring precise timing, such as communication protocols or real-time data logging. However, they are slightly more expensive and require careful PCB layout [11].

2. Crystal Oscillator Module

This option integrates the quartz crystal with an oscillator circuit into a single module, simplifying design and improving stability. It reduces the need for external passive components and offers excellent performance, but it comes at a higher cost and requires more board space [12].

3. Ceramic Resonator

Ceramic resonators offer moderate accuracy at a lower cost compared to quartz crystals. They are compact and easier to integrate, making them common in consumer electronics. However, their frequency tolerance is less precise, which can impact systems relying on strict timing [13].

4. Integrated Silicon Oscillator

Many modern MCUs include a silicon-based oscillator that provides a decent balance between accuracy and convenience. These oscillators are embedded within the chip, saving space and reducing cost, but they are less precise than quartz solutions and may drift over time [14].

5. RC Oscillator (Internal)

An internal RC (resistor-capacitor) oscillator is the simplest and most cost-effective solution. It requires no external components and starts up quickly, making it a popular choice for basic timing needs. However, it is the least accurate option and can vary with temperature and supply voltage, making it suitable only for non-critical timing tasks [15].

Table 4: Clock Source Options

CLOCK SOURCE	ACCURACY	COST & COMPLEXITY	PCB IMPACT	SUITABILITY FOR GREENSYNC
Crystal (XTAL)	High ($\pm 20\text{--}50$ ppm)	Medium (needs layout care)	Medium	Great for precise timing, e.g. UART, logs
Oscillator Module	Very High (± 10 ppm)	High (self-contained)	High (larger size)	Best for precision-critical systems
Ceramic Resonator	Medium	Low	Low	Acceptable for less timing-critical tasks
Silicon Oscillator (Int.)	Medium-Low	Very Low (on-chip)	None	Good for basic systems, moderate accuracy
RC Oscillator (Internal)	Low	None (built-in)	None	Only for non-critical or temporary timing

Selection:

GreenSync utilizes the internal RC oscillator primarily because it offers a cost-effective and space-efficient solution without requiring additional external components. This simplifies the PCB design, reduces overall system cost, and speeds up development. While the RC oscillator provides lower accuracy compared to quartz-based options, its performance is sufficient for GreenSync’s control tasks, which do not rely on high-precision timing. Additionally, the quick startup time and low power draw make it ideal for an IoT device that prioritizes energy efficiency and simplicity over strict timing requirements.

3.2 Water Delivery & Moisture System

3.2.1 Water Delivery Methods

When designing an automated plant care solution like GreenSync, one of the most critical systems to consider is how water will be delivered to the plants in a controlled, efficient, and scalable manner. While there are many ways to manually water plants, automation requires systems that can function reliably without user intervention. The delivery mechanism must balance consistency, precision, and simplicity, all while being compatible with the project’s microcontroller and sensor layout. Below are five different water delivery systems that have been considered during the research and development phase.

1. Timed Drip Irrigation System

A timed drip irrigation system is a widely used method in both home gardening and commercial agriculture. This approach involves a series of

tubing lines, each equipped with tiny emitters or drippers positioned at the base of the plants. The water is released slowly over time, typically regulated by a fixed schedule or timer. Some advanced systems pair the timer with environmental sensors, but many operate purely on a fixed interval basis. This system is praised for its water efficiency, as it minimizes runoff and evaporation.[25] However, it requires careful calibration to avoid overwatering or underwatering if environmental changes occur. For GreenSync, the timed drip system is considered reliable, but it lacks the real-time adaptability that a sensor-triggered system offers.

2. Moisture-Sensor Triggered Pump System

This system forms the backbone of GreenSync's current design. It operates by constantly monitoring the soil's moisture level using a capacitive or resistive soil sensor. Once the moisture reading falls below a predefined threshold, the microcontroller activates a miniature DC water pump to deliver a precise amount of water directly into the plant's soil. This creates a responsive system that adjusts to real-world conditions rather than fixed assumptions. The biggest advantage of this approach is its flexibility, watering is only triggered when necessary, reducing waste and improving plant health.[26] This system can be easily scaled for multiple plants by assigning individual sensor and pump pairs to each module. It also integrates cleanly with the project's ESP32 microcontroller, offering direct control and logging of watering events.

3. Gravity-Fed Reservoir with Valve Control

A gravity-fed water system uses basic physics to its advantage. By placing a water reservoir at an elevation above the plant, water can be released without a pump. Instead, an electronically controlled solenoid valve governs the flow of water. When the microcontroller receives a signal that the soil is dry, the valve opens temporarily to allow water to drain into the soil using gravity. This system is especially useful in low-power applications where using a water pump would consume too much energy. It is also a relatively quiet system compared to pump-based solutions. [27] However, it does require thoughtful physical arrangement, such as securely mounting the reservoir above the plant level. Despite this drawback, it remains an energy-efficient and elegant solution worthy of consideration for future low-power versions of GreenSync.

4. Capillary Action (Wicking System)

Capillary watering systems, often called wicking systems, use a passive technique for hydration. A wick typically made of absorbent fabric, string, or rope connects a water reservoir to the plant's soil. As the soil dries out, it

naturally pulls water up through the wick via capillary action. This method has the benefit of requiring no sensors, no microcontroller input, and no electronic components. [28] It's extremely power-efficient and simple, making it ideal for small-scale or DIY gardening setups. However, this simplicity also means it lacks precision. The rate of water transfer is heavily dependent on environmental conditions like humidity and soil type, and it can become inconsistent over time. For GreenSync, this system offers inspiration for low-tech scenarios, but its lack of data integration and variability make it unsuitable for the goals of automation and monitoring.

5. Mist or Fog Delivery System

A more experimental option is the use of mist or fog-based water delivery. In this setup, water is atomized using either specialized nozzles or ultrasonic mist makers, delivering moisture as a fine aerosol to the plant environment. These systems are often found in high-end greenhouses or hydroponic facilities where humidity and leaf hydration need to be closely managed. Mist systems can operate continuously or be triggered by timers and humidity sensors, allowing for precise environmental control. [29] While this approach is highly effective in controlled environments, it adds complexity and cost. Components such as misting nozzles, high-pressure pumps, or ultrasonic transducers can be difficult to integrate with compact embedded systems. For the GreenSync system, misting remains an intriguing avenue for future development, but its current implementation favors simpler, soil-directed watering methods.

Table 5: Water Delivery Methods

METHOD	ACTIVATION TYPE	HARDWARE COMPLEXITY	POWER CONSUMPTION	PRECISION & CONTROL	SUITABILITY FOR GREENSYNC
Timed Drip Irrigation	Timer-based	Medium (tubing + emitters)	Low to Medium	Moderate	Reliable, but lacks real-time responsiveness
Sensor-Triggered Pump System	Moisture sensor	Medium (sensor + DC pump)	Medium	High	Core of current GreenSync design
Gravity-Fed Reservoir + Valve	Moisture sensor	Medium (valve + elevation)	Very Low	Moderate to High	Energy-efficient, ideal for low-power future
Capillary (Wicking System)	Passive	Very Low	None	Low	Simple backup, not suitable for automation
Mist/Fog Delivery System	Timer or humidity sensor	High (mist maker, nozzles)	Medium to High	Very High	Future upgrade, not ideal for current setup

Selection:

GreenSync employs a moisture-sensor-triggered pump because it provides precise, real-time control over soil hydration, ensuring plants receive the

exact amount of water they need. Unlike timed irrigation, which operates on a fixed schedule regardless of soil conditions, this method leverages sensor feedback to activate the pump only when moisture falls below a defined threshold. This significantly improves water efficiency, reduces the risk of overwatering or underwatering, and supports healthier plant growth. While the system introduces additional components—such as sensors and control logic—the benefits of automation, resource conservation, and adaptability make this approach the most effective and scalable solution for GreenSync’s smart gardening objectives.

3.2.2 Water Pump

In automated garden systems like GreenSync, the water pump plays a critical role in transporting water from a reservoir to the plant. Since the system is designed for compact environments like indoor gardens, balconies, or smart pots, the pump must be efficient, low-power, and easily controllable by a microcontroller such as the ESP32. There are several categories of mini pumps used in these kinds of systems, each offering different advantages in terms of flow rate, installation style, maintenance, and control. Below is an exploration of five common mini pump types suitable for small-scale irrigation or hydroponic setups.

1. Submersible DC Pump

One of the most commonly used pumps in small-scale watering systems is the submersible DC pump. These pumps, such as the JT-180A or RS-360 models, are designed to be fully submerged in water and typically run on 3V to 12V DC power. Their popularity comes from their compact size, low noise operation, and reliable performance. They are easy to install, just drop them into a reservoir and they’re readily compatible with microcontrollers using relay modules or transistor switches. Submersible pumps are ideal for low-lift watering tasks and work well in systems with basic tubing and gravity-fed layouts. [41]

2. Peristaltic Pump

Peristaltic pumps offer a more precise way to control water flow. They operate by rotating rollers that compress flexible tubing, pushing water forward without allowing it to backflow. This mechanism ensures that only the tubing comes into contact with the water, which makes them great for systems that deliver nutrients or need to avoid contamination. These pumps are often used in dosing systems or nutrient delivery modules in hydroponics. Although they

tend to have slower flow rates and can be more expensive than submersible pumps, they offer unmatched precision and cleanliness. [42]

3. Diaphragm Pump

Miniature diaphragm pumps are another strong option. These pumps use a flexible membrane to draw in and expel water, creating suction and pressure through rapid back-and-forth movement. Many models can self-prime, meaning they don't need to be submerged or pre-filled with water to start working. They are also capable of handling slightly dirty or debris-filled water, which gives them an edge in less controlled environments. Diaphragm pumps are slightly bulkier than submersible ones but are durable and can push water over longer distances. [43]

4. Centrifugal Pump

Centrifugal pumps operate using a spinning impeller to generate water flow. Though more commonly used in larger applications, there are brushless mini versions that can be integrated into compact systems. These pumps are typically not self-priming, which means they need to be submerged or filled before use. However, they excel at moving large volumes of water at low pressure, making them ideal for distributing water horizontally across plant beds or multiple zones. They are efficient in continuous-use situations but can require a bit more planning during installation. [44]

5. Vibration Pump

Vibration pumps, also called electromagnetic pumps, move water using high-frequency oscillation. They are often used in ultra-compact gadgets and are favored for their small size and simplicity. These pumps generally offer lower flow rates compared to others and are better suited for systems where minimal water delivery is required. Their main advantage lies in energy efficiency and noise reduction, but their limited pumping power makes them less ideal for multi-plant or large-area irrigation systems. [45]

Table 6: Water Pumps

PUMP TYPE	HOW IT WORKS	AVG. FLOW RATE	BENEFITS	DRAWBACKS	COST
Submersible DC Pump	Uses a small DC motor with an impeller; fully submerged in water	80–240 L/h (depending on size)	Compact; Quiet; Easy to install; Compatible with relays & MOSFETs	Requires submersion; Not self-priming; Limited head height	Low (\$3–\$10)
Peristaltic Pump	Rotating rollers compress flexible tubing to push water	20–100 mL/min (very low)	Highly precise dosing; Prevents contamination; Only tubing contacts liquid	Slow flow; Higher cost; Tubing wear over time	Moderate (\$8–\$20)
Diaphragm Pump	Flexible membrane oscillates to create suction & pressure	1–3 L/min	Self-priming; Handles slight debris; Can pump over long distances	Bulkier; Louder than submersible; Moderate power draw	Moderate (\$10–\$25)
Centrifugal Pump	Spinning impeller creates flow; typically submerged	300–800 L/h (mini versions)	High volume; Efficient for continuous operation; Low maintenance	Not self-priming; Requires submersion or priming; Less precise control	Moderate (\$10–\$30)
Vibration Pump	Electromagnetic oscillation moves water in pulses	100–200 mL/min	Ultra-compact; Low power; Quiet	Low pressure; Low flow; Limited to small systems	Low (\$4–\$8)

Selection:

Peristaltic pumps are an excellent option for applications requiring precise and contamination-free water delivery. Their roller-based mechanism ensures that only the tubing contacts the liquid, making them ideal for nutrient dosing or chemical-sensitive systems. This closed-flow design prevents backflow and allows accurate control of small fluid volumes, which is essential for hydroponics or fertilizer delivery. While they offer slower flow rates compared to submersible pumps and come at a higher cost, their precision and cleanliness make them a valuable addition to advanced smart gardening systems like GreenSync, particularly for optional nutrient management features.

Update:

For the GreenSync system, a **submersible DC water pump** was selected due to its simplicity, low power consumption, and reliable performance in compact indoor environments. This style of pump—as shown in the hardware kit used during prototyping—can be fully placed inside the water reservoir, eliminating the need for priming and reducing installation complexity. Submersible pumps provide a steady and consistent flow rate that is well-suited for small soil-based plants, and they integrate easily with microcontroller control through a standard 5V relay module. Their lightweight construction, quiet operation, and compatibility with basic vinyl tubing made them the most practical choice for GreenSync’s automated irrigation module. Although they do not offer the high precision of peristaltic pumps or the

long-distance performance of diaphragm pumps, submersible pumps offer the best balance of cost, efficiency, and ease of use for a home-scale smart garden system.

Component Pumps:

1. Adafruit Peristaltic Pump

The Adafruit Peristaltic Pump operates at 6–12V DC and delivers a flow rate of around **100 mL/min**, making it suitable for dosing small amounts of water or nutrients. Its compact size and easy integration with microcontroller-driven relays or MOSFETs make it ideal for automation. While not intended for bulk irrigation, its precision makes it a strong choice for nutrient delivery systems.[46]

2. DFRobot Gravity Peristaltic Pump

This pump runs on **12V DC** and provides a slightly higher flow rate of **0.1–0.3 L/min**, giving it more flexibility for small hydroponic systems. It includes silicone tubing for chemical resistance and is designed with durability in mind for continuous or intermittent operation. Its higher cost compared to Adafruit reflects its robust build and reliable performance in extended use scenarios. [47]

3. Kamoer Micro Peristaltic Pump

The Kamoer KFS-B is a premium option known for laboratory-grade dosing accuracy. Operating at **12V DC**, it features precise speed control, making it suitable for research and high-end smart agriculture applications. However, its cost is significantly higher, and its advanced capabilities may be overkill for small consumer systems like GreenSync. [48]

Table 7: Pump Components

PUMP MODEL	OPERATING VOLTAGE	FLOW RATE	HOW IT WORKS	ADVANTAGES	DRAWBACKS	COST RANGE
Adafruit Peristaltic Pump (ID: 1150)	6–12V DC	~100 mL/min	Rotating rollers compress tubing to move liquid	Compact; Easy to control with microcontrollers; Affordable; Ideal for nutrient dosing	Low flow rate (not for bulk watering); Tubing wear over time	\$25–\$30
DFRobot Gravity Peristaltic Pump (FIT0503)	12V DC	0.1–0.3 L/min	Same peristaltic principle with silicone tubing for chemical resistance	Higher flow rate than Adafruit; Durable build; Reliable for intermittent or continuous use	Slightly bulkier; Higher cost than Adafruit	\$35–\$50
Kamoer Micro Peristaltic Pump (KFS-B)	12V DC	Highly precise, adjustable speed (~1–150 mL/min)	Advanced roller mechanism with precision speed control	Laboratory-grade accuracy; Excellent for research; Long lifespan	Very high cost; Overkill for small-scale consumer systems	\$80–\$150

Selection:

The Adafruit Peristaltic Pump was selected for GreenSync’s optional nutrient dosing system because it provides an optimal balance between precision, ease of integration, and cost. Its 6–12V DC operating range and ~100 mL/min flow rate make it ideal for delivering small, controlled quantities of water or liquid fertilizer without risk of contamination, as only the tubing contacts the fluid. This design ensures clean and accurate dosing, critical for maintaining soil health in advanced gardening setups. The pump’s compact size and compatibility with microcontroller-controlled relays or MOSFETs simplify implementation, while its affordability makes it accessible for consumer applications. Compared to higher-cost options like the Kamoer KFS-B or the more powerful DFRobot model, the Adafruit pump provides sufficient functionality without unnecessary complexity or expense, making it the most practical choice for GreenSync’s nutrient delivery feature.

Update:

For the final GreenSync design, the team selected the **Adafruit 3.3V Submersible Pump**, which offered the best compatibility with our low-voltage architecture and ESP32-based control system. Unlike standard 5V or 12V pumps, this model is optimized for direct integration with 3.3V microcontroller systems, reducing the need for additional regulators or driver circuitry. Its compact size, quiet operation, and fully submersible design make it ideal for small indoor reservoirs and enclosed garden enclosures. The pump provides a stable and gentle flow rate that prevents over-watering and soil disruption, while still being strong enough to deliver consistent irrigation through standard vinyl tubing. Additionally, Adafruit’s documentation, reliability, and part consistency ensured predictable performance during both prototyping and

long-term operation. This made the 3.3V submersible pump the most efficient and integration-friendly choice for GreenSync's automated watering system.

3.2.3 Moisture Sensing Technologies

When designing a smart gardening solution like GreenSync, one of the most essential pieces of technology is the soil moisture sensor. These devices are responsible for detecting how much water is present in the soil, helping the system make decisions about irrigation. While many users might assume there's a one-size-fits-all solution, the reality is that moisture sensing technology comes in many forms—each with unique benefits, drawbacks, and levels of precision. In this section, we explore several types of soil moisture sensing technologies that are commonly used in gardening systems, from basic resistive sensors to high-end reflectometry-based devices. Understanding these options not only helps justify our design decisions, but also paints a broader picture of how complex plant care systems can truly be.

1.Capacitive Soil Moisture Sensors

Capacitive soil moisture sensors are one of the most modern and practical technologies for detecting water levels in soil. These sensors work by measuring changes in the dielectric permittivity of the soil, which correlates directly with its moisture content. Because they avoid direct electrical contact with the soil, they are less prone to corrosion compared to older resistive models, resulting in longer lifespan and more reliable long-term use. Capacitive sensors are ideal for microcontroller-based systems like GreenSync due to their low power requirements, analog or PWM output formats, and compatibility with most 3.3V or 5V logic systems. Their compact size and high precision make them the go-to choice for many DIY and commercial automated irrigation projects. [30]

2. Resistive (Single-Point) Moisture Sensors

Resistive moisture sensors function by measuring the electrical resistance between two exposed probes inserted into the soil. Water conducts electricity better than dry soil, so as moisture levels increase, the resistance decreases. While these sensors are cost-effective and simple to use, they are notorious for degrading over time due to electrochemical corrosion of the probes. This makes them less ideal for long-term installations unless frequent maintenance or replacements are factored in. Despite their drawbacks, they remain popular for educational purposes and entry-level smart garden systems because of their ease of integration and very low price point. [31]

3. Soil Water Tension Sensors (Tensiometers)

Tensiometers offer a more traditional approach by measuring the soil's water potential, or tension, which reflects how strongly water is held by soil particles. This method doesn't measure actual water content, but rather how difficult it is for plant roots to extract water. A tensiometer typically consists of a water-filled tube with a porous tip and a pressure gauge. As the soil dries, water is drawn out of the tube, creating a measurable vacuum. These sensors are commonly used in agricultural research and commercial farming. Though not always practical for compact systems like GreenSync, they offer valuable insight into plant water accessibility and are excellent candidates for discussion when analyzing precision irrigation techniques. [32]

4. Volume Water Content Sensors (TDR and FDR)

Time Domain Reflectometry (TDR) and Frequency Domain Reflectometry (FDR) sensors determine soil moisture by measuring the time or frequency shift of a reflected electromagnetic signal as it travels along a probe. These systems offer highly accurate and repeatable measurements of volumetric water content in soil. While incredibly effective, these sensors are generally more expensive and require more complex circuitry to operate, making them better suited for research institutions or high-end agricultural setups. Still, their inclusion in a research context helps demonstrate the cutting-edge capabilities available in soil science and automated irrigation. [33]

5. Gypsum Block Sensors

Gypsum block sensors represent a legacy method for tracking soil moisture levels. These sensors rely on two electrodes embedded in a block of gypsum. As soil moisture permeates the block, it alters the electrical resistance between the electrodes. Gypsum has a stable and known conductivity profile, which allows for relatively consistent readings. Though not as responsive or durable as modern alternatives, gypsum blocks are still used in some agricultural environments due to their low cost and ability to mimic the water availability conditions experienced by plants. They offer a good contrast to newer digital methods and highlight the evolution of moisture sensing technology over time. [34]

6. Moisture Sensor Arrays / Multi-Point Probes

Moisture sensor arrays use multiple sensing elements along a vertical or horizontal axis to measure moisture distribution across soil layers or plant

beds. These systems are especially valuable in large-scale gardens, greenhouses, or agricultural plots where spatial variability of soil moisture plays a critical role in plant health. While overkill for small systems like GreenSync, sensor arrays introduce the concept of scalable data acquisition. They also allow for zoned irrigation, where only specific areas receive water based on localized data. Their advanced capabilities, while more complex and expensive, provide insight into how large-scale systems handle diverse soil moisture conditions effectively. [35]

Table 8: Moisture Sensor Technologies

SENSOR TYPE	MEASUREMENT PRINCIPLE	ACCURACY / PRECISION	DURABILITY	COST & COMPLEXITY	BEST SUITED FOR
Capacitive Sensor	Dielectric permittivity	High	High (non-corrosive)	Low cost, easy integration	DIY & microcontroller projects
Resistive Sensor	Electrical resistance	Low to Moderate	Low (corrodes easily)	Very cheap, very simple	Education, hobby projects
Tensiometer	Soil water tension (suction)	Moderate	Moderate	Moderate cost, analog setup	Agricultural studies
TDR/FDR Sensor	Electromagnetic reflection	Very High	High	High cost, complex circuit	Precision irrigation research
Gypsum Block Sensor	Resistance via gypsum block	Low to Moderate	Moderate	Low cost, old tech	Legacy systems, field demos
Sensor Array / Multi-Point	Spatial moisture sampling	Varies (multi-depth)	High (varies by model)	High cost, advanced logic	Greenhouses, large plots

Selection:

GreenSync uses capacitive soil moisture sensors because they provide the best balance of accuracy, durability, and ease of integration for an automated plant care system. Unlike resistive sensors, which suffer from probe corrosion and short lifespans, capacitive sensors measure soil moisture indirectly through dielectric permittivity, avoiding direct metal-to-soil contact. This ensures long-term reliability and reduces maintenance requirements. Capacitive sensors also offer fast response times and produce stable readings, making them ideal for real-time control of irrigation systems. Their compatibility with low-voltage microcontrollers like the ESP32, combined with simple analog or PWM output formats, further streamlines system design. While slightly more expensive than basic resistive probes, their superior accuracy, robustness, and low power consumption make capacitive sensors the most practical and future-proof choice for GreenSync.

3.2.4 Capacitive Soil Moisture Sensor

Soil moisture sensing is an essential component of any automated gardening system, particularly for precision-based designs like GreenSync. Accurate moisture data ensures efficient water management, reduces waste, and promotes healthy plant growth. The choice of sensor directly impacts system reliability, maintenance requirements, and integration complexity. To determine the most suitable option for GreenSync, we analyzed several popular soil moisture sensors that represent different technologies, costs, and performance characteristics. Below are five widely used models that illustrate the range of solutions available in this domain.

1. AITRIP Capacitive Soil Moisture Sensor

The AITRIP Capacitive Soil Moisture Sensor was selected for GreenSync due to its durability, corrosion resistance, and excellent compatibility with microcontrollers like the ESP32. This sensor measures soil moisture by detecting changes in dielectric permittivity, which eliminates direct electrical contact with the soil and minimizes corrosion—a common issue in resistive sensors. It outputs an analog voltage signal that is easy to process, supports both 3.3V and 5V operation, and integrates seamlessly into automated irrigation systems. Its slim form factor allows for easy placement in potted plants or confined garden setups, and its PCB coating enhances lifespan in humid environments. [36]

2. DFRobot Capacitive Soil Moisture Sensor (SKU: SEN0193)

The DFRobot SEN0193 is an upgraded capacitive moisture sensor designed for long-term stability and outdoor durability. It features a conformal coating for enhanced moisture protection and improved signal consistency compared to standard models. Like the AITRIP sensor, it uses dielectric permittivity detection, ensuring minimal corrosion and reliable readings. However, its higher cost makes it better suited for premium or outdoor systems where additional ruggedness justifies the price difference. [37]

3. Chirp I2C Soil Moisture Sensor

The Chirp sensor is a digital alternative that uses I²C communication instead of analog output, providing stable data transmission and immunity to analog signal noise. It also measures soil temperature and includes an optional audible alert feature to indicate dry conditions. While its feature set is attractive, it increases complexity by occupying an I²C address and requiring additional software libraries. This makes it more suitable for modular or experimental setups rather than cost-sensitive systems like GreenSync. [38]

4. Vegetronix VH400 Soil Moisture Sensor

The VH400 is a high-end sensor designed for agricultural and research applications. It delivers precise volumetric water content measurements, fast response times, and excellent durability in outdoor conditions. Its waterproof design and high accuracy make it a benchmark in soil moisture sensing. However, the cost is significantly higher—often five to ten times that of basic capacitive models—rendering it impractical for consumer-focused projects like GreenSync, where affordability is a key design criterion. [39]

5. SparkFun Soil Moisture Sensor Kit (SEN-13322)

This sensor kit represents a traditional resistive sensing solution that measures soil moisture by monitoring electrical resistance between two exposed probes. While inexpensive and easy to integrate, it suffers from rapid corrosion due to electrochemical reactions in wet soil, leading to poor long-term reliability. Frequent calibration and probe replacement are necessary in continuous-use systems, which makes this option more suitable for educational projects than for automated solutions requiring minimal maintenance. [40]

Table 9: Capacitive Moisture Sensor

SENSOR	ACCURACY	DEPTH RANGE	OUTPUT	WATER CALCULATION	ADVANTAGES	LIMITATIONS
AITRIP Capacitive Sensor	High for general plant care	~3–5 cm (typical for potted plants)	Analog voltage	Indirect (based on dielectric change)	Corrosion-resistant; Low power; Easy ESP32 integration; Affordable bulk packs	Requires calibration for different soil types; Not lab-grade accuracy
DFRobot SEN0193	High, more stable than generic capacitive sensors	~3–5 cm	Analog voltage	Indirect (dielectric method)	Enhanced PCB coating; Reliable long-term use; Durable for outdoor	Higher cost than standard models
Chirp I ² C Sensor	High consistency (digital output reduces noise)	~3–6 cm	I ² C digital	Indirect (dielectric method)	Measures moisture & temperature; Noise-immune digital signal	Requires I ² C libraries; Adds bus complexity; Slightly larger size
Vegetronix VH400	Very high (research-grade)	Up to 8–10 cm	Analog voltage	Yes (calibrated for volumetric water content)	Exceptional accuracy; Waterproof; Professional-grade durability	Extremely expensive; Overkill for consumer systems
SparkFun SEN-13322	Low (susceptible to drift and corrosion)	~3–4 cm	Analog voltage	Indirect (via resistance)	Very low cost; Simple to use; Good for education	Rapid corrosion; Poor lifespan; Requires frequent calibration

Selection:

The AITRIP Capacitive Soil Moisture Sensor was chosen for GreenSync because it delivers the best combination of accuracy, durability, and affordability for a consumer-oriented smart gardening system. Unlike resistive sensors, it uses capacitive sensing to measure soil dielectric permittivity, eliminating corrosion issues and reducing long-term maintenance. Its analog voltage output integrates seamlessly with the ESP32 microcontroller without requiring complex digital protocols or additional converters, which simplifies both hardware and software design. The sensor operates reliably at 3.3V or 5V, supports compact installation, and provides stable readings suitable for automated irrigation control. Additionally, purchasing in bulk packs lowers the overall cost of multi-plant configurations, making it ideal for a scalable solution like GreenSync.

3.2.5 Release Valve

In automated irrigation systems like GreenSync, the release valve is a key component that controls the flow of water from the reservoir to the delivery tubing. By opening or closing in response to microcontroller commands, the valve ensures precise irrigation cycles and prevents unwanted water leakage or backflow when the pump is not running. Several valve technologies were evaluated during the design process to determine which type best suits GreenSync's requirements for reliability, efficiency, and integration simplicity. Below are five widely used valve technologies analyzed in this study.

1. Standard On/Off Solenoid Valve

The standard on/off solenoid valve is one of the most common solutions in irrigation systems. It uses an electromagnetic coil to lift or drop a plunger inside the valve body, allowing or blocking water flow. These valves typically operate in either normally closed (NC) or normally open (NO) configurations and are widely available in plastic or metal housings. They are reliable and affordable, making them a popular choice for residential and commercial applications. However, they consume power continuously when energized, which makes them less suitable for battery-powered setups. [49]

2. Miniature Liquid Solenoid Valve

Miniature solenoid valves are a compact variant of standard on/off valves, designed for low-flow applications such as smart pots, hydroponic systems, and small-scale irrigation setups. These valves are optimized for low-pressure systems and often operate at voltages like 6V or 12V DC, making them easy to integrate with microcontrollers via relays or MOSFET circuits. Their small

size and low power consumption make them ideal for space-constrained and energy-conscious environments like GreenSync. [50]

3. Latching (Bistable) Solenoid Valve

Latching solenoid valves differ from conventional types by requiring power only during actuation, not for maintaining their state. This means they use a short electrical pulse to switch open or closed and then hold that position without continuous energy consumption. These valves are excellent for ultra-low-power applications, especially solar-powered or battery-operated systems. However, they require polarity reversal to change states, which adds design complexity and necessitates additional components like H-bridge drivers.[51]

4. Proportional Solenoid Valve

Proportional valves provide variable water flow instead of simple on/off control. By adjusting the coil current, these valves allow precise control of water delivery, making them ideal for advanced irrigation systems or nutrient dosing setups. Despite their high level of control, proportional valves are complex and expensive, which limits their practicality for consumer-grade smart gardening systems like GreenSync. [52]

5. Motorized Ball Valve

Motorized ball valves use a small DC motor to rotate a spherical element within the valve body, providing full shut-off or partial flow control. They are highly durable and capable of handling high pressures, making them common in industrial automation, plumbing, and HVAC systems. However, they are slower to actuate, more expensive, and require higher power than solenoid valves, which makes them less suitable for compact, energy-efficient systems like GreenSync. [53]

Table 10: Release Valves

VALVE TYPE	TYPICAL VOLTAGE	FLOW CONTROL	POWER CONSUMPTION	ADVANTAGES	DRAWBACKS	COST RANGE
Standard On/Off Solenoid Valve	12V or 24V DC	Fully open or closed	Continuous when energized	Simple design; Reliable; Widely available	Higher energy usage; Bulkier; Not ideal for battery power	\$10–\$20
Miniature Liquid Solenoid Valve	6–12V DC	Fully open or closed	Low; Continuous when active	Compact; Low power; Easy MCU integration; Ideal for small systems	Limited flow capacity; Not for high-pressure systems	\$15–\$25
Latching (Bistable) Solenoid Valve	6–12V DC	Fully open or closed	Power only during switching	Excellent energy savings; Great for solar/battery setups	Requires polarity reversal; Added control complexity	\$25–\$40
Proportional Solenoid Valve	12–24V DC	Variable flow control	Moderate to high	Precise flow adjustment; Ideal for dosing systems	Expensive; Complex to control; Overkill for small gardens	\$50–\$100+
Motorized Ball Valve	12–24V DC	Fully open, closed, or partial	Higher than solenoids	Extremely durable; Handles high pressure; Good for large-scale systems	Slow actuation; More expensive; Consumes more energy	\$30–\$60

Selection:

Among the various valve technologies considered, the **Miniature Liquid Solenoid Valve** was selected for GreenSync because it aligns perfectly with the system’s compact design and low-flow irrigation requirements. Unlike motorized ball valves or proportional valves, which offer advanced control but add unnecessary complexity and cost, miniature solenoid valves provide simple on/off water control in a small form factor with minimal power consumption. While standard solenoid valves are effective for large-scale irrigation, their size and higher power draw make them less suitable for indoor or container-based gardening. Similarly, latching valves, though energy-efficient, introduce additional circuitry requirements that are unnecessary for a wall-powered system like GreenSync. The miniature liquid solenoid valve offers the ideal combination of simplicity, affordability, and compatibility with microcontroller-based control, making it the most practical choice for our release valve technology.

Update:

After evaluating several valve technologies, the GreenSync team ultimately determined that a dedicated release valve was **not required** for the final system design. Because our irrigation approach uses a small submersible pump that only activates during watering cycles, water flow can be reliably controlled through the pump itself without the need for an additional mechanical shut-off component. The system’s low-pressure environment and short tubing length further reduce the risk of backflow or passive leakage, making a solenoid or motorized valve unnecessary. Eliminating the release

valve also simplified the electronics, reduced power consumption, and minimized overall cost and system complexity. For these reasons, GreenSync operates effectively with pump-only flow control, and no standalone release valve was implemented in the final prototype.:

Components:

1. Adafruit Mini Liquid Solenoid Valve (Product ID: 997)

The Adafruit Mini Liquid Solenoid Valve is designed for compact irrigation systems where space and efficiency are critical. Operating at **6–12V DC**, it offers a flow rate of approximately **0.6 L/min at 0.02 MPa**, making it ideal for low-pressure, small-scale watering setups. Its normally closed configuration ensures water remains blocked until activated, reducing leakage risks. Integration with microcontrollers such as the ESP32 is straightforward through a MOSFET or relay driver circuit. Its compact size, low power consumption, and compatibility with standard tubing make it the preferred choice for GreenSync. [54]

2. DFRobot Gravity Mini Solenoid Valve (SKU: FIT0503)

The DFRobot Gravity Mini Solenoid Valve provides slightly higher performance for small irrigation systems. It operates at **12V DC** and delivers a flow rate of around **0.8 L/min**, making it suitable for setups that require slightly higher water throughput. Designed for compatibility with the DFRobot ecosystem, it features rugged construction for improved durability in continuous-use environments. While it offers strong performance and reliability, its slightly bulkier size and higher price point make it less optimal for GreenSync’s compact design requirements. [55]

3. U.S. Solid Mini DC Solenoid Valve (USS-MSV00008)

The U.S. Solid Mini DC Solenoid Valve represents a more industrial-grade solution. Operating at **12V DC**, it supports a flow rate of up to **1.0 L/min**, handling higher pressures and extended usage cycles without degradation. Its robust construction makes it ideal for professional or heavy-duty applications. However, its higher cost and physical size make it excessive for consumer-level systems like GreenSync, where compactness and affordability are priorities. [56]

Table 11: Valve Components

MODEL	VOLTAGE	FLOW RATE	PORT SIZE	ADVANTAGES	DRAWBACKS	COST RANGE
Adafruit Mini Liquid Solenoid Valve (997)	6–12V DC	~0.6 L/min	1/8" NPS	Compact; Low power; Easy MCU integration; Affordable	Limited to low-pressure systems	\$20–\$25
DFRobot Gravity Mini Solenoid Valve (FIT0503)	12V DC	~0.8 L/min	1/4"	Rugged; Higher flow capacity; Designed for Gravity ecosystem	Slightly bulkier; Higher price than Adafruit	\$30–\$35
U.S. Solid Mini DC Solenoid Valve (USS-MSV00008)	12V DC	~1.0 L/min	1/4"	Industrial-grade reliability; Handles higher pressure; Durable	More expensive; Overkill for small systems	\$40–\$50

Selection:

The **Adafruit Mini Liquid Solenoid Valve** was selected for GreenSync because it offers the best balance of size, cost, and compatibility for a compact indoor irrigation system. Its 6–12V DC operating range allows easy integration with the ESP32 through MOSFET or relay control, and its flow rate of approximately 0.6 L/min is sufficient for small plant watering tasks without adding unnecessary bulk or power requirements. While alternatives like the DFRobot Gravity Valve and U.S. Solid Mini Valve provide higher flow and durability for heavy-duty use, they introduce added cost and complexity without significant benefit for GreenSync’s design goals. The Adafruit valve’s compact design, affordability, and proven reliability make it the most practical choice for our system.

3.2.6 Water Reservoir

The water reservoir is a critical component of GreenSync as it serves as the main water supply for the irrigation system. Its design influences both the ease of integration with other components—such as the submersible DC pump and the release solenoid valve—and the overall functionality of the system. Factors such as capacity, material durability, opacity, and compatibility with tubing and fittings were considered during the selection process. A reservoir that is too small would require frequent refills, while one that is too large would make the system unnecessarily bulky. Below are four reservoir options evaluated for GreenSync.

1. Reliance Aqua-Tainer (Selected)

The Reliance Aqua-Tainer is a rugged, BPA-free plastic container commonly used for camping and emergency water storage. With a capacity of 4 gallons, it provides ample water for multiple irrigation cycles without frequent refills. The container features a large lid opening that easily accommodates a

submersible pump and includes a built-in spigot that can be adapted for tubing, simplifying integration with the Adafruit Mini Liquid Solenoid Valve. Its semi-opaque design reduces—but does not fully eliminate—light penetration, so some algae prevention measures may be necessary. Although slightly larger than required for a tabletop prototype, its durability and ease of adaptation make it a practical choice for long-term use and future scalability. [57]

2. Adafruit 2-Liter Reservoir with Cap

The Adafruit 2-Liter Reservoir is a compact, purpose-built solution designed for small-scale watering and dosing applications. Made from lightweight plastic, it features a screw-top cap that allows easy integration with tubing. While this reservoir is well-suited for prototypes and ultra-compact designs, its limited 2-liter (0.5-gallon) capacity means frequent refilling, which reduces convenience for extended operation. Additionally, its transparent body allows light to pass through, increasing the risk of algae growth if left uncovered. Despite these limitations, the Adafruit reservoir remains an excellent choice for educational demonstrations or projects with very constrained space. [58]

3. Active Aqua 2-Gallon Hydroponic Reservoir

The Active Aqua 2-Gallon Reservoir is a professional-grade container designed specifically for hydroponic and automated irrigation systems. Constructed from UV-resistant black plastic, it prevents light exposure and significantly reduces algae formation. Its included snap-on lid often features pre-drilled holes for tubing and wiring, streamlining integration with pumps and valves. At 2 gallons, it offers a balance between capacity and compactness, making it suitable for home and small-scale systems. The only drawbacks are its higher cost and limited availability compared to standard containers, which may deter users looking for a quick or budget-friendly option. [59]

4. Sterilite 1.5-Gallon Plastic Container

The Sterilite 1.5-Gallon Plastic Container is a common household storage solution often repurposed for irrigation systems. Its food-safe construction and lightweight design make it an accessible and affordable option. While the container is easy to modify by drilling holes for tubing, it lacks any built-in ports or sealing features, which can make the installation process more involved. Furthermore, because it is transparent, it requires an opaque cover or wrap to prevent algae growth. This container is a viable low-cost choice for

early prototypes, but its shortcomings in durability and algae prevention limit its suitability for long-term deployment. [60]

Table 12: Water Reservoir

RESERVOIR	CAPACITY	MATERIAL	OPAQUE	ADVANTAGES	DRAWBACKS	COST
Reliance Aqua-Tainer	4 gal	BPA-free plastic	Semi-opaque	Durable; Built-in spigot; Large lid for pump	Slightly large; Needs algae prevention	\$15–\$20
Adafruit 2L Reservoir	0.5 gal	Plastic	No	Compact; Easy tubing integration	Very small; Algae risk	\$8–\$10
Active Aqua 2-Gal Reservoir	2 gal	UV-resistant plastic	Yes	Opaque; Pre-drilled lid; Pro-grade	Higher cost; Less available	\$25–\$30
Sterilite 1.5-Gal Container	1.5 gal	Plastic	No	Cheap; Easy to find	Requires drilling; Algae risk	\$5–\$7

Selection:

The Reliance Aqua-Tainer was selected as GreenSync’s reservoir due to its superior durability, generous capacity, and ease of integration with the system’s components. Its large opening simplifies installation of the submersible DC pump, while the built-in spigot can be adapted for tubing connected to the Adafruit Mini Liquid Solenoid Valve. Although its size exceeds what is strictly necessary for a tabletop prototype, the Aqua-Tainer’s robustness and suitability for extended operation make it an ideal choice for both demonstration purposes and future scaling. Compared to smaller alternatives like the Adafruit 2-Liter Reservoir, which would require frequent refills, the Aqua-Tainer ensures longer operation with minimal maintenance while remaining affordable and widely available.

Update:

For the final GreenSync prototype, the team opted to use a **standard plastic cubby bin** as the water reservoir instead of a commercial tank or hydroponic container. This choice aligned with the project’s modular 6-cubby setup, where different compartments housed the plants, electronics, and irrigation components. The cubby selected for the reservoir offered sufficient depth to fully submerge the submersible DC pump while still being lightweight, easy to clean, and simple to integrate with basic vinyl tubing. Although it lacks specialized features such as built-in ports or opaque walls, the cubby provided more than enough capacity for repeated watering cycles during demonstrations and allowed the entire system to remain compact and visually cohesive. Using an off-the-shelf plastic cubby also minimized cost and

fabrication time, making it a practical and effective reservoir solution for a prototype-level smart gardening system.

3.3 Lighting Systems

3.3.1 Light Distribution Systems

Light plays a critical role in plant health, influencing photosynthesis, nutrient uptake, and overall growth. In indoor or controlled environments, sunlight is often insufficient or inconsistent, making it necessary to implement structured light distribution strategies. For GreenSync, the goal is to ensure reliable illumination while minimizing complexity and energy consumption. Below are five commonly used approaches to delivering and managing light for plants in smart gardening systems:

1. Natural Light Utilization

The most basic and sustainable approach is to rely entirely on natural sunlight. This method requires no additional hardware, reducing energy use and overall cost. However, it introduces major limitations in terms of consistency and control. Seasonal changes, cloudy days, and the orientation of windows can all impact light exposure, making it unreliable for autonomous plant care. While natural light can be monitored with sensors to adjust watering schedules or provide alerts, it often needs to be supplemented for consistent results. [61]

2. Fixed Overhead Lighting

This approach places a stationary artificial light source directly above the plant, ensuring even coverage regardless of outdoor conditions. Fixed lighting systems are cost-effective, easy to implement, and integrate well with automation, as they can be controlled based on light sensor data. For single-plant or small-scale systems like GreenSync, this is one of the most practical solutions, offering predictable results without unnecessary complexity. [62]

3. Adjustable Lighting Systems

For more advanced or multi-plant setups, adjustable lighting systems allow users to modify the position and angle of light sources as plants grow. This can be done manually or through automated rails and brackets, providing optimal light coverage for plants of different heights or arrangements. While this flexibility improves plant health and system adaptability, it requires

additional hardware and potentially motorized components, increasing both cost and design complexity. [63]

4. Modular or Clip-On Lighting

Clip-on grow lights or modular light units offer a flexible and low-cost solution for users with multiple small plants or limited space. These lights can be attached directly to pots, shelves, or other surfaces, allowing targeted lighting for individual plants. However, modular lights often lack integration with smart control systems and may not provide uniform light distribution for larger plants or grouped arrangements. Their simplicity makes them popular for casual users but less ideal for fully automated systems. [64]

5. Dynamic Adaptive Lighting

The most advanced approach to light distribution involves dynamic control of brightness, angle, and spectrum based on real-time sensor feedback and programmed cycles. These systems aim to replicate natural daylight patterns, including gradual intensity changes throughout the day, to promote healthy growth and flowering. While highly effective, adaptive lighting is typically used in commercial greenhouses or research environments due to its cost, complexity, and processing requirements. For a single-plant GreenSync prototype, this option represents a potential future upgrade rather than an immediate implementation. [65]

Table 13: Light Distribution Methods

METHOD	PROS	CONS	COST	COMPLEXITY
Natural Light	Free; Energy-efficient; No hardware needed	Unreliable; Seasonal variation; Hard to automate	Very Low	Very Low
Fixed Overhead Lighting	Consistent coverage; Easy to automate; Affordable	No height adjustment; Single-zone focus	Low	Low
Adjustable Lighting	Adaptable as plants grow; Better light angles	Requires brackets or motorization; More parts	Medium	Medium
Modular/Clip-On Lights	Flexible placement; Good for multi-plant setups	Uneven lighting; Poor integration with automation	Low	Medium
Dynamic Adaptive Lighting	Mimics natural cycles; Optimizes growth	Expensive; Requires sensors & advanced control	High	High

Selection:

Fixed overhead lighting was selected for GreenSync because it offers the best balance of simplicity, cost-effectiveness, and reliability for a compact smart gardening system. Unlike natural light, which is inconsistent and difficult to control, fixed lighting ensures uniform illumination regardless of

environmental conditions. It also integrates seamlessly with automation, allowing the ESP32 and light sensor to manage on/off cycles based on real-time ambient light readings. Compared to adjustable or adaptive systems, fixed lighting eliminates the need for additional mechanical components and complex control logic, reducing both hardware costs and power consumption. This makes it the most practical choice for the current GreenSync prototype while leaving room for future upgrades to more advanced lighting strategies if needed.

3.3.2 Grow Light Technologies

In controlled-environment gardening systems like GreenSync, selecting the right grow light technology is essential for ensuring healthy plant growth while balancing energy efficiency and cost. Unlike natural sunlight, which provides a complete and balanced spectrum, artificial lighting must replicate these conditions or deliver specific wavelengths to support photosynthesis and growth stages. Each lighting technology offers unique trade-offs in energy use, heat generation, spectral accuracy, and ease of integration with automated systems. Below is an overview of six commonly used grow light technologies, highlighting their suitability for compact, automated solutions like GreenSync.

1. Fluorescent Grow Lights

Fluorescent grow lights, including T5 tubes and compact fluorescent lamps (CFLs), are among the most common lighting solutions for small-scale gardening. They provide a relatively broad light spectrum, with a strong emphasis on blue wavelengths that promote vegetative growth. Fluorescents produce less heat than older technologies and are affordable, making them popular for seedlings and indoor plant propagation. However, they are bulkier, less energy-efficient than LEDs, and require frequent replacement, limiting their practicality for space-saving, long-term automated systems such as GreenSync. [75]

2. High-Intensity Discharge (HID) Lamps

HID lamps—such as Metal Halide (MH) and High-Pressure Sodium (HPS)—deliver intense light output, making them ideal for large-scale operations or plants requiring high light levels. MH lamps favor blue light for vegetative stages, while HPS emits more red/orange light for flowering. Despite their effectiveness in commercial horticulture, HID systems generate significant heat, require complex ventilation, and consume large amounts of

energy. These drawbacks make them incompatible with GreenSync's compact, low-power design goals. [76]

3. LED Grow Lights

LED technology has become the leading solution for indoor gardening due to its energy efficiency, long lifespan, and customizable light spectra. LEDs can produce full-spectrum light or focus on specific wavelengths such as red and blue, which are critical for photosynthesis. They generate minimal heat, reducing the need for active cooling, and integrate easily with automated control systems via low-voltage drivers. For GreenSync, LEDs represent the most practical and cost-effective choice, offering consistent light quality with minimal energy consumption in a compact form factor. [77]

4. Incandescent Grow Bulbs

Incandescent bulbs were once common for plant lighting but are now considered obsolete for serious horticultural applications. Although they produce a continuous light spectrum with an emphasis on red wavelengths, they are extremely inefficient, converting most energy into heat rather than usable light. Their short lifespan, excessive heat output, and high energy consumption make them unsuitable for modern automated systems beyond very basic experiments. [78]

5. Plasma Grow Lights

Plasma grow lights offer a nearly complete spectrum that closely mimics natural sunlight, making them highly effective for advanced research and premium plant care. However, their benefits come at a high cost in both equipment and energy consumption. Plasma lights also require specialized drivers and generate considerable heat, making them better suited for large-scale commercial or laboratory environments rather than compact, consumer-grade systems like GreenSync. [79]

6. Infrared (IR) Lighting

Infrared lighting operates outside the visible spectrum and does not directly drive photosynthesis but influences plant morphology and development stages. It can be beneficial for enhancing growth rates or triggering flowering responses when integrated into full-spectrum systems. However, IR lighting is typically used as a supplementary feature in advanced horticultural setups rather than as a primary light source. For GreenSync, IR integration is

unnecessary at this stage but may be considered in future upgrades for specialized plant growth optimization. [80]

Table 14: Grow Light Technologies

TECHNOLOGY	ENERGY CONSUMPTION	LIFESPAN	COST	HEAT EMISSION	LIGHTING CONTROL
Fluorescent	Medium	10,000–20,000 hrs	Low	Low–Medium	Limited (On/Off only)
HID (MH/HPS)	High	10,000–24,000 hrs	Medium	High	Limited (Manual Dimming)
LED	Low	50,000+ hrs	Medium	Very Low	Excellent (PWM, Smart Control)
Incandescent	Very High	~1,000 hrs	Very Low	Very High	Minimal (On/Off only)
Plasma	Very High	30,000 hrs	Very High	High	Limited (Complex Drivers)
Infrared (IR)	Medium (Supplemental)	30,000 hrs	High	Medium	Moderate (Spectrum Blending)

Selection:

LED grow lights were selected for GreenSync because they offer the best combination of energy efficiency, lifespan, and automation compatibility for a compact smart gardening system. Unlike traditional options such as fluorescent or HID lamps, LEDs consume significantly less power while delivering consistent light intensity and customizable spectra, which are critical for healthy plant growth. Their low heat output minimizes the need for additional cooling systems, making them ideal for small enclosures. Furthermore, LEDs support advanced control methods, including PWM dimming and integration with smart automation, allowing precise light scheduling based on sensor feedback. While the initial cost of LEDs is higher than that of basic fluorescent or incandescent lights, their durability, efficiency, and flexibility make them the most practical and future-proof choice for GreenSync.

3.3.3 ☀️ Light Sensing Technologies

Light sensing plays a critical role in automated gardening systems by monitoring ambient illumination levels and ensuring that plants receive the necessary light for healthy growth. In environments where natural light is inconsistent, these sensors provide data that allows systems to decide when to activate supplemental grow lights. Several sensing technologies exist, each with different characteristics in terms of accuracy, cost, response speed, and integration complexity. Below are the most commonly used light-sensing technologies in embedded systems and smart garden applications.

1. Photoresistors (LDRs)

Photoresistors, or Light Dependent Resistors (LDRs), are among the simplest and most affordable options. Their resistance decreases as light intensity increases, providing an analog signal that can be read by a microcontroller's ADC. LDRs are best suited for basic light detection tasks where precision is not critical. However, they lack the ability to measure exact light levels, and their performance can vary under changing environmental conditions. [66]

2. Photodiodes

Photodiodes operate as semiconductor devices that generate a current when exposed to light. They offer fast response times and can be designed to detect specific wavelengths, making them suitable for precise applications. However, because they produce very small currents, photodiodes typically require amplification and conditioning circuits, adding to system complexity. [67]

3. Phototransistors

Phototransistors function similarly to photodiodes but include internal gain, which amplifies the signal and reduces the need for external amplification. This makes them easier to interface with microcontrollers while providing better sensitivity than simple photodiodes. However, they are slower than photodiodes and offer less spectral selectivity. [68]

4. Digital Ambient Light Sensors

Modern digital light sensors combine photodiode arrays with onboard processing, providing calibrated digital output in standardized units such as lux. These sensors often support communication protocols like I²C or SPI, making integration with microcontrollers straightforward. They also offer adjustable gain and measurement ranges for adaptability across different environments. While highly accurate and easy to use, they are more expensive than analog alternatives and may require additional libraries for software integration. [69]

5. Multichannel Spectral Sensors

Advanced light sensing technologies include sensors that measure multiple spectral bands, such as visible and infrared light. These sensors provide greater precision by compensating for environmental factors and are capable of simulating human eye response or plant-optimized lighting. While ideal for high-end horticultural systems, these sensors are the most expensive and

complex to integrate, making them less practical for small-scale consumer products. [70]

Table 15: Light Sensing Technologies

TECHNOLOGY	OUTPUT TYPE	ACCURACY	COMPLEXITY	PROS	CONS	COST
Photoresistor (LDR)	Analog (resistance)	Low	Very Low	Cheap; Easy to use; Low power	Poor precision; Affected by environment	Very Low
Photodiode	Analog (current)	Medium	Medium	Fast response; Detects specific wavelengths	Requires amplification; Sensitive to noise	Low
Phototransistor	Analog (voltage)	Medium-High	Low	Built-in gain; Easy MCU interface	Slower than photodiode; Limited spectral range	Low
Digital Light Sensor	Digital (I ² C/SPI)	High	Low-Medium	Accurate; Lux output; No ADC needed	Requires libraries; Higher cost	Medium
Multichannel Spectral Sensor	Digital (I ² C)	Very High	High	Measures full spectrum; Great precision	Expensive; Complex integration	High

Selection:

Digital light sensors were chosen for GreenSync because they offer high accuracy, simplified integration, and reliable lux-based measurements compared to analog alternatives. Unlike photoresistors or phototransistors, which require ADC conversion and calibration, digital sensors provide direct digital output via communication protocols such as I²C, reducing processing complexity. They also feature built-in compensation for temperature and environmental variations, ensuring consistent performance under different conditions. This precision allows the system to make more intelligent decisions about grow light activation, optimizing energy usage while maintaining proper illumination levels for plant health. While more expensive than basic analog sensors, their accuracy and ease of implementation make digital light sensors the most practical choice for an automated gardening system like GreenSync.

Components:

1. VEML7700 Ambient Light Sensor

The VEML7700 is a highly accurate ambient light sensor that outputs lux readings through an I²C interface. It features a wide dynamic range (up to 120,000 lux), making it suitable for environments with varying light conditions. Its integrated calibration and low power consumption simplify implementation in microcontroller-based systems. The sensor is particularly useful for applications where precise light measurements are critical to controlling grow light activation. [71]

2. BH1750 Ambient Light Sensor

The BH1750 is a popular digital sensor for ambient light detection, also communicating over I²C. It provides 16-bit lux readings with moderate accuracy and a measurement range up to about 65,000 lux. While slightly less precise than higher-end models, the BH1750 is inexpensive and widely supported by libraries, making it a good fit for budget-conscious projects requiring basic light control functionality. [72]

3. TSL2591 Digital Light Sensor

The TSL2591 offers advanced features, including dual-channel measurements for visible and infrared light and a very high dynamic range (up to 88,000 lux). It includes programmable gain and integration time, allowing it to adapt to extremely low or high light environments. While this flexibility makes it suitable for complex horticultural systems, its configuration complexity and higher cost may be excessive for simple consumer-grade applications. [73]

4. OPT3001 Digital Ambient Light Sensor

The OPT3001 is a precision digital sensor from Texas Instruments that outputs lux readings via I²C with built-in photopic response, closely simulating human eye perception. It offers excellent linearity and low power consumption, making it suitable for portable or low-energy designs. However, it tends to be more expensive and may require additional configuration compared to simpler alternatives like the BH1750. [74]

Table 16: Light Sensor Component

SENSOR	INTERFACE	DYNAMIC RANGE	ACCURACY	POWER USE	PROS	CONS	COST
VEML7700	I ² C	Up to 120,000 lux	High	Low	Wide range; Accurate; Easy integration	Slightly higher cost	Medium
BH1750	I ² C	Up to 65,000 lux	Medium	Low	Affordable; Simple to use	Lower precision; Smaller range	Low
TSL2591	I ² C	Up to 88,000 lux	High	Medium	Dual-channel; Adjustable gain	More complex setup; Higher cost	High
OPT3001	I ² C	Up to 83,000 lux	Very High	Low	Photopic response; Excellent accuracy	Expensive; Less hobby support	High

Selection:

The VEML7700 was selected for GreenSync because it provides the optimal balance of accuracy, integration simplicity, and performance for an automated gardening system. Its wide dynamic range of up to 120,000 lux ensures reliable operation in both low-light indoor environments and bright sunlight near windows. Unlike more basic options like the BH1750, which offers limited precision, the VEML7700 delivers highly accurate lux readings without requiring complex calibration or extensive configuration. Compared to advanced sensors such as the TSL2591 or OPT3001, the VEML7700 achieves similar measurement accuracy while maintaining lower cost and easier software integration, making it ideal for a compact, cost-effective system like GreenSync. Its low power consumption and compatibility with the I²C protocol further streamline implementation with the ESP32 microcontroller, ensuring efficient and dependable light-based automation.

3.3.4 🌞 LED Grow Light

When it comes to providing supplemental lighting for indoor plant systems like GreenSync, choosing the right grow light is crucial to maintaining healthy plant development. Since plants rely on specific wavelengths of light for processes like photosynthesis and photomorphogenesis, LED grow lights must be carefully selected based on spectrum, intensity, and control. Here, we explore five distinct types of LED grow light configurations that offer varying degrees of suitability for smart garden applications.

1. Roleadro LED Grow Light Panel (75W Equivalent)

The Roleadro LED Grow Light Panel is a popular choice for small indoor gardens due to its slim design and balanced light spectrum that includes both red and blue wavelengths. It operates at an actual power draw of approximately 45W, making it highly energy-efficient while providing enough intensity for single-plant or small setups. Its full-spectrum output supports both vegetative and flowering stages, eliminating the need to switch lighting systems as the plant matures. While cost-effective and easy to mount, this panel lacks advanced dimming features and requires an external timer for automated control, which may limit integration flexibility for fully automated systems. [81]

2. GE BR30 LED Grow Light Bulb

The GE BR30 is an LED grow bulb designed to fit into standard E26 lamp sockets, making it a convenient option for retrofitting existing fixtures. It emits a full-spectrum light optimized for photosynthesis and has a relatively low power consumption of about 9W, which is suitable for a single plant or small

herb garden. Its compact size and affordability make it appealing for budget-conscious users, but its light coverage is limited, and it does not support dimming or smart control out of the box. For simple and space-efficient installations, however, it remains a solid choice. [82]

3. Spider Farmer SF-1000 LED Grow Light Bar

The Spider Farmer SF-1000 is a premium LED grow light designed for serious indoor gardening. It uses high-quality diodes to deliver a broad spectrum with excellent light penetration, covering an area of up to 2' x 2'. Consuming around 100W, it offers superior brightness and includes dimming functionality, which enables precise control of light intensity when paired with automation systems. While this light offers top-tier performance and build quality, its higher cost and physical size make it more suitable for medium-scale gardens than for a compact, single-plant system like GreenSync. [83]

4. PicoFarm LED Grow Light Panel (18W)

The PicoFarm LED panel is a compact, efficient solution designed for small-scale indoor gardening. Drawing just 18W, it produces high brightness at 220 lumens per watt while generating minimal heat, making it safe for continuous use in close proximity to plants. It uses 36 LM301b full-spectrum LEDs, which provide the wavelengths necessary for all stages of plant growth. Measuring only 11 × 10 cm, the PicoFarm is versatile in tight spaces and integrates easily with automation systems through its onboard driver that supports 3.3V or 5V PWM and analog control. These features make it highly suitable for GreenSync, offering both energy efficiency and direct microcontroller compatibility for automated lighting control.

Table 17: Grow Light Component

MODEL	POWER DRAW	SPECTRUM	COVERAGE AREA	PROS	CONS	COST
Roleadro LED Grow Light Panel	~45W	Full Spectrum (Red + Blue)	~2' x 2'	Energy-efficient; Good light intensity; Affordable	No dimming control; External timer needed	\$30–\$40
GE BR30 LED Grow Light Bulb	~9W	Full Spectrum	Single Plant	Compact; Fits standard sockets; Low cost	Limited coverage; No dimming; Low intensity	\$10–\$15
Spider Farmer SF-1000	~100W	Full Spectrum (Premium diodes)	Up to 2' x 2'	High-quality light; Dimmable; Great for automation	High cost; Large for small systems	\$120–\$150

Selection:

The Roleadro LED Grow Light Panel was selected for GreenSync because it provides the ideal balance of performance, cost, and integration simplicity for a compact smart gardening system. With an actual power draw of approximately 45W, it delivers sufficient light intensity for healthy plant growth while maintaining energy efficiency. Its full-spectrum output supports all stages of plant development, eliminating the need for multiple lighting solutions. Unlike high-end options like the Spider Farmer SF-1000, which offer premium features at a significantly higher cost and larger size, the Roleadro panel remains affordable and space-efficient, making it well-suited for a single-plant system. While it lacks built-in dimming control, this limitation can be easily addressed with an external timer or relay-based automation, ensuring seamless integration with the ESP32-controlled system. Its affordability, reliability, and balanced feature set make it the most practical choice for GreenSync's lighting solution.

Update:

For the final GreenSync prototype, the team selected the **WS2812B individually addressable LED strip** as the primary grow light solution. Although WS2812B LEDs are not traditional horticulture LEDs, their fully controllable RGB channels allowed the system to simulate a wide range of color temperatures and custom light patterns while remaining extremely easy to integrate with the ESP32. This flexibility made it possible to create experimental lighting modes, plant-friendly brightness levels, and visual feedback animations within a single lighting module. The strip's low voltage requirement, compact form factor, and minimal heat output made it safe for close-range placement inside the cubby system. While it does not provide the full-spectrum, high-intensity output of dedicated grow lights such as the PicoFarm or Spider Farmer units, the WS2812B strip offered unmatched control, programmability, and ease of installation—features that aligned perfectly with GreenSync's focus on automation, user interaction, and modular prototyping. Its affordability and seamless microcontroller compatibility made it the most practical and versatile lighting solution for the project.

3.4 Camera Integration

3.4.1 Camera Hardware Technologies

Incorporating a camera into GreenSync enhances the system's functionality by enabling visual monitoring of plant health, growth tracking, and potential future features such as automated image-based diagnostics or AI-driven plant

care recommendations. Unlike other sensors, which provide numerical data, a camera offers contextual information that users can view remotely via the app or web dashboard, improving engagement and trust in the system's automation. However, camera integration introduces additional considerations such as image quality, processing requirements, power consumption, and network bandwidth. For GreenSync, selecting a camera solution that balances affordability, ease of integration, and wireless connectivity is critical for maintaining system efficiency without adding unnecessary complexity.

1. USB Webcam

USB webcams are widely available and offer plug-and-play functionality with excellent image quality. They are compatible with PCs, Raspberry Pi, and similar platforms, making them a common choice for DIY projects. However, they require a host device with a USB interface and significant processing power, making them unsuitable for compact microcontroller-based systems like GreenSync. [84]

2. IP Camera

IP cameras operate as standalone network devices, streaming video over Wi-Fi or Ethernet. They provide good image quality and remote access features but are relatively expensive and require complex setup for integration with embedded systems. Their higher power consumption and advanced networking requirements make them more appropriate for security systems than for compact automated gardens. [85]

3. PTZ Camera (Pan-Tilt-Zoom)

PTZ cameras offer advanced functionality by allowing remote adjustment of the viewing angle and zoom level. While ideal for large areas or professional monitoring setups, they are bulky, power-hungry, and prohibitively expensive for consumer-oriented systems like GreenSync, where space and energy efficiency are priorities. [86]

4. Onboard MCU-Compatible Camera Modules

Camera modules designed for microcontrollers, such as the ESP32-CAM, provide an affordable and compact solution for IoT applications. These modules integrate directly with the main controller, often featuring built-in wireless communication for easy streaming. While they do not match the resolution or quality of high-end cameras, their low cost, small footprint, and straightforward integration make them an excellent choice for smart gardening systems that require basic visual monitoring. [87]

5. DSLR or High-End Cameras

Professional-grade cameras offer superior image quality and flexibility, but their cost, size, and power requirements make them completely impractical for embedded systems. These cameras are reserved for specialized use cases such as research or commercial photography, not for consumer-level automation projects. [88]

Table 18: Camera Technologies

TECHNOLOGY	IMAGE QUALITY	POWER USE	INTEGRATION COMPLEXITY	COST	PROS	CONS
USB Webcam	High	Medium	High (requires host device)	Low–Medium	Good image quality; Widely available	Needs PC/Raspberry Pi; Not MCU-friendly
IP Camera	High	High	High	Medium–High	Standalone streaming; Remote access	Expensive; High bandwidth and power requirements
PTZ Camera	High–Very High	Very High	Very High	High	Flexible viewing; Professional features	Bulky; Costly; Overkill for small systems
MCU-Compatible Module	Medium	Low	Low	Low	Compact; Low power; Easy to integrate	Limited resolution; No advanced optics
DSLR / High-End Camera	Excellent	Very High	Very High	Very High	Superior image quality; Full manual control	Impractical size, cost, and power for IoT systems

Selection:

MCU-compatible camera modules were selected for GreenSync because they provide the best balance of size, cost, and integration simplicity for a compact, microcontroller-based system. Unlike USB webcams or IP cameras, which require a dedicated host device and higher power consumption, MCU-compatible modules integrate directly with the microcontroller and typically include built-in support for wireless data transmission. This eliminates the need for additional hardware, reducing complexity and cost. While these modules do not offer the high resolution of professional cameras, their image quality is more than sufficient for monitoring plant growth and capturing periodic status updates. Their low power requirements and small footprint make them ideal for embedded applications, ensuring that GreenSync remains efficient and user-friendly while incorporating visual monitoring capabilities.

3.4.2 Camera Hardware Components

1. ESP32-CAM (with OV2640)

The ESP32-CAM is a compact development board that integrates the ESP32 microcontroller with an OV2640 camera sensor. It supports wireless connectivity through Wi-Fi and Bluetooth, making it suitable for IoT-based image applications. The board also includes a microSD card slot for local storage and is programmable through common development environments like Arduino IDE or ESP-IDF. Offering a 2MP resolution, it provides basic image capture capabilities for applications requiring lightweight visual monitoring. [89]

2. Arducam Mini Camera Module (OV2640)

The Arducam Mini is a camera module based on the OV2640 sensor, commonly used in embedded systems for basic image processing tasks. Unlike integrated boards, this module connects to a separate microcontroller via the SPI interface. It provides flexibility for modular designs and can be incorporated into systems requiring a detached camera component. However, it does not include onboard processing, so its integration depends on the capabilities of the host microcontroller. [90]

3. OV5640 Camera Module

The OV5640 module is a higher-resolution camera sensor offering up to 5MP image capture and advanced imaging features, including autofocus support. It is typically used in applications requiring improved detail and low-light performance. The module connects using DVP or MIPI interfaces, which may require specialized hardware support and drivers. Due to its advanced capabilities, the OV5640 is more complex to integrate and generally intended for systems with higher processing power requirements. [91]

4. 1080P HD USB Webcam

The 1080P HD USB Webcam is a plug-and-play camera designed for desktops and laptops, offering full HD video capture with auto light correction and a built-in microphone featuring noise cancellation. It includes a privacy cover and a wide-angle lens, making it versatile for both streaming and monitoring applications. Unlike sensor-only modules, this webcam connects directly via USB, simplifying integration with the Raspberry Pi 4 for live streaming. Its affordability, ease of setup, and compatibility with common software environments make it an excellent fit for GreenSync's high-resolution imaging needs

Table 19: Camera Component

Component	Resolution	Interface	Connectivity	Pros	Cons	Cost
ESP32-CAM (OV2640)	2 MP	UART / SPI	Built-in Wi-Fi & BT	Integrated MCU and camera; Wireless support; Includes microSD storage	Limited resolution; Requires flashing tools	Low (\$8-\$12)
Arducam Mini (OV2640)	2 MP	SPI	None	Compact; Flexible for modular systems	Needs external MCU; Additional wiring and coding	Low-Medium (\$12-\$18)
OV5640 Module	5 MP	DVP / MIPI	None	Higher resolution; Better low-light performance	Complex integration; Requires more processing power	Medium-High (\$20-\$35)

Selection:

The ESP32-CAM was selected for GreenSync because it offers an all-in-one solution that combines processing capability, camera functionality, and wireless communication in a single compact board. Unlike standalone camera modules, it eliminates the need for an additional microcontroller, reducing hardware complexity and cost. Its built-in Wi-Fi allows seamless image transmission to the GreenSync dashboard without requiring extra networking components, and the inclusion of a microSD card slot provides optional local storage for captured images. While its 2MP resolution is modest compared to higher-end modules like the OV5640, it is sufficient for visual monitoring and growth tracking. This balance of affordability, ease of integration, and functional completeness makes the ESP32-CAM the most practical choice for GreenSync’s camera system.

3.5 Communication & Integration

3.5.1 Wireless Communication Protocols

Wireless communication technologies are the backbone of modern smart systems, allowing devices to exchange data without physical connections. Different protocols vary in terms of speed, range, power requirements, and network complexity, making protocol selection a critical decision for GreenSync. Below are five commonly used wireless communication protocols for IoT and smart automation systems:

1. Wi-Fi

Wi-Fi is one of the most widely adopted wireless communication protocols, offering high data transfer speeds and robust connectivity for IoT devices. It enables seamless integration with cloud services, mobile applications, and web dashboards, making it ideal for systems requiring real-time data access

and user control. [92] However, Wi-Fi consumes more power than other low-energy protocols and typically requires an existing router or network infrastructure. For stationary systems like GreenSync, Wi-Fi provides excellent convenience and functionality without additional gateways.

2. Bluetooth Low Energy (BLE)

BLE is a power-efficient wireless protocol designed for short-range communication, commonly used in wearable devices and compact IoT systems. It offers moderate data rates and is supported by most smartphones, enabling simple mobile app pairing without additional hardware.[93] While BLE is ideal for low-power designs, its limited range and slower speed compared to Wi-Fi make it less suited for systems that require constant remote monitoring or large data transfers such as image uploads.

3. Zigbee

Zigbee is a low-power, mesh-networking protocol widely used in home automation. It excels in environments requiring multiple interconnected devices, as its mesh topology enhances coverage and reliability. [94] Zigbee's low power consumption makes it suitable for battery-operated systems. However, it requires a dedicated Zigbee hub or coordinator to connect to standard networks or the internet, which adds complexity and cost for end-users. This makes it less appealing for simple, single-node systems like GreenSync.

4. LoRa (Long Range)

LoRa is designed for long-distance, low-power communication, making it an excellent choice for outdoor and agricultural IoT applications where connectivity over several kilometers is needed. [95] It operates at low data rates, which is ideal for transmitting small sensor data packets but unsuitable for high-bandwidth tasks like image transmission. Implementing LoRa also requires specialized hardware and a gateway for internet access, adding additional complexity for consumer-grade solutions.

5. NRF24L01 (Proprietary 2.4GHz)

The NRF24L01 module is a low-cost transceiver commonly used for wireless communication in embedded systems. It offers low power consumption and moderate range, making it attractive for DIY IoT projects. [96] However, it lacks standardized protocols and smartphone compatibility, which significantly limits its utility in systems that require direct user interaction through mobile

applications. Integration with cloud-based services also requires additional hardware, making this option less favorable for GreenSync.

Table 20: Wireless Protocols

PROTOCOL	SPEED	RANGE	POWER USE	INTEGRATION COMPLEXITY	COST	PROS	CONS
Wi-Fi	High (Up to 100 Mbps)	~50–100 m indoors	High	Low	Low	Fast; Direct internet access; Works with mobile apps	Higher power consumption; Requires router/network
BLE	Moderate (1 Mbps)	~10–30 m	Very Low	Low	Low	Energy-efficient; Native smartphone support	Limited range; Slower data transfer
Zigbee	Low (250 Kbps)	~10–100 m (Mesh)	Low	Medium	Medium	Great for multi-device networks; Low power	Requires hub/gateway; Limited bandwidth
LoRa	Very Low (Up to 50 Kbps)	Up to 10 km+	Very Low	High	Medium	Excellent long-range communication; Low power	Very slow; Requires gateway; Unsuitable for large data
NRF24L01	Moderate (2 Mbps)	~50–100 m	Low	Medium	Very Low	Inexpensive; Low power	No smartphone compatibility; Requires extra components

Selection:

Wi-Fi was selected as the wireless communication protocol for GreenSync because it offers the best combination of speed, compatibility, and ease of integration for an IoT-based gardening system. Unlike low-power alternatives such as BLE or Zigbee, Wi-Fi enables direct connectivity to existing home networks, allowing real-time data transmission and remote access through a mobile app or web dashboard without requiring additional hubs or gateways. Its high data rate supports future features such as image uploads from the integrated camera, which would be impractical with slower protocols like LoRa. While Wi-Fi consumes more power than BLE, this trade-off is acceptable for GreenSync’s stationary design, which uses a reliable DC power source. Overall, Wi-Fi provides a robust, cost-effective, and user-friendly solution for enabling remote monitoring and control.

3.5.2 Sensor Integration Methods

Integrating sensors with the main controller is a critical aspect of any IoT-based system, as it determines how efficiently data is transmitted and processed. For GreenSync, the integration method must ensure reliable communication between the ESP32 microcontroller and multiple peripherals, such as the soil moisture sensor, light sensor, and pump control circuits. Common integration methods include **GPIO**, **I²C**, **SPI**, and **UART**, each with different trade-offs in terms of speed, wiring complexity, scalability, and data

handling capabilities. Selecting the appropriate methods is essential for achieving system stability and reducing hardware complexity. Below is an overview of these four integration approaches.

1. GPIO (General Purpose Input/Output)

GPIO pins are the simplest and most versatile interface available on microcontrollers. They allow digital HIGH or LOW signals for basic on/off control, making them ideal for simple devices like LEDs, relays, or push buttons. [97] GPIO is easy to implement, requiring minimal programming and no special communication protocols. However, it is not suitable for transmitting complex data, as each signal represents only a binary state. For GreenSync, GPIO is primarily used for controlling the relay or MOSFET that drives the water pump and solenoid valve.

2. I²C (Inter-Integrated Circuit)

I²C is a widely used serial communication protocol designed for short-distance, multi-device communication. It uses just two lines (SDA for data and SCL for clock), allowing multiple sensors to share the same bus through unique addresses. [98] This reduces wiring complexity and makes I²C highly scalable for adding components like light sensors or environmental sensors. While slower than SPI, I²C offers sufficient speed for most sensor applications and is well-supported on the ESP32, making it ideal for managing GreenSync's VEML7700 light sensor and future expansion.

3. SPI (Serial Peripheral Interface)

SPI is another serial communication protocol that provides higher data transfer speeds than I²C, making it suitable for applications that require fast or continuous data exchange, such as displays or high-speed sensors. [99] However, SPI requires more wires, typically four (MISO, MOSI, SCLK, and CS) and an additional chip-select pin for each device. This increases hardware complexity in multi-device systems. For GreenSync, SPI is less critical since the sensors used do not require high-speed data transfers, but it remains an option for future upgrades involving faster peripherals.

4. UART (Universal Asynchronous Receiver/Transmitter)

UART is a point-to-point communication protocol widely used for serial data transfer between two devices, such as a microcontroller and a module. It requires only two wires (TX and RX) but does not support multi-device configurations without additional hardware. [100] UART is ideal for debugging

and communicating with peripherals like Bluetooth modules or GPS receivers. In GreenSync, UART is primarily relevant for programming the ESP32-CAM module and could be utilized for optional external modules in future designs.

Table 21: Sensor Integrations

METHOD	SPEED	COMPLEXITY	WIRING	SCALABILITY	PROS	CONS
GPIO	Instant (digital)	Very Low	One pin per device	Poor	Simple; No protocol required; Great for on/off control	Not suitable for complex data; Consumes many pins
I ² C	Moderate (~100 kbps–1 Mbps)	Low	2 wires for all devices	Excellent	Supports multiple devices on shared bus; Minimal wiring	Limited speed compared to SPI; Requires addressing
SPI	High (up to 10 Mbps+)	Medium	4 wires + extra per device	Good	Very fast; Reliable data transfer	More wiring; Uses more MCU pins
UART	Moderate (~115 kbps–1 Mbps)	Low	2 wires (TX/RX)	Poor	Simple setup; Ideal for direct device-to-device link	Limited to point-to-point; No native multi-device support

Selection:

The combination of **I²C** and **GPIO** was selected for GreenSync because it offers the most practical balance between simplicity, scalability, and functionality. I²C provides a reliable, two-wire interface that allows multiple sensors—such as the ambient light sensor and future environmental monitors—to share a single communication bus, minimizing wiring and conserving microcontroller pins. This makes it highly scalable and efficient for the system’s sensor network. GPIO, on the other hand, is ideal for controlling simple digital devices like the relay or MOSFET that activates the water pump and solenoid valve. Together, these methods ensure streamlined integration for both control and data acquisition, reducing hardware complexity while maintaining flexibility for future expansion.

3.6 Software Systems

3.6.1 Data Logging & Storage Options

In an embedded system like GreenSync, where real-time environmental data and user commands must be tracked, storage is a critical design consideration. Reliable data logging supports system transparency, troubleshooting, and future scalability. The storage solution must balance

capacity, power consumption, complexity, and ease of access. Below are six commonly used storage options in microcontroller-based systems, evaluated for their relevance to GreenSync.

1. MicroSD Card Modules

MicroSD cards remain a leading option for local data storage in microcontroller projects, offering large non-volatile capacity in a compact format. Using SPI communication, they allow easy integration with boards like the ESP32 for saving images, logs, and timestamps. [101] Their removability makes data retrieval simple without relying on wireless connectivity. However, managing a FAT32 file system introduces software complexity, and frequent writes can draw more power compared to smaller-scale memory solutions. Despite these considerations, microSD cards provide excellent flexibility for moderate to large data storage needs.

2. EEPROM (Electrically Erasable Programmable Read-Only Memory)

EEPROM provides a reliable, small-scale non-volatile storage option for configuration values, calibration data, or last-known states. [102] Although it offers very limited capacity typically in kilobytes it is highly efficient for storing persistent settings. EEPROM's endurance is around 100,000 write cycles, making it unsuitable for frequent logging. The ESP32 lacks native EEPROM but can emulate this function using its internal flash memory. For GreenSync, EEPROM (or its emulation) is ideal for retaining network credentials or user-defined thresholds.

3. Flash Memory (Internal to MCU)

Flash memory embedded in the ESP32 primarily stores program code, but portions can be allocated for user data using SPIFFS or LittleFS partitions. [103] This approach eliminates additional hardware and is useful for lightweight storage tasks such as configuration files, log summaries, or credentials. While convenient, internal flash has limited write endurance and capacity, making it impractical for large or frequent data writes such as storing images or detailed sensor histories.

4. Cloud-Based Storage via Wi-Fi

Cloud storage offers real-time access and scalability by transmitting data from the device to a remote server or platform such as Google Firebase, AWS IoT, or Adafruit IO. [104] This enables users to monitor system activity from anywhere while reducing dependence on physical media. However, it requires

consistent internet access, higher power during transmissions, and additional backend development for authentication and dashboards. Despite these challenges, cloud storage provides excellent potential for future upgrades, especially for remote monitoring and predictive analytics.

5. External FRAM Modules (Ferroelectric RAM)

FRAM combines non-volatility with near-unlimited write endurance, making it ideal for frequent and rapid data logging without wear concerns. [105] It offers low power consumption and fast writes, outperforming EEPROM and flash in durability and speed. Integration is straightforward via I²C or SPI, but its cost and limited availability compared to traditional memory options make it less common in consumer-level products. For GreenSync, FRAM could support continuous high-frequency sensor logging in advanced implementations.

6. USB-Enabled External Storage

External USB storage offers extremely high capacity and flexibility for storing images, videos, or extended logs. [106] Implementing this option, however, requires additional hardware such as a USB host controller and careful power regulation. Software complexity also increases due to mass storage protocol handling. While unnecessary for the current GreenSync prototype, USB storage is relevant for large-scale or research-grade systems requiring extensive data retention or multimedia storage.

Table 22: Storage

STORAGE OPTION	CAPACITY	COST	POWER CONSUMPTION	DATA RETENTION	PROS	CONS
MicroSD Card	High (GBs)	Low–Medium	Moderate	Excellent (Years)	Large storage; Removable; Ideal for images and logs	Requires SPI interface; File system handling; Higher power
EEPROM	Very Low (KBs)	Low	Very Low	Excellent (Decades)	Great for configs and small data; Simple to implement	Very limited space; Limited write cycles (~100k)
Flash Memory (MCU)	Low (MBs)	Included in MCU	Low	Good	No extra hardware; Stores configs and small files	Limited endurance; Not for frequent logging
Cloud Storage	Virtually Unlimited	Medium–High	High (during Tx)	Unlimited (Remote)	Remote access; Scalable; Enables app dashboards	Requires internet; Increased complexity and power use
FRAM	Low–Medium (KB–MB)	High	Low	Excellent	Fast writes; High endurance; Ideal for frequent logging	Expensive; Less available than flash or EEPROM
USB Storage	Very High (GB–TB)	Medium–High	High	Excellent	Huge capacity; Supports advanced features like video logs	Complex hardware/software; Overkill for small systems

Selection:

MicroSD storage was selected for GreenSync because it provides an ideal balance of capacity, accessibility, and integration simplicity for a compact, IoT-based gardening system. With support for several gigabytes of non-volatile storage, MicroSD cards can reliably handle images captured by the ESP32-CAM as well as detailed sensor logs without frequent data offloading. Their removability offers users a straightforward way to retrieve data without relying on a network connection, which is particularly useful during initial development or offline scenarios. While cloud storage offers unmatched scalability and real-time remote access, it introduces additional complexity, requires stable internet connectivity, and consumes more power during data transmissions. For these reasons, cloud storage is reserved as a future enhancement if GreenSync evolves into a fully networked platform with advanced analytics and user dashboards.

3.6.2 Embedded Programming Environment

Selecting the right programming environment is essential for efficient firmware development, debugging, and maintenance in embedded systems like GreenSync. The development environment influences code portability, library support, and overall ease of integration with hardware components. For GreenSync, the chosen environment must support the ESP32 platform, provide robust community resources, and balance simplicity with flexibility for

future feature expansion. Below are three widely used environments evaluated for suitability in this project.

1. Arduino IDE

The Arduino IDE is one of the most popular platforms for embedded development due to its simplicity and extensive community support.[107] It provides a beginner-friendly interface, straightforward code structure, and built-in libraries for common sensors and peripherals. With support for the ESP32 through board packages, Arduino IDE simplifies programming, compilation, and uploading of firmware. Its limitations include basic debugging features and limited project organization capabilities, making it less suited for very large or complex projects. However, for GreenSync, which prioritizes rapid development and reliability, Arduino IDE offers an excellent starting point.

2. ESP-IDF (Espressif IoT Development Framework)

ESP-IDF is the official development framework for ESP32 provided by Espressif. [108] It offers low-level hardware access, advanced debugging tools, and direct integration with the latest ESP32 features. ESP-IDF supports FreeRTOS and other advanced functionalities, making it ideal for large-scale IoT applications requiring real-time multitasking and high performance. However, the learning curve is steeper compared to Arduino IDE, and development requires more complex build processes. While powerful, ESP-IDF may be excessive for a small consumer-grade system like GreenSync, though it remains a strong option for future versions that may need advanced capabilities.

3. PlatformIO

PlatformIO is a modern, cross-platform development environment that integrates with popular editors like VS Code and provides advanced features such as dependency management, unit testing, and multi-board support. [109] It bridges the gap between simplicity and professional-grade functionality, supporting both Arduino-based and ESP-IDF projects within a single ecosystem. While PlatformIO offers excellent flexibility, its setup can be slightly more complex than Arduino IDE, and beginners may find it overwhelming at first. Nevertheless, it is an excellent choice for teams aiming for scalability and modular development workflows.

Table 23: Embedded Environment

ENVIRONMENT	EASE OF USE	FEATURES	HARDWARE SUPPORT	DEBUGGING	COMPLEXITY	PROS	CONS
Arduino IDE	Very Easy	Basic (Core Libraries)	Wide (ESP32 supported)	Limited (Serial Monitor only)	Low	Simple setup; Large community; Extensive library support	Minimal debugging tools; Limited project management
ESP-IDF	Moderate–Hard	Advanced (FreeRTOS, RTOS)	Full ESP32 feature set	Excellent (GDB, Trace)	High	Official ESP32 framework; Access to low-level features	Steep learning curve; Longer development cycles
PlatformIO	Moderate	Rich (Multi-board, Testing)	Broad (ESP32, Arduino, more)	Good (Integrated with IDE)	Medium	Flexible; Works with VS Code; Dependency management	More complex setup; Requires familiarity with IDE tools

Selection:

Arduino IDE was selected as the programming environment for GreenSync due to its simplicity, strong community support, and seamless compatibility with the ESP32 platform. Its user-friendly interface and extensive library ecosystem allow for rapid development and straightforward integration of sensors, actuators, and communication protocols. Unlike more complex environments like ESP-IDF or PlatformIO, Arduino IDE minimizes the learning curve and reduces development time, making it ideal for a consumer-oriented prototype. While it lacks advanced debugging and large-scale project management features, these trade-offs are acceptable for GreenSync’s current scope, which prioritizes reliability and ease of implementation over highly complex functionality.

3.6.3 Programming Language

The choice of programming language in an embedded system determines not only performance and efficiency but also ease of development and compatibility with available libraries. For GreenSync, which runs on an ESP32 microcontroller, the language must provide reliable real-time performance, robust hardware control, and strong community support. Several options exist for microcontroller programming, ranging from traditional compiled languages like C/C++ to interpreted languages like MicroPython, and even higher-level frameworks for specialized IoT development. Below is an overview of the most relevant options for GreenSync.

1. C/C++ (Arduino Framework)

C and C++ remain the industry standard for embedded development due to their performance, efficiency, and low-level hardware access. [110] The

Arduino framework, built on C++, provides an abstraction layer that simplifies development while maintaining access to core microcontroller features. It offers an extensive library ecosystem, making integration with sensors, communication modules, and peripheral devices straightforward. The main drawback of C/C++ is its steeper learning curve compared to interpreted languages and the need for manual memory management, but these trade-offs are acceptable for performance-critical applications like GreenSync.

2. MicroPython

MicroPython is an embedded implementation of the Python programming language designed for microcontrollers. [111] It offers a more beginner-friendly syntax and faster development cycles compared to C/C++, making it attractive for rapid prototyping. However, as an interpreted language, MicroPython sacrifices execution speed and introduces higher memory consumption, which can limit its suitability for resource-constrained systems performing real-time tasks. While MicroPython runs on ESP32 and supports basic I/O and networking, its performance limitations make it less ideal for GreenSync's timing-sensitive operations, such as precise sensor reading and actuation control.

3. JavaScript (NodeMCU / IoT Frameworks)

JavaScript, through frameworks like Espruino or NodeMCU, has found its way into the IoT space, appealing to developers familiar with web technologies. [112] It simplifies event-driven programming and offers integration with web services, which can benefit cloud-connected projects. However, JavaScript-based environments are significantly slower than C/C++ and have limited support for low-level hardware features. They also add unnecessary overhead for GreenSync, where real-time control and efficiency are paramount. While viable for some IoT projects focused on connectivity, JavaScript is not recommended for embedded automation systems requiring precise timing and optimized resource use.

Table 24: Programming Language

LANGUAGE	PERFORMANCE	MEMORY USAGE	EASE OF USE	HARDWARE CONTROL	PROS	CONS
C/C++ (Arduino)	High (Compiled)	Low	Moderate	Excellent	Fast execution; Full control of hardware; Large library support	Steeper learning curve; Manual memory management
MicroPython	Moderate (Interpreted)	Moderate-High	Easy	Good	Simple syntax; Quick prototyping; Good for beginners	Slower execution; Limited performance for real-time tasks
JavaScript (IoT)	Low (Interpreted)	High	Moderate-Easy	Limited	Familiar for web developers; Event-driven model	Poor real-time performance; Limited low-level hardware access

Selection:

C/C++ was chosen as the programming language for GreenSync because it offers the performance, reliability, and hardware-level control necessary for real-time embedded applications. As a compiled language, C++ provides fast execution and efficient memory usage, ensuring responsive interactions between the ESP32 microcontroller and connected sensors, actuators, and communication modules. Its compatibility with the Arduino framework further simplifies development by providing an extensive library ecosystem and a familiar structure for rapid prototyping. While interpreted languages like MicroPython and JavaScript offer ease of use, their slower execution speed and higher resource requirements make them less suitable for time-sensitive operations. Therefore, C/C++ delivers the best balance of efficiency, control, and scalability for GreenSync’s requirements.

3.6.4 User Interface

A smart gardening system like GreenSync relies on an intuitive user interface to ensure ease of control and monitoring. Whether users want to check soil moisture, adjust lighting schedules, or initiate watering, the interface should be reliable, user-friendly, and accessible. Multiple options exist for implementing UI in embedded IoT systems, each offering distinct trade-offs in complexity, cost, and flexibility. Below are six common approaches considered for GreenSync.

1. Mobile Application Interface (Custom App)

A custom mobile app is one of the most modern and user-centric solutions. It provides real-time data visualization, notifications, and direct control over system functions such as watering and lighting. [113] Apps can connect via Wi-Fi or Bluetooth, and platforms like MIT App Inventor and FlutterFlow make development accessible without deep programming knowledge. This method

offers the best balance of portability and interactivity but requires additional design and development time.

2. Web Dashboard (Hosted or Local)

Web dashboards provide platform-independent access through any device with a browser. They can be served locally from the ESP32 or hosted on the cloud for remote connectivity. [114] Dashboards often include live charts, control buttons, and system logs. While highly versatile and easier to update than mobile apps, they depend on a stable network connection and require some web development expertise.

3. Serial Console Interface (USB or UART)

Serial consoles are essential for debugging and setup. By connecting via USB and using tools like the Arduino Serial Monitor, users can issue commands and view raw system outputs. [115] This method is simple and effective for developers but is not practical for everyday end-users, as it requires a physical connection and technical knowledge.

4. Physical Interface (Buttons and LEDs)

Physical interfaces provide basic control and feedback without requiring a phone or internet. Buttons can trigger manual watering or lighting, while LEDs indicate system states such as “watering” or “low moisture.” [116] This approach is low-cost, reliable, and useful as a fallback in case of connectivity issues, but it lacks advanced features like data visualization or remote access.

Table 25: Interface

UI OPTION	COMPLEXITY	COST	CONNECTIVITY	PROS	CONS
Mobile App	High	Medium	Wi-Fi / BLE	Intuitive GUI; Full control and monitoring; Real-time updates	Requires app development; Dependent on smartphone
Web Dashboard	Medium	Low–Medium	Wi-Fi	Platform-independent; Accessible via any browser	Needs stable network; Requires web development skills
Serial Console	Very Low	Low	USB (Local only)	Ideal for debugging; Simple and lightweight	Not user-friendly; Requires physical connection
Physical Interface	Low	Low	None	Reliable; No network needed; Easy for basic control	Limited functionality; No advanced monitoring

Selection:

The mobile application interface was selected for GreenSync because it offers the most user-friendly and feature-rich method for controlling and monitoring the system. A custom app allows real-time data visualization, remote access, and direct interaction with components such as the water pump and grow light through an intuitive graphical interface. Its compatibility with Wi-Fi ensures seamless connectivity to the ESP32 without requiring additional hardware. While options like web dashboards or physical controls provide useful alternatives, they lack the portability and interactive design that mobile apps deliver. This approach aligns with GreenSync's goal of providing a modern, convenient, and scalable user experience.

3.6.4.1 Comparisons of App Development Technologies

In order to determine the most effective method for developing the mobile component, we conducted research on the many development platforms and technologies that are currently accessible across a variety of criteria, including the following:

Table 26: Basic App Development Technology Comparison

	Mobile Operating Systems Supported	Required Coding Language(s)	Complexity
Progressive Web Apps (WPAs)	Any operating system with the latest browser	JavaScript, HTML, and CSS	Low
Cordova	Android and iOS	JavaScript, HTML, and CSS	Low-Medium
Ionic	Android and iOS	JavaScript, HTML, and CSS	Low-Medium
Kotlin	Android	Kotlin	High
Swift	iOS	Swift	High
Flutter	iOS and Android	Dart	Medium
React Native	Android and iOS	JavaScript	Medium

Table 27: Comprehensive Pros and Cons of Every Technology

	Pros	Cons	ESP32 Integration
Progressive Web Apps (WPAs)	Instant updates, a single codebase, and an app-like experience without requiring app store deployment	Very restricted access to device functionality; sensitive to previously configured privacy settings and reliant on browser capabilities	Indirect integration through the use of web protocols like WebSockets and HTTP.
Cordova	Using plugins to access native device capabilities, having a single codebase across platforms, and wrapping web apps in native containers	Performance restrictions and reliance on plugins for native features	Plugin integration; native code may be needed
Ionic	Quicker development, a single codebase across platforms, the use of web technologies, and a large collection of user interface elements	Performance restrictions and reliance on plugins for native features	Plugin integration; native code may be needed
Kotlin	Excellent speed, smooth interaction with Android APIs, and complete access to Android capabilities	Restricted to the Android environment; an independent codebase is needed for iOS	Direct ESP32 integration over Wi-Fi or BLE connectivity
Swift	Excellent speed, smooth	Restricted to the iOS	Direct ESP32 integration over

	interaction with iOS APIs, and complete access to iOS capabilities	environment; an Android codebase is needed.	Wi-Fi or BLE connectivity
Flutter	High speed, a rich collection of customisable widgets, a single code base for several platforms, and hot reload for real-time code updates	Larger program size; restricted access to third-party libraries; potential increases in development costs due to project scope	Accessible through native modules or plugins
React Native	Reusable components, a single code base for numerous platforms, robust community support, and hot reload for immediate user interface modifications	Dependency on third-party plugins, performance delays relative to native apps, and potential increases in development costs depending on project requirements	Possible through native modules or third-party libraries

The smooth integration of the ESP32 microcontroller for real-time user interaction was a non-negotiable need when deciding which application development technology to utilize in order to meet the requirements of our project. With regard to this particular role, Flutter and React Native came up as particularly excellent choices. As a result of the fact that both frameworks enable development across several platforms inside a single codebase, they also decrease the amount of time and money required for development overall. Furthermore, these hot reload capabilities make it possible to make changes to the code in real time, which allows us to accommodate the quick feedback and modifications that we had wanted to incorporate into our entire system. With regard to the needs of our project, the direct interface with the ESP32 was a feature that was very appealing in both Swift and Kotlin. WPAs, on the other hand, were effectively excluded from our scope of investigation since they were unable to contain either a direct or plugin-based link to the

MCU.

In spite of the fact that Swift and Kotlin are appealing due to their direct interaction with ESP32, our group did not find their limited capabilities inside the mobile operating system environment to be appealing. As a result, we eliminated them as potential candidates for our mobile development environment.

The reason for this is that a codebase that lacks unity may be a significant obstacle to the fulfilment of the project's requirements for real-time responsiveness to the input of the user and efficient integration of ESP32. In addition, that Ionic and Cordova were not on the list of technologies that were sought since there was a lack of expertise with both of them as well as resources addressing them.

When it came to the creation of the mobile application, both Flutter and React Native appeared to be excellent choices. Flutter, on the other hand, ended up being the best option for us because it better coincided with our expectations for the smart garden project and the technical requirements. Along with the unified codebase that is compatible with both operating system environments, the Hot Reload feature, excellent speed, Flutter's flexible user interface design, and the widget architecture all contribute to the simplicity of implementation. Furthermore, the plugin ecosystem support for Bluetooth Low Energy (BLE), Wi-Fi, and Firebase are all appropriate supporting aspects for the real-time connectivity that is necessary for our project with the ESP32. Despite the fact that it is necessary to write native code in order to incorporate intricate hardware capabilities, the plugin ecosystem that Flutter provides is able to substantially ease this process when it comes to combining the software and hardware communication.

By virtue of its architecture, performance features, and development tools, Flutter's architecture enables GreenSync to realize its full potential. Through the use of Ahead-of-Time (AOT) compilation, the Dart compiler that is included within Flutter is inserted straight into the native ARM machine code. This technique, in contrast to React Native, is able to circumvent JavaScript bridges, which results in shorter startup times and improved runtime performance. These are extremely important for a system like ours, which requires regular updates from the ESP32 module. The latency is greatly decreased in order to boost performance during continuous sensor updates or other control unit activities. This is made possible by the fact that none of

the JavaScript bridges that are required to interface with native APIs are required.

The data processing from real-time data streams received from the sensors may be simplified with the use of state management libraries such as Riverpod, Bloc, or Provider. This would ensure that the user interface is updated immediately. Given that the Riverpod app would be able to establish a constant Wi-Fi connection to the ESP-32, it is possible that Riverpod might be very beneficial. Additionally, Riverpod is able to provide assistance in the process of saving the data that arises from soil moisture sensors into the state objects. A further advantage of the Skia UI element implementation for graphics is that it is ideal for managing complex animations and efficiently displaying UI changes. This satisfies the direct requirements of displaying dynamic data for the garden, such as the levels of soil moisture, water levels, light status, or power consumption in real time. One example of a comprehensive and diverse plugin ecosystem that is available for Flutter is the `flutter_blue` plugin, which allows for Bluetooth Low Energy (BLE) connectivity. Another example is the `wifi_iot` plugin, which allows for Wi-Fi management and direct contact with ESP32. In the event that these plugins are unable to carry out the communication duties that are wanted between Flutter and the ESP32 module, there is a failsafe that may be implemented by directly incorporating platform-native code written in Java or Kotlin for Android and Swift or Objective-C for iOS. In this scenario, the direct contact with the ESP32 hardware would be achieved by Bluetooth Low Energy (BLE), Wi-Fi, or HTTP, depending on the software implementation.

Because of its adaptable user interface toolkit, Flutter's UI toolkit will be extremely useful in communicating real-time data to consumers because of its flexibility. The user interface capabilities that Flutter is able to offer bring it to the forefront of our mobile app development competition. These capabilities include the ability to create interactive graphs that represent soil moisture trends or water requirements of the system, as well as the ability to create custom alert widgets for low levels and sliders or toggles for adjusting lighting or watering.

3.6.4.2 Mobile App Storage

During the course of this project, there will be a few crucial occasions at which the data contained within the application will need to be preserved from one session to the next. As a means of accomplishing this, we examined the many possibilities.

Table 28: Basic App Storage Technology Comparison

	Cloud Sync	Local Storage
Sharing of Data	Sharing instantly	Challenging to distribute across devices
Support Offline	Only offline caching	Complete offline support
Access Speed	Depending on the internet connection, it might be slower.	Instant-from the memory of the device
Data Availability	Automatic and online device syncing	Sync cycles that are manually initiated or offline

Table 29: Flutter's Local Storage Options

	SQLite	Drift	Hive
Type of Storage	Relational Database (SQL)	Relational Database (SQL-based)	Key-Value Database
Performance	Fast and reliable	High performance, especially with caching	Extremely fast; works natively with Dart
Querying	Raw SQL queries with full SQL string support	SQL-like queries integrated with Dart syntax	Basic CRUD operations using key/object pairs in Dart
Best Used For	Standard SQL database tasks	Handling complex data relationships	Lightweight storage needs
Setup Difficulty	Moderate	Moderate setup required	Very simple

3.6.5 Custom PCB Design Software

Designing a custom PCB for GreenSync is essential for integrating sensors, power management, and control components into a compact and efficient layout. PCB design software determines how easily schematics, board layouts, and manufacturing files can be created while balancing complexity, cost, and collaboration. Below are the primary tools evaluated for this project.

1. KiCad

KiCad is a free, open-source PCB design suite widely used in academic and open-source communities. It supports advanced features like multi-layer layouts, real-time design rule checks, and 3D visualization.[117] Its large component libraries and zero licensing cost make it highly appealing for students and hobbyists. The trade-off is a moderate learning curve and lack of integrated manufacturing services, which can extend turnaround time.

2. EasyEDA

EasyEDA is a browser-based tool designed for simplicity and speed. It combines schematic capture, PCB layout, simulation, and direct ordering through services like JLCPCB. [118] Its intuitive interface and cloud-based collaboration make it ideal for rapid prototyping. While feature depth is less than high-end tools, its seamless link to manufacturing significantly reduces design-to-production time for projects like GreenSync.

3. Altium Designer

Altium Designer is the industry gold standard for PCB development, offering advanced routing, real-time collaboration, and powerful analysis tools. [119] Its capabilities make it indispensable for large, complex designs but excessive for small-scale IoT systems. High cost and a steep learning curve also make it impractical for GreenSync, though it serves as a reference for professional-grade features.

4. Autodesk Fusion 360 (Electronics)

Fusion 360 integrates PCB design with mechanical CAD, making it ideal when enclosure design and board layout must align perfectly. [120] This dual capability is valuable for GreenSync, where space optimization is critical. While its electronics features are not as advanced as Altium's, its 3D modeling and mechanical integration make it a strong contender for projects involving custom enclosures.

5. EAGLE (by Autodesk)

EAGLE has long been a favorite for small-to-medium PCB projects, offering schematic capture, PCB layout, and robust component libraries. [121] While

Autodesk has shifted development focus toward Fusion 360, EAGLE remains widely used due to its simplicity and community resources. However, its future viability may be limited as newer tools integrate more features.

Table 30: PCB Software

SOFTWARE	COST	FEATURES	EASE OF USE	COLLABORATION	PROS	CONS
KiCad	Free	Advanced (Multi-layer, 3D View, DRC)	Moderate	Limited (File-based)	Open-source; Professional-grade features; Large library support	No built-in manufacturing link; Higher learning curve
EasyEDA	Free (Basic) / Low (Pro)	Integrated design, simulation, and ordering	High	Excellent (Cloud-based)	Simple UI; Direct PCB ordering (JLCPCB); Great for quick prototyping	Limited advanced features; Internet-dependent
Altium Designer	Very High (Commercial License)	Full advanced toolset; Multi-board support	Low	Excellent (Team tools)	Industry standard; Powerful analysis and routing tools	Extremely expensive; Overkill for small projects
Fusion 360	Subscription	PCB + Mechanical CAD integration	Moderate	Good	Ideal for enclosure + PCB fit; 3D modeling included	Electronics module less advanced; Paid tool
EAGLE	Subscription (Limited Free Tier)	Schematic + PCB layout, good libraries	Moderate	Limited (Legacy)	Simple and widely supported; Great for small projects	Autodesk shifting focus to Fusion; Limited long-term support

Selection:

Eagle was selected as the PCB design tool for GreenSync due to its strong balance of ease-of-use, professional capabilities, and integration with popular platforms like Fusion 360. Eagle offers advanced features such as multi-layer schematic capture, layout editor, and real-time design rule checking, which are essential for ensuring precision in compact IoT hardware like GreenSync. Its user-friendly interface, extensive component libraries, and robust autorouting tools streamline the prototyping process while reducing manual errors. Although alternatives like KiCad provide open-source flexibility and Altium offers enterprise-tier power, Eagle's support for both beginners and experienced designers, along with its solid ecosystem and compatibility with mechanical design tools, make it a dependable and efficient choice for this project.

3.7 Environmental Design Factors

3.7.1 Enclosure and Housing Design

The enclosure is essential for protecting GreenSync's electronic components from environmental factors while ensuring accessibility and aesthetic appeal.

A well-designed housing allows proper ventilation, neat cable management, and easy access to key components, while enhancing durability and visual presentation. Below are the enclosure options considered for GreenSync:

1. 3D-Printed Enclosure (PLA/ABS)

3D printing offers full customization to fit the exact dimensions of the PCB, sensors, and wiring. [122] It allows precise cutouts for ports, ventilation, and buttons, making it ideal for prototyping and small-scale production. Materials like PLA or ABS are lightweight and affordable, though PLA can be less heat-resistant than ABS. Overall, this approach offers flexibility for design changes and a polished final appearance.

2. Acrylic or Polycarbonate Panels

Clear acrylic or polycarbonate sheets provide a modern, transparent design that showcases internal components, which is great for demonstrations and educational use. [123] These materials are lightweight and easy to clean but require precision cutting and assembly. While visually appealing, they are less impact-resistant than metal and may scratch over time.

3. Pre-Fabricated Project Box (ABS or Metal)

Pre-made enclosures are cost-effective, durable, and available in a wide range of sizes. [124] They reduce design time but require manual modifications such as drilling holes for sensors and ventilation. While they provide strong protection and a professional look, they offer limited flexibility compared to custom-designed options.

4. Wooden Casing

Wooden enclosures bring a natural, aesthetic appeal that blends well with indoor environments. [125] They are eco-friendly and easy to machine with basic tools. However, wood is susceptible to moisture and temperature changes, making it less suitable for variable environments or outdoor use.

5. Metal (Aluminum) Enclosure

Metal housings, especially aluminum, offer superior durability and excellent heat dissipation. [126] They are ideal for rugged applications where physical protection is a priority. However, metal adds weight and requires specialized tools for precise cutouts. It may also increase overall project cost compared to plastic-based solutions.

Table 31: Enclosure

ENCLOSURE TYPE	CUSTOMIZATION	DURABILITY	COST	AESTHETICS	PROS	CONS
3D-Printed (PLA/ABS)	Excellent	Moderate	Low	High	Fully customizable; Lightweight; Ideal for prototyping	Requires 3D printer; PLA less heat-resistant
Acrylic/Polycarbonate	Moderate	Moderate	Medium	High (Clear view)	Modern look; Lightweight; Easy to clean	Brittle under impact; Scratches easily
Pre-Fabricated Box	Low	High	Low–Medium	Moderate	Strong protection; No design time	Limited flexibility; Requires manual drilling
Wooden Casing	Moderate	Low–Moderate	Medium	High (Natural)	Eco-friendly; Easy to machine	Sensitive to moisture; Not ideal for fluctuating environments
Metal (Aluminum)	Low	Very High	High	Moderate	Excellent protection; Great heat dissipation	Heavy; Requires special tools; Higher cost

Selection:

The wooden enclosure was selected for GreenSync due to its eco-friendly nature, aesthetic appeal, and ease of fabrication with basic tools. Wood provides a natural look that blends well in indoor environments, making it an excellent choice for a consumer-focused product. Its ability to be customized with simple drilling and cutting adds flexibility for fitting components like sensors, buttons, and ventilation openings. While wood is not as durable as metal and can be sensitive to humidity, these limitations are manageable for GreenSync’s intended indoor use. The combination of sustainability, low cost, and visual appeal makes a wooden casing an ideal choice for this prototype.

3.7.2 Soil Type Considerations

Soil composition significantly impacts how plants absorb water, retain nutrients, and maintain aeration. In automated systems like GreenSync, understanding soil characteristics is critical for ensuring accurate moisture sensor readings and effective irrigation logic. Different soil types influence water retention and sensor calibration, and improper pairing can lead to under- or overwatering. Below are five common soil types evaluated for indoor plant setups:

1. All-Purpose Potting Mix

The most common choice for houseplants, all-purpose potting mix offers a balanced blend of nutrients, drainage, and aeration.[127] Typically composed of peat moss, perlite, and composted material, it provides consistency and stability, making it highly compatible with automated watering systems. Its uniform texture allows moisture sensors to deliver reliable readings, making this mix a preferred baseline option.

2. Garden Soil

While nutrient-rich, garden soil is generally too dense for container use. [128] Its tendency to compact restricts root aeration and creates unpredictable water retention patterns, which complicates automated irrigation. These variations can lead to inaccurate sensor readings, making garden soil unsuitable for GreenSync’s controlled watering approach.

3. Succulent and Cactus Mix

Designed for drought-tolerant plants, this mix emphasizes rapid drainage through added sand and perlite. [129] While ideal for succulents, its fast-drying nature requires recalibrating moisture thresholds and watering intervals in automated systems. Sensors placed in this mix detect quick moisture loss, demanding tighter control logic to prevent plant stress.

4. Peat-Based Mix

Peat-based soils retain water effectively, making them suitable for moisture-loving plants. [130] However, their high retention capacity increases the risk of overwatering in automated systems. Peat is also under environmental scrutiny due to sustainability concerns, prompting interest in eco-friendly alternatives like coco coir. Despite this, peat remains widely used in commercial potting blends.

5. Coco Coir Mix

Coco coir is an eco-friendly substitute for peat moss, offering excellent water retention and aeration without compaction issues. [131] Its fibrous texture supports consistent moisture distribution, making it highly compatible with automated irrigation. Gradual water release ensures stable conditions, reducing the risk of root rot and enabling accurate sensor performance.

Table 32: Soil Types

SOIL TYPE	WATER RETENTION	DRAINAGE	SENSOR CALIBRATION EASE	ENVIRONMENTAL IMPACT	PROS	CONS
All-Purpose Potting Mix	Moderate	Good	High	Moderate	Balanced for most plants; Consistent texture; Ideal for automation	May require occasional aeration to prevent compaction
Garden Soil	High	Poor	Low	Moderate	Nutrient-rich; Cost-effective	Compacts easily; Poor drainage; Inaccurate sensor readings
Succulent & Cactus Mix	Low	Excellent	Medium	Moderate	Great for drought-tolerant plants; Prevents root rot	Dries quickly; Requires frequent watering adjustments in automation
Peat-Based Mix	Very High	Poor-Moderate	Medium	Low Sustainability	Excellent for moisture-loving plants; Lightweight	Risk of overwatering; Environmental concerns with peat extraction
Coco Coir Mix	High	Good	High	High Sustainability	Eco-friendly; Good aeration; Consistent moisture distribution	Higher initial cost than peat; Requires nutrient supplementation

Selection:

All-Purpose Potting Mix was selected for GreenSync because it provides an ideal balance of water retention, drainage, and aeration, which is essential for maintaining healthy root systems in indoor plants. Its consistent texture ensures accurate and stable readings from moisture sensors, allowing the automated watering logic to function reliably without frequent recalibration. Unlike garden soil, which compacts and disrupts sensor accuracy, or specialty mixes like cactus soil that require frequent watering adjustments, all-purpose potting mix offers predictable performance suitable for a wide variety of plants. This makes it the most practical and versatile choice for an automated smart gardening system like GreenSync.

3.8 ⚡ Power Source

3.8.1 ⚙️ Power Supply

Powering an automated system like GreenSync requires a dependable and efficient energy source capable of supporting sensors, actuators, and wireless communication modules. The choice of power source affects the overall portability, scalability, and reliability of the system. Below are several common power supply options considered for projects of this type:

1. Wall Adapter (DC Power Supply)

Wall adapters are among the most reliable and widely used solutions for stationary embedded systems. They convert AC mains voltage into a stable DC output (typically 5V or 12V) suitable for powering microcontrollers and

peripherals. This method ensures uninterrupted operation as long as the system remains plugged in, making it ideal for indoor environments where mobility is not a concern.[16]

2. Rechargeable Li-Ion / Li-Po Battery Pack

Rechargeable lithium-ion and lithium-polymer battery packs offer excellent energy density and are commonly used in portable devices. For a project like GreenSync, these batteries can provide backup power during outages or enable temporary wireless operation. However, they require charging circuits, protection modules, and voltage regulation to ensure safe operation. [17]

3. AA Battery Pack (Alkaline or NiMH)

AA batteries remain a simple and low-cost power source for low-current applications. They are easy to source and replace, making them attractive for small-scale or educational projects. However, their limited capacity and inability to sustain high-load components like pumps or grow lights for long periods make them less practical for continuous operation.[18]

4. USB Power Source

Using USB as a power supply is common during prototyping because of its convenience and compatibility with microcontrollers like the ESP32. While USB provides a stable 5V output, its current limitations (typically up to 500mA on standard ports) can restrict powering additional hardware such as water pumps or LED grow lights. [19]

5. Solar Panel (with Battery Storage)

Solar power offers an eco-friendly and off-grid solution for plant care systems, especially for outdoor setups. When paired with an energy storage system like a rechargeable battery and a charge controller, solar panels can maintain continuous operation without reliance on grid electricity. However, their complexity and dependence on weather conditions make them better suited as an advanced feature rather than a core power source.[20]

Table 33: Power Supply

POWER SUPPLY TYPE	PORTABILITY	POWER CAPACITY	COMPLEXITY	SUITABILITY FOR GREENSYNC
Wall Adapter (DC Supply)	Low (stationary)	High (stable output)	Low	Ideal for indoor, continuous operation
Li-Ion / Li-Po Battery Pack	High	High (needs regulation)	Medium-High	Good for backup or temporary wireless use
AA Battery Pack (Alkaline/NiMH)	Medium	Low	Very Low	Simple but insufficient for long-term use
USB Power Source	Low	Low-Medium ($\leq 500\text{mA}$)	Very Low	Great for prototyping; limited for hardware
Solar Panel + Battery	High (off-grid)	Variable (weather-based)	High	Promising for outdoor use; not primary yet

Selection:

GreenSync will use a AA battery pack as its primary power supply, which will provide the ability to regulate to 3.3V and 5V, as needed. This configuration will ensure steady output to power the ESP32 microcontroller, AHT20 sensor, VEML770 sensor, I2C multiplexer, and ESP32 camera. This is practical as batteries are easy to source and provide flexibility.

GreenSync uses a wall adapter as a power supply, specifically an AC-to-DC converter paired with voltage regulation circuitry. This configuration ensures a reliable and steady output voltage necessary to power critical components such as the water pump and LED grow light. By converting standard AC mains power to a regulated DC output, the system avoids the complexity and limitations of battery management, while maintaining consistent indoor operation. Since GreenSync is designed for stationary deployment in environments like homes or greenhouses, this solution offers the best balance of stability, simplicity, and cost-effectiveness.

3.8.2 Wall Adapter

A wall adapter is a critical component in embedded systems like GreenSync, serving as the bridge between AC mains power and the DC voltages required by onboard electronics. A well-chosen adapter ensures stable power delivery, protects components from voltage spikes, and accommodates current demands from sensors, actuators, and communication modules. Below are five wall adapter options evaluated for the GreenSync system:

1. Alitove 12V 2A Power Supply

This adapter delivers 12 V at 2 A (24 W total) and is designed for broad compatibility with 5.5 mm x 2.1 mm barrel jacks. It operates with an input range of 100–240 V AC and includes overvoltage, overcurrent, and

short-circuit protection. This model is cost-effective and widely used in embedded and hobbyist applications for powering ESP32 boards and medium-load peripherals [152].

2. Mean Well GS25A12-P1J

A 12 V, 2.08 A (25 W) desktop-style adapter with high electrical efficiency and robust international certifications. This unit meets UL, CE, and FCC safety standards and offers better thermal management and electromagnetic interference suppression than generic models. It's a preferred choice for systems requiring continuous operation under moderate load, including pumps and grow lights [153].

3. UXCELL 9V 2A Adapter

This unit provides 9 V at 2 A and is a lightweight, budget-friendly option. Although 9 V is less common than 12 V in automation, it can be stepped down easily using voltage regulators to match 5 V or 3.3 V rails. It is suitable for systems with lower total power consumption or when separate regulation stages are already planned [154].

4. DROK Buck Module with 12V Adapter

This flexible option combines a 12 V adapter with an inline adjustable buck converter, enabling output voltages from 1.25 V to 36 V at up to 5 A. It features overcurrent and thermal protection and includes a digital display for fine adjustment. This setup is ideal when GreenSync needs to drive multiple subsystems at different voltages from one unified source [155].

5. HiLetgo 5V 2A USB Power Supply

Optimized for microcontrollers and logic-level circuits, this 5 V 2 A USB power brick offers simplicity and plug-and-play convenience. While it lacks the power to drive high-load peripherals like water pumps, it is ideal for powering the ESP32-WROOM or ESP32-CAM during isolated testing or when connected to minimal peripherals [156].

Table 34: Comparison of Power Supply Adapters

NAME	OUTPUT	CURRENT	VOLTAGE REGULATOR	COMPATIBILITY
HiLetgo 5V 2A Wall Adapter	5V DC	2A	Yes (direct 5V logic)	Common with ESP32; good for sensors only
ALITOVE 12V 2A Adapter	12V DC	2A	Yes (can be stepped down to 5V)	Better for powering pumps/relays
MEAN WELL GS25A12-P1J	12V DC	2.08A	Yes	High-quality industrial-grade; more durable
Uxcell AC 100-240V to DC 9V 2A	9V DC	2A	Yes (LDO/AMS1117 down to 5V)	Mid-point voltage, flexible
DROK Adjustable Buck Module + 12V Adapter	12V in → 1.25–12V out	3A	Integrated voltage regulation	For tunable multi-voltage systems

Selection:

For the GreenSync system, the Alitove 12 V 2 A adapter was selected as the most balanced choice in terms of power capacity, compatibility, and reliability. Its 24 W output comfortably exceeds the combined current requirements of the ESP32, ambient sensors, and low-to-moderate load peripherals, while still leaving headroom for future expansion such as small pumps or solenoid valves. The standardized 5.5 mm x 2.1 mm barrel connector ensures easy integration with both off-the-shelf modules and custom PCBs, reducing wiring complexity. With built-in overvoltage, overcurrent, and short-circuit protection, the Alitove unit adds a layer of safety against common electrical faults without driving up cost. Compared to more expensive industrial adapters, it provides dependable operation at a fraction of the price, making it the best fit for GreenSync’s prototype stage where affordability and practicality must align with system stability.

3.8.3 Voltage Regulation Modules

Maintaining a stable voltage supply is crucial for the proper operation of embedded systems, as fluctuations can cause erratic behavior or damage sensitive components. For a system like GreenSync, which includes sensors, actuators, and wireless communication modules, voltage regulation ensures consistent performance and safety. Below are three commonly used voltage regulation methods considered for this design:

1. Linear Regulators (e.g., LM7805, AMS1117)

Linear voltage regulators are one of the simplest and most cost-effective solutions for achieving a fixed output voltage such as 5V or 3.3V. They regulate voltage by dissipating excess energy as heat, which makes them inherently less efficient—particularly when the input voltage is significantly higher than the output. While they offer low noise and are ideal for low-power

circuits, their inefficiency and heat generation make them unsuitable for high-current loads like water pumps or grow lights. [21]

2. Buck Converter (Step-Down Switching Regulator)

Buck converters are highly efficient voltage regulation devices that use high-frequency switching to step down voltage with minimal energy loss. Unlike linear regulators, they do not waste power as heat, making them the preferred choice for components that draw higher currents, such as pumps or LED grow lights. Although they require additional components (such as inductors and capacitors) and introduce switching noise, their superior efficiency and thermal performance make them ideal for GreenSync's design. [22]

3. Boost Converter (Step-Up Switching Regulator)

Boost converters serve the opposite function of buck converters, stepping up a lower input voltage to a higher output voltage. This is particularly useful in battery-powered applications where the available voltage may be lower than required by system components. While highly efficient in such scenarios, boost converters are unnecessary for GreenSync's wall-adaptor-based design, which already provides a stable and adequate input voltage. [23]

4. Buck-Boost Converter

Buck-boost converters combine the functionality of both buck and boost regulators, enabling them to maintain a stable output even when the input voltage fluctuates above or below the desired level. This makes them valuable in battery-operated systems or outdoor setups where voltage variation is common. However, their increased complexity, higher cost, and lower efficiency compared to dedicated buck or boost converters make them unnecessary for GreenSync's current configuration. [24]

Table 35: Voltage Regulators

METHOD	FUNCTION	EFFICIENCY	NOISE	COMPLEXITY	SUITABILITY FOR GREENSYNC
Linear Regulator	Fixed voltage (e.g., 5V/3.3V)	Low (heat dissipation)	Very Low	Very Low	Good for low-power sensors; not pumps
Buck Converter	Step-down (V_i)	High	Moderate (switching)	Medium (external components)	Ideal for pumps and grow lights
Boost Converter	Step-up (V_i)	High	Moderate	Medium	Not needed; input voltage is sufficient
Buck-Boost Converter	V_i or V_i (flexible)	Moderate	Moderate	High	Overkill for current wall-powered design

Selection:

The Buck Converter was selected for GreenSync due to its superior efficiency and ability to step down the variable voltage supplied by a battery to stable levels required by the ESP32 microcontroller and other peripherals. In battery-powered systems, minimizing energy loss is critical for extending operational life, especially when powering high-current components like water pumps and LED grow lights. Buck converters achieve this through high-frequency switching, which significantly reduces power dissipation compared to linear regulators. This not only enhances energy efficiency but also reduces heat buildup, ensuring greater system reliability and longer runtime on a single battery charge. While they do require additional circuitry such as inductors and capacitors, the performance benefits far outweigh the design complexity, making buck converters the optimal choice for GreenSync's mobile, off-grid configuration.

Update:

For GreenSync's power architecture, two switching regulators were selected to provide stable and efficient voltage rails for the system: the **LM2679S-ADJ** for the 5V line and the **LM2576** for the 3.3V line. The LM2679S-ADJ was chosen due to its high current capability (up to 5A), adjustable output, and excellent efficiency at moderate loads, making it ideal for powering components such as the relay module, WS2812B LED strip, and auxiliary peripherals that require a robust 5V supply. Its adjustable output feature allowed the team to fine-tune the voltage during testing and ensure compatibility with varying demands. Likewise, the LM2576 was selected for the 3.3V rail because of its proven reliability, simplicity of implementation, and ability to deliver up to 3A—more than sufficient for the ESP32, sensors, and control electronics. Both regulators use widely available inductors and capacitors, support stable operation with straightforward PCB layout requirements, and are known for their thermal performance under continuous load. Together, the LM2679S-ADJ and LM2576 offered a balance of efficiency, cost-effectiveness, and ease of integration, making them the most practical pair of regulators for GreenSync's mixed-voltage design.

Chapter 4 Standards and Design Constraints

4.1 Standards

4.1.1 Safety and Health Concerns

There will be no methods, equipment, or techniques that involve extremely hazardous chemicals that will be dealt with in this project. The wiring between components, on the other hand, is a clear area of concern with regard to safety. This includes the connection between the AC to DC converter, the connection between the AC to DC converter and the water pump, the connection between the AC to DC converter and the LED light strips, and any other connection that is responsible for the transfer of current between wired components.

When it comes to preventing electrical risks like short circuits, overheating, or inadvertent exposure to live wires, proper insulation and secure connections are very necessary. In addition, strain relief measures must be put into place in order to guarantee that wires are not pulled or detached by accident when the system is functioning normally. Protecting components from the possibility of overcurrent circumstances can be accomplished by the incorporation of fuses or circuit breakers, which can further increase safety. In spite of the fact that the system runs at relatively low DC voltages after conversion, the original AC input still offers a potential risk. Because of this, it is vital for the system to be handled carefully and for electrical components to be enclosed appropriately in order to ensure safe operation.

4.1.1.1 Adherence to the Standards Established for Health and Safety

A wide range of distinct safety requirements have been observed and adhered to during the course of this project. Particularly with regard to the wiring of the water pump, the International Electrotechnical Commission (IEC) 60335-2-51 [132]: stationary circulation pumps for heating and service water installations and IEC 60335-2-55 [133]: electrical appliances for use with aquariums and garden ponds will be adhered to in a meticulous manner. This is especially true with regard to the restrictions on the bending of the wiring near the wall outlet. In addition, all plug connections will be made in line with the plug standards established by the National Electrical Manufacturers

Association (NEMA), specifically NEMA 5-15 [134]. This is done to guarantee that the plug that is used for any given current carrying application is of the proper size. When it comes to the pressure ratings of the pipes and tubing, the American Society for Testing and Materials (ATSM) requirements that are mentioned in ASTM D1785 are strictly adhered to for the ¼" schedule 40 PVC pipe that is being utilized for the water system [135].

4.1.1.2 The Electrical Safety Standards

The dependability, user safety, and durability of the GreenSync system are all greatly impacted by the electrical safety regulations that are in place. IEC 60529 provides a description of ingress protection (IP) ratings, which enable outdoor devices [136] to be protected against elements such as water, dust, insects, and plant debris that are typically found in garden settings. In order to reduce the likelihood of hazards like electric shocks and fires, the National Electrical Code (NFPA 70) [137] is strictly adhered to while planning wiring. ISO 50001 [138] encourages the effective control of energy consumption, which is especially advantageous given that the system makes use of renewable energy sources such as solar and battery power.

4.1.1.3 Standards for the Design of PCBs

Standardization of the design of printed circuit boards (PCBs) establishes the criteria that must be met in order to guarantee that a PCB can be properly manufactured and functionally utilized as intended. Frequently, the producer will specify certain criteria for the product. As was indicated before, the manufacturer that has been chosen for this project is JLCPCB. The design process will be guided by their specific criteria, and the specific standard that will be employed in relation to the PCB design will be different for each firm.

For instance:

- The holes that are drilled in the printed circuit board (PCB) should have a range of 0.3 mm to 6.3 mm, with the minimum size of the via holes being 0.3 mm and the pad diameter being 0.5 mm.
- Vias are required to be positioned at a minimum distance of 0.254 millimeters from other vias and traces, regardless of whether or not they are net.
- A minimum clearance of 0.3 millimeters is needed between any trace and the edge of the board.

- The lowest permitted trace width is 0.127 millimeters, and traces must maintain a spacing of 0.127 millimeters from one another.
- When it comes to 2-layer boards, the options for the thickness of the outer copper layer are restricted to either 1 ounce or 2 ounces.
- The minimum spacing requirements for different nets include a distance of 0.5 millimeters between the holes and pads.
- It is recommended that pads have a minimum clearance of 0.127 millimeters from pads on different nets, and only 0.2 millimeters from traces.

4.1.1.3.1 PCB Standard IPC-2221

For the purpose of creating printed circuit boards that are long-lasting, functional, and acceptable for manufacturing, the IPC-2221 standard acts as a basic guideline [139]. It covers a wide variety of subjects with the goal of improving the performance of printed circuit boards (PCBs) as well as their dependability. In the standard, the following important aspects are brought to the forefront:

1. Management of the Temperature:

For the purpose of preventing component failure and ensuring long-term dependability, thermal management is necessary. Effective heat dissipation is particularly important. In order to effectively regulate temperature distribution, the International Power Control (IPC) 2221 gives suggestions about the utilization of thermal vias, the selection of suitable board materials, and the positioning of heat-generating components.

Design for the purpose of manufacturing:

The International Product Standards (IPC) 2221 places an emphasis on simplifying the manufacturing process by making design decisions that improve fabrication, assembly, and testing practices. During the production process, adhering to DFM procedures helps decrease costs, boost yield, and eliminate mistakes.

2. Routing of Traces:

The standard provides comprehensive instructions for planning the arrangement of traces, with a particular emphasis on maintaining

signal integrity, controlling impedance, and reducing electromagnetic interference (EMI). It provides procedures for the routing of high-speed communications in order to lessen the effects of crosstalk and signal degradation.

Placement of Components:

An appropriate arrangement of components improves both the electrical performance and the thermal balance of a system while simultaneously simplifying assembly. Through the use of strategic layout, the standard provides instructions on how to prevent interference, improve access for testing, and increase the dependability of the system.

The Selection of Materials:

IPC-2221 specifies criteria for selecting board materials based on thermal conductivity, dielectric constant, and mechanical robustness. These criteria are used to select materials for boards. The selection of the material has a direct influence on the quality of the signal transmission and the resilience to external stress.

3. Guidelines for the Design:

Among the fundamental factors that are defined by the standard are the minimum trace width, clearance, and layer stack-up criteria. The presence of these guarantees that the board is able to meet power needs, prevent short circuits, and keep its general functioning intact.

4. Reliability and Quality:

IPC-2221 offers suggestions for inspection techniques, testing protocols, and quality assurance processes that help uncover faults and certify conformity with industry benchmarks. These recommendations are included at the conclusion of the document to guarantee that consistent performance is achieved.

5. Considerations Regarding the Environment:

As an additional recommendation, IPC-2221 advocates design techniques that take into account the operating context and the influence on the environment. This involves making certain that the material is compatible with severe circumstances such as fluctuations

in temperature and humidity, as well as vibrations caused by mechanical motion. When environmental resilience is taken into consideration throughout the design process, it results in longer product life cycles and lower failure rates in the field.

6. The integration of electromechanical systems:

Through the detailing of mechanical limitations, such as mounting hole location, board outline tolerances, and connection alignment, the standard provides assistance for integration with mechanical components and housings. By ensuring that the printed circuit board (PCB) fits seamlessly into enclosures and interacts appropriately with other system parts, proper electromechanical design strengthens the overall resilience of the system and makes it more compatible with assembly.

These ideas are included into IPC-2221, which acts as a sturdy foundation for the creation of high-performance printed circuit boards (PCBs) that are capable of meeting the stringent criteria of current electronic systems. Our design will be guided by this standard, which, when paired with the limits that are specific to the manufacturer, will assure quality, efficiency, and the ability to be manufactured. These factors that are taken into account within the standard are summarized in the table that follows:

Table 36: List of Categories and Descriptions for the IPC-2221 Specification

Classifications of the Standard Specifications	Description
Management of the Temperature	Thermal vias and layout approaches are recommended to reduce overheating and maintain long-term dependability.
Design for the purpose of manufacturing	Promotes layout methods that ease fabrication, assembly, and testing, reducing mistakes and manufacturing costs.
Routeing of Traces	Facilitates layout methods that ease fabrication, assembly, and testing,

	reducing mistakes and manufacturing costs.
Placement of Components	Specifies signal integrity, EMI mitigation, and impedance control for high-speed signal routes.
The Selection of Materials	Improves signal flow, thermal performance, and testing and assembly accessibility by guiding part location.
Guidelines for the Design	Choose materials based on electrical, thermal, and mechanical qualities for performance and environmental demands.
Reliability and Quality	Supports power and signal needs by defining fundamental PCB features such trace width, spacing, and layer configuration.
Considerations Regarding the Environment	Specifies inspection and testing methods to assure board durability and performance.
The integration of electromechanical systems	Ensures system-level integration by recommending board fit, mounting holes, and interface alignment.

4.1.1.4 Standards for Wireless Communication and Connectivity

The successful operation of the GreenSync. system is contingent upon the components, which include soil moisture sensors, microcontroller units (MCUs), actuators, and the user interface on a mobile application, communicating with one another in a seamless manner. The compliance with wireless communication standards guarantees the dependability and responsiveness of this contact. Bluetooth Low Energy (BLE) is employed for communication that is both short-range and energy-efficient, notably between sensors and the modular computer unit (MCU). Real-time data monitoring and system control are made possible by the implementation of IEEE 802.11 (Wi-Fi), which allows for high-speed and robust connectivity to and from the

cloud or mobile application environments [140][141]. Furthermore, the MQTT protocol, which is standardized under ISO/IEC 20922, facilitates the transfer of data between devices and the cloud in a lightweight and efficient manner, which is very important for Internet of Things-based gardening systems [142]. The use of these standards guarantees the establishment of a communication system that is not only quick and scalable, but also strong enough to manage a variety of smart gardening activities, including logging, automation, and real-time notifications.

4.1.1.4.1 Wireless Comm Standard IEEE 802.11

In the context of this project, wireless communication is an essential component since it enables the exchange of data between the system and electronic equipment that are external to it. This communication is made possible by the IEEE 802.11 standard, which provides a set of protocols that enable compatibility across a range of wireless devices. This standard is crucial in making this communication possible. By defining fundamental characteristics like as frequency ranges, modulation methods, and data transmission protocols, this standard sets the platform for devices to share data in a unified manner. It does this by laying the foundation for the technology.

The specification of frequency bands that are utilized for communication is one of the most important aspects of the IEEE 802.11 standard. The bands operating at 2.4GHz and 5GHz are two of the most frequently utilized frequencies. The 2.4GHz band is particularly useful since it provides a larger range and better penetration through barriers such as walls. As a result, it is perfect for usage in contexts such as the house and interior spaces, where devices may need to communicate over longer distances. On the other hand, the 5GHz band is capable of supporting quicker data transmission speeds; however, its range is more restricted, and it has a greater difficulty dealing with obstructions than the 2.4GHz band does. The standardization of these frequency bands makes it possible for devices to quickly identify and tune into the right frequency, hence reducing the amount of interference that is caused by other devices that carry out their operations on other frequencies.

The IEEE 802.11 standard also features sophisticated data transmission techniques, such as Orthogonal Frequency Division Multiplexing (OFDM), which splits the data into numerous smaller streams in order to optimize the speed at which it can be transferred. In addition, Direct Sequence Spread Spectrum (DSSS) is utilized in order to broaden the bandwidth and decrease

interference. Wireless communication is needed for reliable data transfer in dynamic contexts, and these technologies increase the robustness and efficiency of wireless communication, which is crucial for achieving this.

There are several variants of the IEEE 802.11 family, each of which offers a unique set of benefits that are determined by the requirements of the application. For this particular project, the microcontroller that has been chosen is the ESP32, which is capable of supporting 802.11b, 802.11g, and 802.11n. 11 megabits per second is the maximum speed that can be achieved with the 802.11b version, which works on the 2.4GHz band. In spite of the fact that it is adequate for fundamental data transmission tasks such as streaming video, it is much slower than previous versions. Despite the fact that it operates on the 2.4GHz range, the 802.11g standard incorporates OFDM, which allows it to achieve data speeds of up to 54 megabits per second. As a result of this enhancement, the video quality may be improved, and the data transfer can be accelerated, providing a satisfactory balance between speed and range.

It is possible to use the 802.11n version for applications that require greater speeds since it runs over both the 2.4GHz and 5GHz bands. This provides flexibility in terms of both coverage and speed. Within the context of this project, the ESP32 makes use of the 2.4GHz band of the 802.11n standard, which is capable of achieving rates of up to 150 megabits per second. Due to the fact that it guarantees efficient communication between the moisture sensor boards, the main processing unit, and the associated mobile app, this is more than sufficient for the requirements of this project.

Incorporating IEEE 802.11 assures interoperability with a wide variety of devices and provides a smooth integration into networks that are already in place. Due to the fact that it adheres to a well-established standard, it ensures that communication is both efficient and dependable, making it an absolutely perfect option for this project. Furthermore, it simplifies the work of ensuring that both the data supplied from the sensor devices and the data received by the system are both comprehensible and interoperable, which makes it easier for users to interact with the final product in a fluid manner.

4.1.1.4.2 ESP-NOW Wireless Communication

At Espressif Systems, the company that is responsible for the widely used ESP32 and ESP8266 microcontroller families, a unique wireless communication protocol known as ESP-NOW was developed. Direct device-to-device data transfer is made possible by ESP-NOW, which does not

require the use of a conventional Wi-Fi network or router. This was designed to be an efficient, low-power, peer-to-peer communication method. It is the intention of this project to make use of this communication protocol in order to transmit the moisture sensing data from an ESP32 that is collecting the data to an ESP32 that is located in the central location. This protocol has become an essential component of Espressif's wireless stack, particularly for situations that call for low-latency communication and minimal infrastructure.

The 2.4 GHz ISM band is the same unlicensed frequency range that Wi-Fi and Bluetooth technologies employ. This is where ESP-NOW's main functionality is located. Wi-Fi, on the other hand, is dependent on the IEEE 802.11 association mechanism, but ESP-NOW does not. For the purpose of exchanging data between devices, it employs a simplified form of 802.11 frame transmission, which enables communication that is both quick and lightweight. Without the overhead that is generally associated with Wi-Fi authentication and IP-based protocols, devices that use ESP-NOW are able to transmit and receive messages in a manner that was nearly immediate. It is specifically quicker than other transmission techniques because ESP-NOW lowers the five levels of the OSI layer model to just the data connection and physical layers. This is the case since ESP-NOW is a specific reduction.

A broadcast and peer-to-peer communication paradigm is utilized by ESP-NOW, which sets it apart from other similar systems. The MAC address of each device, which is referred to as a peer, is used to identify the device. When using ESP32 devices, communication can take place between as many as twenty peers. Although it is not necessary for these peers to be linked to a Wi-Fi network, it is possible for ESP-NOW and Wi-Fi to coexist on the same chip, which enables hybrid use cases to be implemented. Those applications that require many embedded devices to synchronize or exchange data without requiring internet connectivity or a centralized server are ideal candidates for the protocol since it is well-suited for those applications.

With the use of the 802.11 Action frame subtype, ESP-NOW is able to send data in either encrypted or unencrypted formats, depending on the specific technological requirements. As a result of the payload size being restricted to 250 bytes per message, it is more appropriate for relatively brief bursts of data than for communication that requires a large bandwidth. Naturally, for the purpose of this project, the data pertaining to moisture sensing can be readily wrapped in just a single byte. Therefore, 250 is more than plenty to communicate the data, and size is not at all a worry in this particular scenario. This packet format offers a reasonable compromise between complexity and

performance, making it suitable for applications that require low-latency and predictable behavior. Some examples of such applications are sensor networks, device orchestration, and basic automation and control systems.

ESP-NOW is a protocol stack and operating model that is significantly different from Wi-Fi and Bluetooth, despite the fact that it operates on the same frequency range as both of these technologies. The ESP-NOW protocol, in contrast to Bluetooth, provides a greater range and quicker transfer rates without the need for any pairing or discovery processes to be carried out. The fact that ESP-NOW significantly reduces the amount of time required for the devices to handshake during the Bluetooth pairing process makes this a particularly crucial point to keep in mind. The ESP-NOW protocol, in contrast to the Zigbee protocol, does not rely on a mesh network architecture and does not require a coordinator or gateway. The differences between the two protocols underscore its particular function as a lightweight, rapid-deployment protocol that is best suited for contexts that require point-to-point or small-group communication without requiring a significant amount of network overhead.

ESP-NOW's status as a proprietary product is a significant source of differentiation for the company. It is not a standard that is under the control of a formal standards body like the IEEE or the Bluetooth Special Interest Group (SIG). As an alternative, Espressif Systems is responsible for its whole development, maintenance, and support. While this does make it possible for Espressif to have a deep integration with its own hardware platforms, it also means that ESP-NOW is not generally compatible with microcontrollers or chips that are manufactured by other companies. It is beyond the range of official industry standards due to the fact that ESP-NOW is a proprietary product, despite the fact that it plays a significant role within the ecosystem of ESP-NOW development.

Even though it has some limits that are unique, ESP-NOW is in compliance with the basic regulatory criteria that are imposed on devices that operate in the 2.4 GHz ISM band. In the United States, it is governed by the Federal Communications Commission (FCC), and in Europe, it is governed by the European Telecommunications Standards Institute (ETSI). In a manner analogous to that of Wi-Fi and Bluetooth devices, this means that devices that utilize ESP-NOW may be deployed in the majority of locations without the need for specialist licensing. With that being said, its utilization is inextricably linked to Espressif hardware, which limits its usability in more generalized scenarios that involve several vendors.

As a conclusion, ESP-NOW is a wireless communication protocol that is very effective yet proprietary. It was built to suit certain needs for performance and implementation. By providing ultra-lightweight peer-to-peer communication with little latency and low power consumption, it makes a significant contribution to the wireless landscape and fulfills a function that is unique to it. Despite the fact that it is not a formal standard, it reflects Espressif's strategic ambition to deliver powerful communication capabilities that are designed for embedded applications within its hardware platform. Because of this, ESP-NOW is best understood as a communication protocol that was created specifically for a specific purpose, rather than as a general-purpose industry standard.

4.1.1.5 Standards for Data Security and Privacy

In light of the fact that smart gardening systems capture, transmit, and retain sensitive data—including environmental conditions, watering schedules, and user preferences—it is essential to have data security and privacy guidelines in place. The International Organization for Standardization (ISO) 27001 offers a framework for managing information security [143]. This framework ensures that the data that is gathered is encrypted, kept securely, and safeguarded against unwanted access. This is of utmost significance in situations when the system links to cloud platforms or mobile apps respectively. Furthermore, the National Institute of Standards and Technology Cybersecurity Framework for the Internet of Things provides actionable information to improve the security of devices. This guidance covers topics such as secure firmware upgrades, authentication mechanisms, and network safeguards [144]. In addition to safeguarding user information, the GreenSync system also fosters trust and resiliency within the smart gardening experience. This is accomplished through the implementation of these standards.

4.1.1.6 Standards for Smart Lighting Components

When it comes to fostering the growth of plants and preserving the equilibrium of the environment, smart lighting is an essential component, particularly in places that are shaded or indoor gardens. The system adheres to the International Electrotechnical Commission (IEC) 60598, which specifies the standards for the safe functioning of lighting fixtures that are designed to withstand the elements and are used in outdoor settings [145]. This guarantees that the lights are able to withstand different temperatures, as well as exposure to UV radiation and moisture. Light sensors and controllers that

are created in accordance with ISO 9241-303 are utilized by the system in order to achieve optimal energy utilization and real-time responsiveness [146]. These controllers and sensors are responsible for guiding the performance of visual displays and ensuring precise brightness adjustment based on their surroundings. Observing these lighting standards encourages the development of healthy plants, cuts down on the use of energy that is not essential, and guarantees that the system will be able to function in a secure and dependable manner throughout the course of time.

4.1.1.7 Standards for Sustainability and Energy Efficiency

Energy efficiency criteria are the driving force behind the GreenSync system's environmentally responsible functioning, which is centered on the concept of sustainability. The capacity of the system to monitor and minimize energy usage across all key components, including pumps, sensors, and lighting systems, is supported by ISO 50001 [137]. This helps to ensure that power is utilized and consumed in an efficient manner. Solar power or hardware that is efficient in terms of energy consumption can be included into the system to lessen its reliance on external energy sources. Furthermore, the EPA WaterSense recommendations, which encourage the use of water-saving technology to limit waste while supporting healthy plant development, may be aligned with irrigation efficiency in order to enhance irrigation efficiency [147]. A gardening solution that is both environmentally sensitive and cost-effective is promoted by these standards collectively. This solution reduces the amount of resources that are consumed without affecting the performance of the garden.

4.1.1.8 Standards for Sensors and the Environment

To ensure the success of any intelligent gardening system, it is essential to exercise precision and accuracy when monitoring the surrounding environment. For the purpose of ensuring the gathering of accurate data from soil, water, air, and light conditions, the GreenSync system adheres to environmental and sensor standards. Both ISO 18400 and ISO 11074 are standards that specify procedures for evaluating and monitoring soil quality [148][149]. These methodologies provide direction for the positioning and calibration of sensors that measure moisture, pH, and nutrients. In order to provide intelligent irrigation scheduling, ISO 772 assures that water flow rates and efficiency are measured in a consistent manner [150]. This is helpful for monitoring water use. Components of air quality, such as temperature and carbon monoxide sensors, contribute to the maintenance of a consistent

microclimate, which is essential for the development of plants. Through adherence to these criteria, the GreenSync system is able to provide precise data on the environment, which in turn enables intelligent automation and contributes to the development of a more robust garden ecosystem.

4.2 Constraints

4.2.1 Constraints of Sustainability

4.2.1.1 Battery Longevity

This project has a number of significant restrictions, one of the most significant of which being environmental impact caused by powered gadgets. Because the objective of this project is to provide the user with hands-off automation as soon as the project is handed over to them, it is very undesirable for the design to require the replacement of power supplies such as batteries. For this reason, the objective was to avoid utilizing batteries wherever it was practicable and reasonable to do so in the design. This was accomplished by extracting 120V AC power from the wall. The water pump was one instance in which this point was brought up. As an illustration, the device might be simply powered by attaching it to a 12V lead acid battery. This would be applicable to any water pump that operates at 12V. However, despite the fact that this is a workable solution, the battery will ultimately lose its charge over the course of time, and if this happens, there are many limits that might become an issue. As a first and foremost matter, the fact that the pump battery is dying and there is no means to recharge it renders the idea intrinsically unacceptable. Furthermore, a safety risk is produced as a result of the fact that the user may attempt to connect a battery charger to the device. If this connection is not secured by a fuse or breaker, it may result in short circuit circumstances, which almost invariably lead to damage to property or even worse. In light of these considerations, it was decided that, if it was feasible to do so, the devices would be powered either directly by 120V AC or by the AC to DC converter, which is powered by 120V AC from the wall directly.

On the other hand, there are certain hardware components that cannot be separated from batteries due to their practicality. One of the cases is the mobile device that is used to execute the application; however, this is not a significant cause for concern because mobile charging is standardized and an extremely safe practice in every respect. On the PCB for the Moisture Sensor and the Main MCU are the batteries that are the main cause for worry. This is

because these batteries are more difficult to access, and in order to make the design as efficient as possible for reasons related to sustainability, the devices are designed to have the lowest possible power consumption in order to guarantee a long system life. B cell "coin" batteries, which will be used for the PCBs of the moisture sensor, adhere to IEC 61960 REV, which is a standard that describes the battery life that can be anticipated from such batteries under specific operating circumstances [151]. Both the power consumption and the charge of the battery will be monitored throughout the testing phase of the design. This will make it possible to test the battery to guarantee that it satisfies the requirements of the standard.

4.2.2 Optimizing Battery Longevity

The batteries that are going to be used in this project will not be exposed to temperatures that are significantly higher than normal. However, the coin batteries that are going to be linked to the PCB that is functioning as a moisture sensor will be exposed to water since it will be stuck in the soil, which will then be sprayed with water. Nevertheless, this specific battery type was selected because of this particular reason. When it comes to wet situations, coin batteries with a model name beginning with "CR" utilize lithium chemistry to build their voltage. This makes them significantly more suitable than alkaline-based batteries for usage in these settings. By using coin batteries that are based on lithium, they will be able to keep a sophisticated appearance with a limited amount of surface area that is exposed to water, while yet providing sufficient power to operate the moisture sensors.

4.2.3 Reliability and Maintenance Constraints

4.2.3.1 The process of corrosion

During the course of our investigation into moisture sensors, we came across the issue that a significant number of customers have expressed their dissatisfaction with the corrosive properties of various types of sensors, including direct contact electrode sensors. It would appear that the majority of the sensors that are constructed in this manner are only utilized for a single instance or for a little length of time. When it comes to the moisture sensor that we are going to use for our plants, we strive to prevent corrosion to the greatest extent feasible. This is because of the reasons stated above.

Electrolytic and galvanic corrosion are the two forms of corrosion that occur most frequently. For the purpose of this project, it is essential to have a solid understanding of both of these forms of corrosion because they are extremely similar to one another. When two different metals come into touch with one other and an electrical current is delivered between them, a phenomenon known as galvanic corrosion takes place. This is something that may occur on a printed circuit board (PCB) between the metal of the component and the metal from the trace that comes into touch with each other. After moisture has come into touch with this region on the printed circuit board (PCB), the electrical current will cause the metal that is more resistant and weaker to corrode more quickly. The result is that the voltage measurements between the components are inaccurate, and the components themselves deteriorate as a result.

Electrolysis and the chemical composition of the soil are two of the most prevalent causes of corrosion. The phenomenon known as electrolysis occurs when two distinct kinds of metal come into contact with water together. It is possible for electrolytes to attach to each of the two different metals, and it is also possible for a current to flow between the two metals because of the different metals. Consequently, this makes it possible for oxidation to take place within the metal, which will ultimately result in the metal gradually corroding away. Considering how acidic the water is, soil chemistry is another significant component of corrosion that must be considered. On the other hand, if the water in the soil is more acidic, then it has a greater number of H^+ proton ions. The result of this will be an increase in the number of electrons that are interested in pairing with the solitary hydrogen ions. Corrosion will take place at a quicker rate in proportion to the number of electrons that are escaping from the sensor and entering the soil.

4.2.3.1.1 Erosion caused by electrolysis

The process of electrolytic corrosion takes place when an electrolyte, which is a conducting liquid, comes into contact with an anode and a cathode. Slivers of metal are able to pass through the electrolyte and come into touch with the other metal as a result of this. In the event that this is not addressed in the appropriate manner, the metal will begin to deteriorate over an extended period of time.

When an electrical current is present in a printed circuit board (PCB), a metal oxide will begin to develop between two traces if there is water present between the traces. This will occur if there is water present between the

holes. This can result in a short circuit occurring in the printed circuit board (PCB), and if the metal oxide layer accumulates, this short circuit may become permanent over time. Any short circuit that happens in the circuit has the potential to result in undesired outcomes as well as complete failure of the circuit. We are going to make every effort to reduce the occurrence of this phenomena as much as we can because it is not ideal for any circuit.

4.2.3.1.2 Corrosion caused by galvanic current

The phenomenon known as galvanic corrosion takes place when two metals that are significantly different from one another come into contact with one another while an electrical signal is being transmitted through them in the presence of an electrolyte, which is often water. In the course of this process, one of the metals, known as the cathode, is more powerful than the other, known as the anode. Due to the fact that the weaker metal has a higher resistance, there is a greater amount of current that is transported through that metal. This effect will cause the anode to corrode more quickly, but the cathode will not be affected in any way by the corrosive phenomenon.

In the event that water comes into contact with the trace and the soldering connection on a printed circuit board (PCB), galvanic corrosion may take place. A conductive route will be created between the two metals as a result of this, and the anode will be the one to benefit from the electrical current. As a result, the anode on the printed circuit board (PCB) will degrade at a considerably quicker rate than the cathode. This occurs due to the fact that solder frequently uses tin, but the trace on the printed circuit board (PCB) is constructed of copper. As a result of the different nature of these two metals, galvanic corrosion may take place. Over time, these connections on the printed circuit board (PCB) between the component and the trace will deteriorate to the point that a signal that is either too weak or not conveyed at all will be sent, which will ultimately result in the failure of the component.

The single-point resistive moisture sensors are the sites where this particular sort of corrosion is most frequently discovered. This is due to the fact that these sensors have been manufactured at a low cost and make use of metals of a lesser quality. There have been several reports of corrosion occurring with these sensors, and the degradation ultimately results in the failure of the sensor.

Table 37: Comparison of Electrochemical and Galvanic Deterioration

	Electrochemical Deterioration	Galvanic Deterioration
Causes	Electric current flowing through metallic material within an electrolytic solution	Physical connection of dissimilar metals immersed in an electrolytic medium
Conditions	Presence of wandering or leakage currents	Conductive solution creating electrical potential between dissimilar metals
Contributing Elements	Electrolytic conductivity, current intensity, and metallic composition	Metal compatibility and electrolyte conductivity
Outcomes	Rapid degradation of metallic structure	Progressive deterioration of the less noble metal
Mitigation	Minimize leakage currents, apply protective barriers, widen spacing of conductive paths	Install cathodic protection, apply insulating coatings, use compatible metals

4.2.3.1.3 Preventing Corrosion

In the process of attempting to fix corrosion, there are a wide variety of different sorts of remedies. Capacitive sensors, as opposed to resistive sensors, should be used as the first recommendation. Because the sensor does not require direct contact with the electrodes in the sensor and the water in the soil, capacitor sensors are sometimes referred to as corrosion resistant sensors. This is due to the fact that the sensor does not require an electrical connection. Through the utilization of a PCB soil moisture design, we are able to select the portion of the PCB that we wish to insert into the soil in order to determine the effect of the change in capacitance. We can determine the amount of water that is present in the soil by observing how fast or slowly the capacitor charges back up to its full capacity.

In spite of the fact that a printed circuit board (PCB) is mostly composed of copper, it is possible to apply a coating, such as gold or a gold-nickel alloy,

over the traces in order to avoid corrosion. However, this may be an expensive endeavor. Another method for preventing corrosion on the printed circuit board (PCB) is to use solder masks, epoxy, or aerosol spray. Adding a case around the exposed components is yet another method that we may use to circumvent any corrosion that may occur on the printed circuit board (PCB) sensor. The electrical components would be protected from any stray water that could get onto them thanks to this. If you make sure that all of the components are dry, there will be no standing water on any of the metal components, which would prevent them from corroding.

We can also keep our components dry by making a three-dimensional model of the case for the printed circuit board sensor and then printing it out with a three-dimensional printer. It would be possible for us to prevent water from getting directly onto the components of the sensor if we constructed a casing around the components. Increasing the lifetime of the printed circuit board (PCB) would be of tremendous assistance to us.

4.2.3.2 The presence of salt

It is possible for the salinity of the soil to be one of the most significant elements that might influence the readings of a volumetric sensor. When discussing our project, the term "salinity" refers to the amount of salt that is present in the soil we are working with. A research project that was carried out by the National Library of Medicine investigated the ways in which salinity influences the output values of several sensors. According to the findings of their investigation, the majority of the moisture sensors that they examined incorrectly calculated the real moisture level in the plant when there was an excessive amount of salt present in the soil.

Because of the connection that exists between the salt of the soil and the electrical conductivity of the soil, this phenomenon takes place. According to the research that was presented earlier by the National Library of Medicine, the conductivity of the soil experiences a rise in proportion to the salinity of the soil. Electrons will be able to flow in a manner that is different from what is anticipated as a result of the rise in salinity. In the majority of cases, this effect is concerned with sensors that make use of capacitance-based readings. Under typical circumstances, the dielectric constant of the soil serves as the foundation for the observations that these capacitance moisture sensors take.

When it comes to moisture sensors that are based on capacitors, the capacitor is being discharged and is now waiting to be charged up again. Due to the fact that the dielectric material that the voltage is going through is now

a more conductive medium, the amount of time that the capacitor is waiting to be charged again is now altered as a result of an increase in salinity. Due to the fact that the capacitor is now charging at a higher rate, this is causing the measurements to be inaccurate. It is possible to circumvent this problem by calibrating the sensor at a high salinity; but, in order to do so, you would need to maintain a high soil salinity level, which can be detrimental to the presence of many plant species.

Therefore, it is essential for us to take into consideration the salt level of the soil in order to guarantee that the measurements we obtain are accurate. In the event that we are having difficulty obtaining precise measurements for our project, this is one of the ways in which we may investigate and find solutions. So that we may do this, we can alter the soil that the plant is now growing in.

4.2.4 Constraints due to the Environment

The environmental limits are one facet of this project that can be simple to ignore if attention is not paid to them. It could appear that the environmental aspect of a project that involves growing plants indoors is the least of the concerns, but this is not necessarily the case under all circumstances. For instance, invading plant species have historically been responsible for billions of dollars' worth of damage. The United States Forest Service estimated that the yearly cost of controlling invasive species in the United States was five hundred million dollars as early as the year 2000, and this cost has continued to rise each and every year since then. When this is taken into consideration, the design is meant to allow the user to cultivate and develop only plants that are natural to the region in which they live, or non-native species that do not represent a threat to the populations of native plants. Nevertheless, despite the fact that the user is in charge of this danger, it is still an essential limitation to take into consideration when developing a technology that has the potential to make this improper behavior more convenient. When it comes to the species that they decide to cultivate in this automated indoor garden, it is strongly recommended that all users adhere to the local wildlife and forestry rules.

4.2.5 Limitations on the Security

One more significant limitation to take into account is the issue of security and vulnerability, particularly in relation to the wireless communication that takes place between the various devices involved in this project. More specifically, the communication that takes place between the phone and the main ESP32,

as well as the communication that takes place between the ESP32s that are located on the moisture boards and the main ESP32. For every application in which data is exchanged and can be snooped upon, it is the responsibility of the design party to make the vulnerabilities known to the user in a clear and concise manner and to make an effort to address them to the greatest extent that is reasonably practicable. The information that is being exchanged for this project is completely harmless in every sense of the word. It is either the values from the moisture sensor or the values from the light sensor that are being sent between the main MCU and the app. Certain values are then returned to the user so that they may view them. No particular solutions have been implemented as a result of the fact that there is no obvious breach of data in this situation that might result in significant harm to property or loss of information. Naturally, the Wi-Fi connection will adhere to the IEEE 802.11 standard, which incorporates built-in safety procedures. One example of this is Wi-Fi protected access 3, which encrypts the data that is transmitted over the network. On the other hand, despite the fact that the data being communicated is not very harmful, it is still essential to provide security in order to protect against any potential flaws that may exist inside the system. The prospect of unauthorized access to the wireless network is the key cause for concern. This access might provide hostile actors the ability to interrupt communication between the devices or intercept important data. The use of encryption methods such as WPA3 (Wi-Fi Protected Access 3) on all wireless connections is one method that may be utilized to overcome this challenge. With the implementation of WPA3, every data that is sent over a Wi-Fi network is encrypted, making it far more difficult for malicious actors to decrypt or take control of the transmission. Additionally, the utilization of Secure Sockets Layer (SSL) or Transport Layer Security (TLS) for communication between the mobile application and the ESP32 can further safeguard the data that is being transferred. This is accomplished by providing end-to-end encryption, which guarantees that the information is encrypted before it leaves the phone and continues to be secure until it reaches the ESP32.

Device authentication provides an additional degree of safety by guaranteeing that only approved devices are able to connect with the ESP32. In this particular scenario, the implementation of a device pairing procedure, in which the mobile application and the ESP32 participate in the exchange of authentication tokens or keys prior to the beginning of any data transmission, would offer an extra layer of protection. Additionally, the ESP32 itself ought to be designed to reject any attempts at illegal access, and it ought to include security mechanisms such as firewall rules in order to prevent exposure to

ports that are not required. Last but not least, it is recommended that the ESP32 and the mobile app receive frequent software upgrades in order to repair any vulnerabilities that may have been detected. The project is able to limit the danger of security breaches by complying to certain security standards and best practices, such as employing WPA3, SSL/TLS, and device authentication. This is the case even if the data that is being transferred is not very sensitive. Through the implementation of this strategy, the system will continue to be dependable and secure for its users.

5.0 Comparison of ChatGPT with Other AI

5.1 Navigating the Landscape of AI-Assisted Research

The integration of generative Artificial Intelligence (AI) is a transformative element for a project like the GreenSync Smart Garden System. These tools can serve as powerful assistants, moving beyond the function of a simple search engine to become integral to design, problem solving, and documentation processes. A primary benefit is the significant acceleration of the conceptualization and documentation phases, allowing a team to draft architectural frameworks and explore a wide array of technological solutions in a fraction of the time traditionally required.

However, this research consistently underscores the limitations of AI in the context of complex, hardware dependent engineering. While AI excels at generating foundational code and high level strategies, its recommendations often lack the practical nuance required for real world embedded systems. Issues like the need for interrupt driven processes, power optimization for specific components, and understanding implicit project constraints are frequently overlooked by the AI. This highlights a critical learning outcome: AI is a powerful assistant that can augment human ingenuity, but it cannot replace the critical evaluation, hands on validation, and deep contextual understanding that defines human engineering judgment. The role of the engineer is to scrutinize, refine, and synthesize AI generated information to fit the precise needs of a project.

5.2 Case Studies in AI-Driven Component Analysis

AI generative assistance can be leveraged to research specific technical challenges and guide component comparisons. The following examples

summarize the AI interactions, with the full text of the prompts and responses located in the appendix.

5.2.1 Example 1: Water Pump Selection

To compare different 12V DC water pump technologies, a detailed prompt was formulated requesting a comparison of peristaltic, diaphragm, and centrifugal pumps based on specific performance criteria (see Appendix, [AI 1]). The resulting AI generated response provided a highly detailed comparative analysis. It identified the diaphragm pump as a strong candidate, noting its robust flow rate of approximately 1.8 L/min and its crucial self priming capability, which simplifies system design. However, it also cautioned about potential drawbacks like pulsatile flow and noticeable mechanical noise (in the 50 to 60 dB range). The AI contrasted this with peristaltic pumps, which offer superior precision for dosing but have lower flow rates, and with centrifugal pumps, which are quieter but are typically not self priming, a critical design constraint. The analysis also extended to the electronics, recommending a MOSFET for power control and specific protective circuitry like a flyback diode to suppress EMF spikes.

This type of response is highly valuable because it provides a direct, data driven comparison based on the project's hypothetical constraints. The detailed breakdown affirms the advantages of a diaphragm pump for this application. Furthermore, the AI's warnings about pulsatile flow and electromagnetic interference (EMF) are key considerations for such a design, highlighting the need to incorporate components like a pulsation accumulator and a flyback diode.

5.2.2 Example 2: Sensor Battery Life Optimization

To research strategies for long term battery life, a specific prompt was used, detailing the components, communication protocol, and a target lifespan of 180 days (see Appendix, [AI 2]). The AI's synthesized reply outlined a comprehensive, multifaceted strategy for aggressive power management. For the microcontroller, it recommended leveraging the ESP32's Deep Sleep mode, where current consumption could drop to as low as 10 to 20 microamperes. For the sensor hardware, it proposed a power gating strategy using a low side MOSFET to completely de-power the sensor between readings, as even low power sensors have a quiescent current draw. On the

firmware side, it suggested extending battery life by using longer sampling intervals, such as every 15 to 30 minutes, and optimizing communication by using the low power ESP NOW protocol to send data only when significant changes occur. To demonstrate the viability of this approach, the response included a conceptual power budget calculation, which estimated a potential battery life well over the 180 day target.

This interaction would be highly effective for research. By providing precise constraints, a detailed strategy is generated that is directly actionable for theoretical analysis. The response provides specific hardware and firmware recommendations, and the sample power budget allows for immediate validation of the battery life goals, providing a clear roadmap for potential implementation.

5.3 Triangulating Solutions with Competing AI Models

To address more complex challenges, one can query multiple AI platforms with the same prompt. This approach reveals that each AI often provides a valid but incomplete piece of the solution, requiring human synthesis to create a robust theoretical design. For this research, a complex, multi criteria prompt was given to ChatGPT, DeepSeek, and Claude to propose a corrosion prevention strategy for a PCB soil moisture sensor (see Appendix, [AI 3]).

5.3.1 ChatGPT

ChatGPT's response focused on surface level protection methods, providing a solid foundation of standard industry practices. Its key recommendations included applying a thin acrylic conformal coating for general moisture protection, using a solder mask on non sensing areas, and utilizing Electroless Nickel Immersion Gold (ENIG) plating on the exposed sensor pads for its superior corrosion resistance and signal integrity.

5.3.2 DeepSeek

DeepSeek's advice represented a more rugged, physically robust solution. It recommended a complete epoxy potting of the board's main body using a specific marine grade compound for superior water resistance. Critically, it also introduced the concept of preventing galvanic corrosion between

dissimilar metals by recommending the use of 316 stainless steel for any external metal contacts.

5.3.3 Claude

Claude offered a unique and more holistic view, considering the entire lifecycle and material integrity of the sensor. It advised selecting a high TG (glass transition temperature) FR 4 board for its inherent resistance to moisture induced warping and degradation. It also focused on design elements like trace width and spacing to maintain sensor accuracy and uniquely suggested a potential long term maintenance protocol involving rinsing the sensor.

5.3.4 AI Comparison Synthesis and Conclusion

This multi AI comparison is profoundly valuable because no single platform provides the complete, optimal answer. A final, synthesized strategy would be a synergistic combination of these varied recommendations:

- **Foundational Resilience (from Claude):** The synthesized approach would prioritize a high TG FR 4 board material.
- **Bulk Encapsulation (from DeepSeek):** It would adopt the comprehensive epoxy potting strategy.
- **Precise Sensing Window and Plating (from ChatGPT):** This strategy would integrate the idea of a precisely defined sensing window with ENIG plating.
- **Galvanic Isolation (from DeepSeek):** It would also incorporate the use of 316 stainless steel for external metal contacts.

This research process demonstrates a powerful human AI collaboration. While the AIs provide intelligent components of a solution, the ultimate responsibility for critically evaluating and combining these components into a cohesive engineering design rests with the human engineer. A final design synthesized in this way would be superior to any single AI's suggestion.

5.4 Critical Assessment of AI Tools in an Academic Setting

When applying generative AI platforms to a complex engineering project like GreenSync, a clear pattern of benefits and drawbacks emerges, directly influencing the student learning experience.

Pros (Benefits)

- **Rapid Information Synthesis:** The most significant advantage is the AI's ability to instantly collate and compare vast amounts of technical data. For instance, when researching water pumps, the AI can synthesize datasheets, application notes, and forum discussions into a concise comparison of flow rates, power needs, and other features, a task that would manually take hours or days.
- **Multi Perspective Brainstorming:** Using different AI models (like ChatGPT, Claude, and DeepSeek) on the same problem acts as a virtual brainstorming session with multiple "experts." As seen in the PCB corrosion example, one AI might focus on coatings while another focuses on material science, providing a more well rounded set of potential solutions than a single source might offer.
- **Conceptual Scaffolding:** AI is excellent at creating a starting framework or "scaffold" for complex tasks, whether it's outlining a technical report, generating boilerplate code for a software module, or proposing an initial system architecture. This helps overcome the initial "blank page" problem and provides structure to the design process.

Cons and Limitations

- **Context Blindness:** AI models have no true understanding of the unstated, implicit context of a project. An AI might suggest a power saving strategy without knowing the specific battery chemistry being used or recommend a component without considering the physical space constraints of the enclosure. This requires the engineer to constantly provide highly detailed context.
- **Lack of Practical Nuance:** AI generated solutions are often theoretically correct but practically flawed. For an embedded system, an AI might generate code that works but is highly inefficient, uses blocking delays that would be catastrophic in a real time system, or overlooks critical hardware optimizations like using DMA to free up the CPU.

- **Plausible Inaccuracies ("Hallucinations"):** AIs can generate information that is incorrect but sounds perfectly plausible. It might invent a function name, misstate a technical specification, or combine two unrelated concepts. This places the burden of rigorous verification and fact checking entirely on the engineer, as blindly trusting the output can lead to significant errors.

Impact on the Learning Experience in Senior Design

The use of these platforms creates a unique set of learning opportunities and potential pitfalls:

- **Benefit: Fosters Critical Synthesis Skills:** The most powerful learning experience came from the multi AI comparison for PCB corrosion. The platforms provided conflicting yet complementary advice. This forced a higher level of engineering thinking: not just finding an answer, but critically evaluating multiple partial answers and synthesizing them into a superior, cohesive solution. This directly mimics real world engineering where solutions are rarely monolithic.
- **Benefit: Teaches Precision in Problem Definition:** A major lesson was in prompt engineering. An initial vague prompt like "How to make my sensor last longer?" yields a uselessly generic response. Learning to formulate a precise engineering query, specifying the exact components, operating environment, and performance metrics, is a skill in itself. This process teaches students to define a problem with the technical precision required for a real engineering task.
- **Harm: Risk of Superficial Understanding:** A significant danger is accepting an AI's answer without understanding the underlying principles. For example, if a student accepts a block of code for controlling a motor without understanding why a flyback diode is necessary, they haven't learned circuit protection; they've only learned to copy and paste. This can create a false sense of competence and lead to shallow learning.
- **Benefit: Identifies and Fills Knowledge Gaps:** When an AI provides a high level but incomplete answer (e.g., "use a MOSFET to control the sensor power"), it acts as a signpost. It highlights a necessary component but doesn't explain the implementation details (e.g., choosing N channel vs. P channel, calculating the required gate

resistor). This prompts the student to recognize a gap in their own knowledge and drives them to perform focused, deeper research on that specific topic, leading to more robust learning.

5.5 Concluding Analysis on Human-AI Collaboration

Ultimately, the integration of generative AI into a project like GreenSync is an invaluable and transformative aspect of the senior design experience. While these platforms serve as powerful accelerators for research and initial design, the most profound lesson is in recognizing their inherent limitations. This research consistently demonstrates that AI generated output, while often theoretically sound, requires rigorous human engineering validation to address its frequent lack of practical nuance and real world context. The process of synthesizing complementary, and at times conflicting, advice from multiple AI platforms underscores that the true value of these tools lies in their ability to broaden an engineer's perspective, not to provide a single, definitive answer. Therefore, this exploration concludes that generative AI is best wielded as an augmentative tool, one that enhances, rather than replaces, the indispensable critical thinking, hands on validation, and creative problem solving skills that define the engineering discipline.

Disclosure: The final text of this report, including all analyses and conclusions, were written by the authors but with the aid of AI platforms. ChatGPT, DeepSeek, and Claude were used to rephrase and correct grammar errors. As well as generating the responses to the prompts documented in appendix E which were solely used for research purposes.

Chapter 6: Hardware

6.1 MCU PCB Schematic

6.1.1 ESP32

The ESP32 microcontroller will operate using power supplied through a 3.3V voltage regulator. For USB communication functionality, the system will incorporate a HiLetgo CP2102 USB-to-TTL UART development board. This converter module will establish connections with the ESP32 through four essential pins: VCC for power, TX and RX for data transmission (utilizing pins

35 and 34 respectively on the ESP32), and GND for ground reference. The ESP32 will implement the I2C communication protocol, with pin 13 designated for the SCL (serial clock) line and pin 14 assigned to the SDA (serial data) line. This will allow us to interact with our I2C multiplexer in order to read data from our sensors. GPIO pins have also been chosen to connect to the camera. During the PCB design phase, multiple test points will be strategically incorporated throughout the board to facilitate connection verification and troubleshooting.

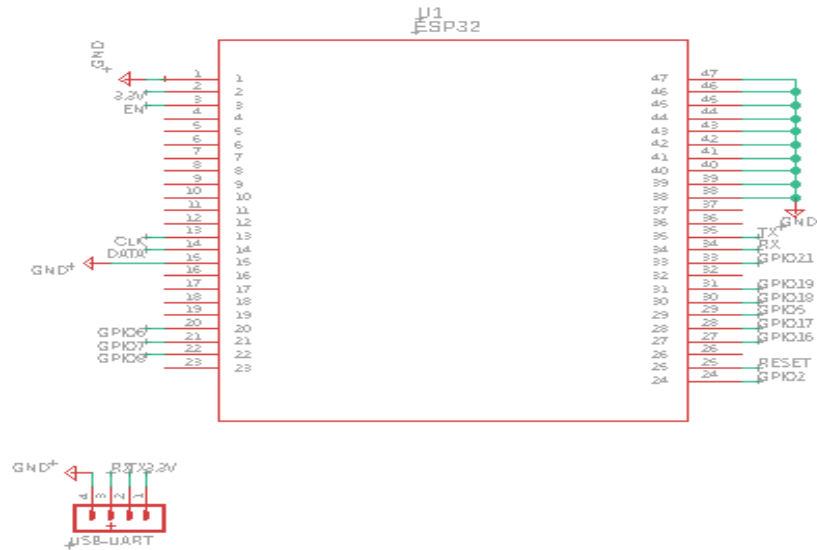


Figure 4: PCB Schematic

The ESP32 is the same chip used previously, but all 38 pins were used, the 3.3V input was changed to have two capacitors in parallel, and the enable and reset were added to have a capacitor in parallel with the switch.

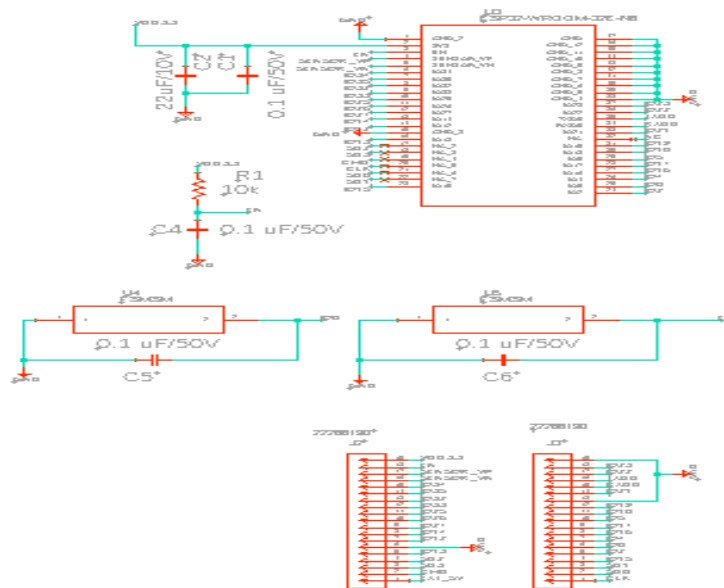


Figure 4 Version 2: PCB Schematic

6.1.2 Regulators

The custom PCB design features dual voltage regulation circuits consisting of both 3.3V and 5V boost converters. These regulators draw power from a PCIFR18650-1500 lithium iron phosphate (LiFePO₄) battery that delivers 3.2V nominal output with 1.5Ah capacity.

For the 3.3V power supply, the TPS613221ADBVT serves as the primary regulator—a synchronous boost converter with high efficiency that accepts input voltages between 0.9V and 5.5V, making it ideal for battery-operated systems with low voltage requirements. This component offers a peak switch current limitation of approximately 1.2A, providing sufficient margin for peak power demands, especially when the ESP32 performs RF transmission operations.

Following manufacturer guidelines, the inductor selection fell within the recommended 0.7 μ H to 13 μ H range, with 4.7 μ H chosen as the optimal value to achieve proper balance between physical dimensions, stability requirements, and current capacity. Output filtering utilizes a 22 μ F ceramic capacitor, selected from the acceptable 4.7 μ F to 100 μ F range, which maintains voltage stability during varying load demands. Near the battery

connection point, an extra bypass capacitor helps mitigate voltage drops that may occur during initial power-up or rapid load transitions.

The TPS613222ADBVT handles 5V regulation, sharing pin compatibility and similar specifications with its 3.3V counterpart. While not utilized in the current hardware iteration, this 5V supply remains part of the circuit design to accommodate potential future additions requiring elevated operating voltages. Component selections for both regulators remain consistent, using matching inductors and capacitors to streamline procurement and board layout.

Manual power control capability exists through an integrated power switch within the regulator circuitry, facilitating controlled power states for testing procedures and field deployment. The 3.3V supply serves the ESP32 microcontroller along with various sensors and logic-level components. Implementation will follow established PCB design principles, including minimized trace lengths for switching signals, thermal via implementation for heat management, and strategic capacitor positioning to maintain regulator performance while reducing electromagnetic interference.

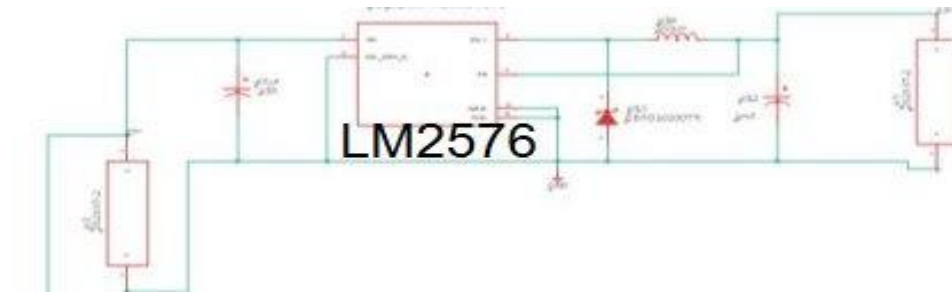


Figure 5: Regulator Schematic

6.1.3 I²C Multiplexer

An I²C multiplexer like the CA9548A lets a single master, such as the ESP32, route its SDA and SCL lines to one of eight independent downstream channels. This avoids address conflicts when multiple I²C devices share the same fixed address and also isolates buses for stability. In the schematic, the chip is powered at 3.3 V with ground returns, while the main SDA and SCL lines are tied to pins 20 and 19 and pulled up to 3.3 V through 1 kΩ resistors. The A0–A2 pins set the device address, defaulting to 0x70 if left low, and the RESET pin allows external resetting. The outputs SD0–SD7 and SC0–SC7 then connect to devices, with the ESP32 enabling the desired channel through simple I²C commands. This arrangement makes it possible to manage multiple devices cleanly on a single I²C system.

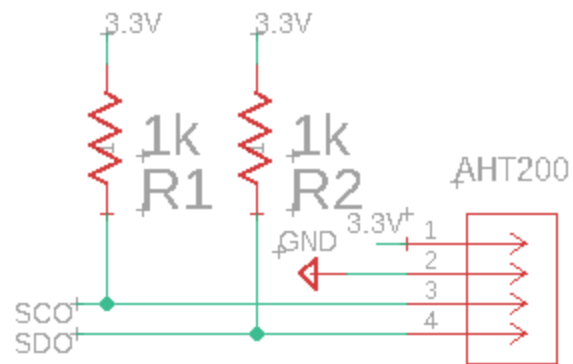


Figure 7: AHT20 Sensor Connection

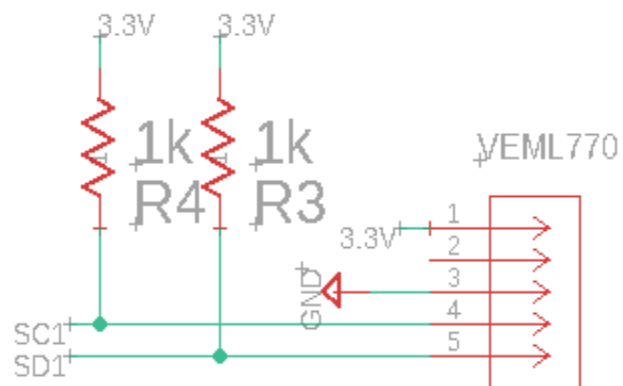


Figure 8: VEML770 Sensor Connection

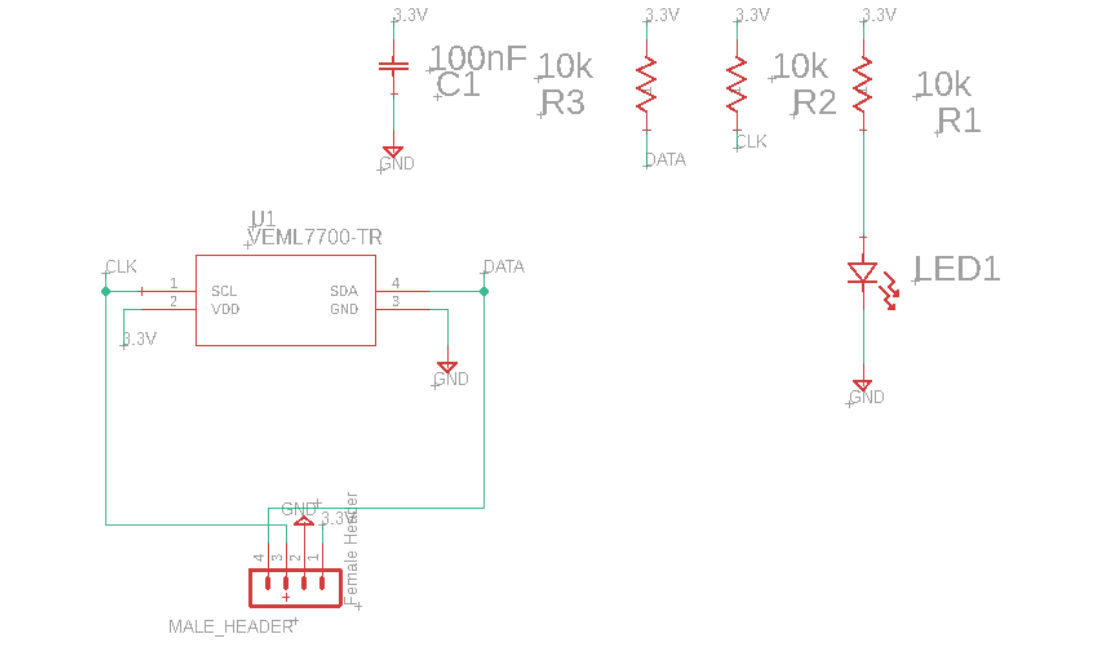


Figure 8 Version 2: VEML770 Sensor Connection

AHT20 humidity and temperature sensor was not used in the formation of this project.

6.2 Water Pump Subsystem Block Diagram

The relationship between system components and their respective power connections is depicted in the included block diagram. A 12V AC-to-DC converter serves as the primary power source, simultaneously supplying both the diaphragm water pump and the 12V solenoid valve through a common power rail.

Water supply originates from a plastic reservoir located below the plant platform. The pump pressurizes the PVC distribution line to a maximum of 50 PSI, triggering an integrated pressure switch that automatically halts pump operation to preserve system pressure. The pressurized water remains static within the plumbing network due to the solenoid valve's normally closed state, which blocks any unintended water release.

Irrigation control relies on continuous soil moisture sensor feedback. Upon detecting moisture levels falling below predetermined parameters, the ESP32 microcontroller energizes a GPIO pin to trigger the solenoid valve opening. This action permits water passage through the valve into drip irrigation tubing that targets the plant's root system.

When sensor readings indicate soil moisture has reached the upper threshold, the microcontroller deactivates the GPIO signal, closing the solenoid valve and terminating water delivery. The pump maintains operation until its pressure switch registers 50 PSI system pressure and initiates automatic shutdown. This design maintains optimal system pressurization independent of irrigation cycle frequency.

Critical design factors include AC-to-DC power supply specifications, which must accommodate concurrent operation of the pump, solenoid valve, and LED growth lighting system. Power supply selection followed established engineering protocols by calculating peak current requirements for all components during maximum load conditions, then incorporating a 10% safety factor. The chosen converter satisfies these specifications while guaranteeing stable performance across all anticipated operating scenarios.

The water subsystem functions without requiring a custom PCB, as both the moisture sensor and microcontroller operate as discrete hardware units. Their individual hardware implementations, encompassing PCB designs and voltage regulation circuits, receive comprehensive coverage in later report sections. The implementation utilizes a 12V solenoid valve appropriate for direct electrical connection, enabling smooth integration with the control architecture.

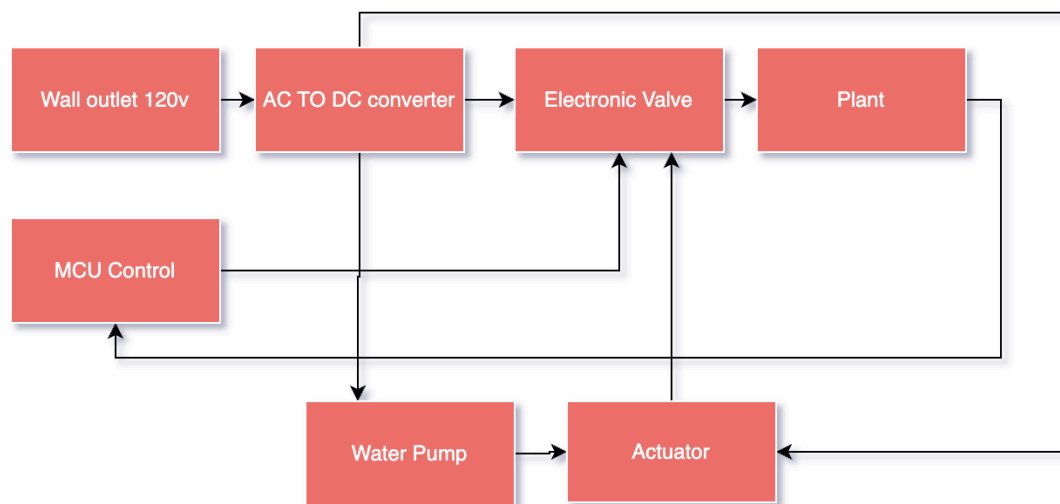


Figure 9: Water Pump Subsystem Diagram

6.3 Moisture Sensor

The soil moisture sensing subsystem designed for individual plant monitoring incorporates multiple interconnected components. The configuration establishes connections between each element and the ESP32 microcontroller while defining interactions among the sensor, power infrastructure, control mechanisms, and user interface components.

Power delivery begins with a standard 120V AC wall outlet serving as the main energy input. Voltage conversion and regulation occur through an AC-to-DC converter that reduces and stabilizes the voltage to provide appropriate DC power levels for the ESP32, 12V water pump, and electronic solenoid valve. Converter specifications account for the cumulative current requirements of all components plus an additional 10% safety factor, guaranteeing reliable performance across varying load scenarios.

Moisture monitoring utilizes a single capacitive soil moisture sensor positioned within the plant's growing medium for continuous water content measurement. The sensor generates an analog voltage output proportional to soil moisture levels. The ESP32's integrated ADC (Analog-to-Digital Converter) transforms this analog signal into digital values suitable for software processing, enabling instantaneous decision-making based on moisture conditions.

Upon detecting moisture levels below established parameters, the ESP32 energizes a designated GPIO pin to open the solenoid valve, initiating water flow through drip irrigation tubing. Concurrently, the diaphragm pump extracts water from the storage tank and builds system pressure until reaching 50 PSI, at which point the pump's integrated pressure switch triggers automatic shutdown. When soil moisture reaches target levels, the ESP32 deactivates the GPIO signal, closing the solenoid valve and halting irrigation. The pump subsequently restores system pressure before shutting down automatically.

User interaction features include an LCD display presenting live information including moisture measurements, pump/valve operational status, and irrigation event logs. This visual feedback enables immediate system monitoring and plant condition assessment. The ESP32's wireless capabilities support communication with mobile applications or web-based interfaces, facilitating remote sensor monitoring, threshold adjustments, and alert notifications, delivering intelligent plant care management from any location.

Additional design considerations include a 555 timer-based PWM (pulse-width modulation) generator circuit, which the current implementation does not utilize but remains noteworthy for its capability to provide duty-cycled power control for solenoid or pump operation. This circuit

topology, commonly employed in smart irrigation applications, offers valuable insights into timing control methodologies and power modulation techniques that could enhance future system iterations.

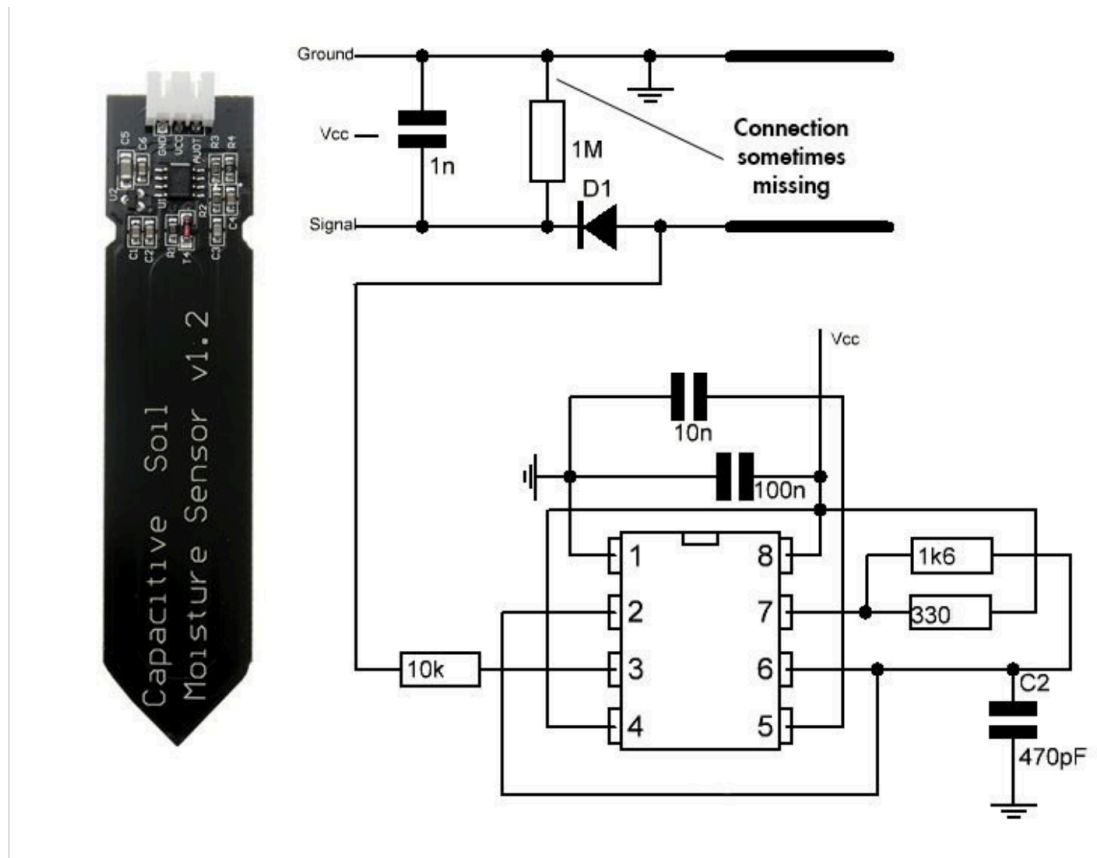


Figure 10: Typical Moisture sensor schematic

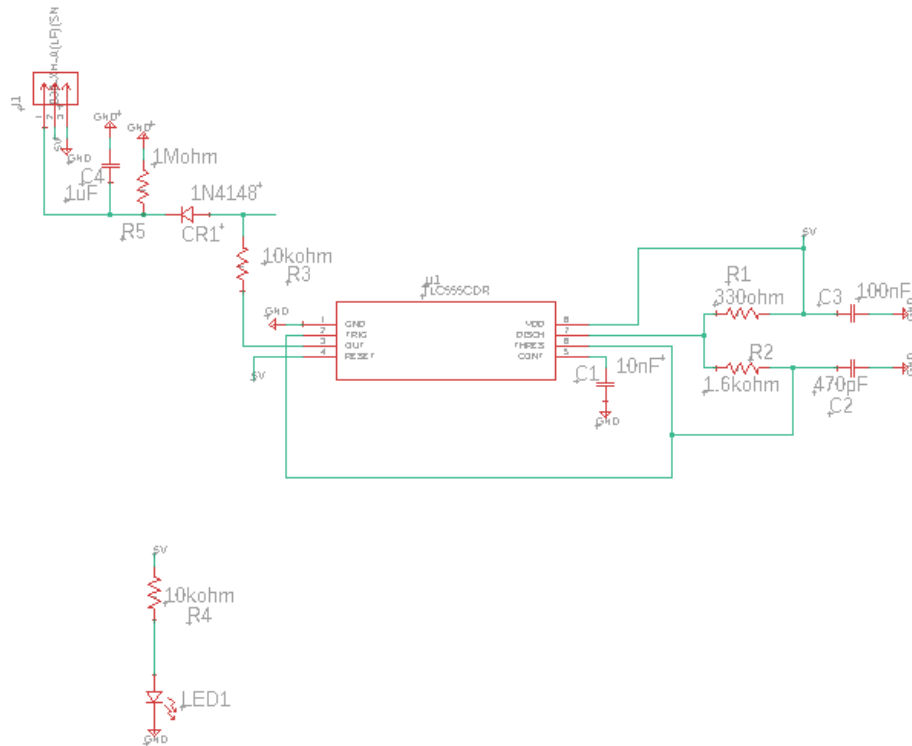


Figure 10 Version 2: Typical Moisture sensor schematic

6.3.1 Voltage Regulation

The voltage regulation subsystem serves an essential function in delivering consistent and dependable power to delicate electronic components. This design produces a stable 3.0V DC output, critical for proper functionality of timing-based circuits like 555 timers and various low-voltage digital devices requiring steady power supplies.

IC1, a Low Dropout Regulator (LDO), forms the central element of this power subsystem. LDO technology excels in scenarios demanding precise voltage control with modest current requirements, maintaining regulated outputs even when input voltages barely exceed target levels. The LDO effectively reduces incoming voltage (commonly sourced from 5V supplies or batteries) to produce a smooth 3.0V output while minimizing thermal generation and voltage ripple—factors that could degrade overall system operation.

Multiple decoupling and bypass capacitors—designated as C1, C2, C6, and C8—enhance power supply stability throughout the circuit. These components eliminate high-frequency interference and suppress temporary voltage variations, creating a "cleaned" power signal. Such filtering proves

essential for embedded applications where minor voltage irregularities can trigger unpredictable responses in timing-critical devices such as 555 timers or microcontroller analog sections.

Ground (GND) nodes throughout this circuit establish the common electrical reference for voltage readings and current return routing. Implementing a robust, low-impedance ground plane during PCB development reduces noise interference and strengthens the power distribution infrastructure's dependability.

This voltage regulation architecture guarantees all connected components access to clean, stable 3.0V power, fundamental for precise timing control and digital circuit operation. Absence of proper regulation would allow voltage fluctuations to introduce timing errors, erratic sensor outputs, or irregular system responses. The combination of LDO-based regulation and appropriate capacitive filtering delivers consistent performance across all essential components regardless of load variations.

6.4 Lighting Subsystem Block Diagram

The block diagram below illustrates the lighting subsystem tailored for a single-plant environment, detailing component interconnections and their associated power supply paths. This subsystem plays a critical role in regulating plant growth conditions by dynamically adjusting light intensity and spectrum based on real-time sensor feedback and user-defined parameters.

As with the water delivery system, power is supplied through a 12V AC-to-DC converter, which provides regulated DC voltage to all lighting-related components, except for the ESP32 microcontroller, which is powered separately via its onboard 3.3V regulator. The central control component in this subsystem is the Shelly RGBW2, a Wi-Fi-enabled smart LED controller that allows precise dimming and color tuning of RGB LED light strips. The Shelly unit supports both 12V and 24V input, and in this application, it is operated at 12V for compatibility with the power supply used across the system.

The Shelly RGBW2 establishes a wireless connection with the ESP32, receiving control signals via the local Wi-Fi network. Based on commands issued from the mobile application interface, the ESP32 adjusts brightness and color output to meet the specific light requirements of the plant. This control includes smooth dimming, color blending, and on/off scheduling, offering full-spectrum lighting customization to simulate optimal growing conditions, including daylight cycles and intensity transitions.

To provide feedback for closed-loop control, a VEML7700 ambient light sensor is positioned near the plant to monitor light intensity in real time. This sensor communicates with the ESP32 over the I²C bus and returns lux measurements that the system uses to evaluate lighting effectiveness. The ESP32 analyzes these readings and adjusts the output of the Shelly-controlled light strips accordingly, ensuring the plant receives the appropriate amount of light based on its species-specific needs or programmed care profile.

As with other subsystems, careful consideration is given to the power budget of the AC-to-DC converter, which must simultaneously support the lighting system, solenoid valve, and water pump. The converter was selected to handle the combined peak current draw with an added 10% safety margin, ensuring stable and reliable performance under all operating conditions.

No custom PCB is required for the lighting subsystem itself. The light sensors are preassembled modules with onboard I²C breakout connectors and are directly wired to the MCU. The ESP32 PCB design—which handles system logic and control—will be covered in a separate subsystem section. This modular design approach reduces complexity, facilitates easier integration and testing, and allows for future expansion with minimal redesign.

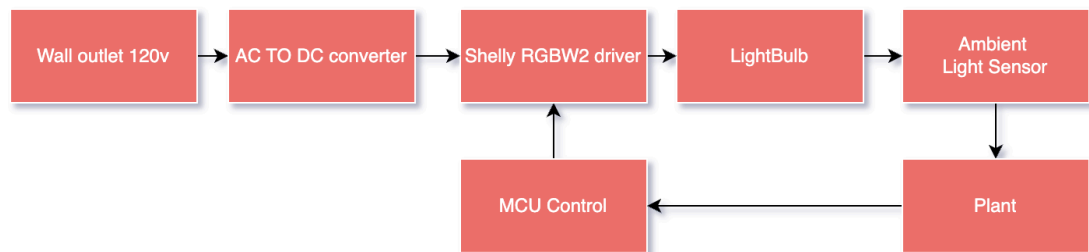


Figure 11: Lighting Subsystem Block Diagram

6.5 Shelly RGBW2 Driver Schematics

The Shelly RGBW2 plays a central role in the lighting and control architecture of this system, serving as a Wi-Fi-enabled relay and LED driver. In this single-plant application, the Shelly's functionality is leveraged both for wirelessly controlling the RGB LED grow lights and as a smart on/off switch for the 12V DC solenoid valve used in the irrigation subsystem. Its compact footprint, direct integration with Wi-Fi networks, and compatibility with both

12V and 24V DC systems make it a practical and flexible choice for small-scale smart agriculture applications.

When configured in LED control mode, the Shelly RGBW2 can dim and color-mix standard RGBW LED strips, enabling fine-tuned light intensity and spectral composition based on plant-specific needs. This feature allows the ESP32 to remotely adjust lighting conditions using color blending and pulse-width modulation (PWM), achieving optimal photosynthetic response. For this application, it is operated in 12V mode to match the system-wide DC supply and ensure compatibility with other components such as the pump and valve.

Alternatively, when configured in generic switch mode, the Shelly RGBW2 can act as a wireless relay driver, providing on/off control over non-lighting devices—such as a 12V solenoid valve—through its output terminals. In this configuration, each output channel is effectively a remote-controlled low-side switch that can be toggled via commands from the ESP32 microcontroller through a local network. For this single-plant system, only one valve is needed, and it is connected to a single channel of the Shelly output. The unused channels remain inactive or available for future expansion.

Shelly provides detailed wiring schematics for both LED lighting and generic load control configurations on its official documentation site. These can be followed with minimal adaptation: in the valve control setup, for example, the 12V DC solenoid valve is connected in place of the LED strip, with power routed through one of the RGBW channels. The ESP32 communicates with the Shelly via Wi-Fi, sending commands to toggle the output based on moisture sensor readings or user input from the mobile interface.

No custom PCB is required for integrating the Shelly RGBW2 into this system, as all connections are made via terminal blocks or screw terminals on the module itself. This allows for rapid prototyping and simplified installation without the need for additional circuit design. However, if desired, the system could be migrated in the future to a more compact and power-efficient alternative—such as an ESP32-based MOSFET switch or dedicated relay module—to reduce cost and eliminate dependency on third-party cloud services.

In summary, the Shelly RGBW2 provides a robust, network-ready solution for both lighting control and solenoid actuation in this smart gardening application. Its wireless capabilities, versatility, and ease of integration make it an ideal fit for the current project scope, with room for future scaling and customization.

White LED bulbs scheme

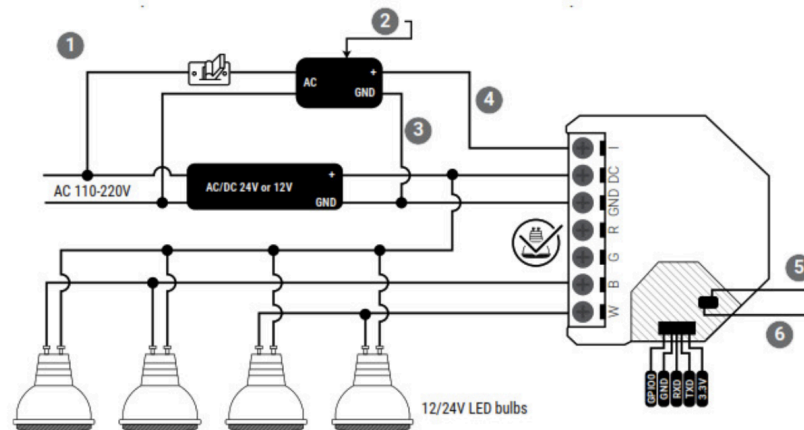


Figure 12: Shelly RGBW2 Driver Schematic

Didn't use in the overall formation of the project.

6.6 Camera:

To enhance plant monitoring and support long-term health tracking, the system incorporates an ESP32-compatible camera module to capture periodic images of the plant. For this application, the ESP32-CAM module is selected due to its compact design, low power requirements, and native integration with the existing ESP32 platform. This module combines an OV2640 camera sensor, microSD card slot for local image storage, and onboard Wi-Fi and UART connectivity—making it well-suited for standalone or networked image capture tasks within embedded IoT environments.

The OV2640 sensor provides a resolution of up to 1600×1200 (UXGA), more than sufficient for assessing plant growth, detecting discoloration, and documenting environmental changes over time. Configurable resolution and compression settings allow images to be optimized for bandwidth or memory constraints, depending on whether they are stored locally on an SD card or streamed over Wi-Fi to a cloud dashboard or mobile app.

The ESP32-CAM module operates at 3.3V, which aligns with the system's regulated voltage rail and avoids the need for level shifting. Power is supplied through the same 3.3V LDO regulator used for the ESP32 mainboard and other sensors, ensuring stable operation. Like the ESP32 MCU, the camera module's current spikes during Wi-Fi transmission or image capture are brief but notable, so proper bypass capacitors should be placed near the module to handle transient demand and avoid brownouts.

In this system, the ESP32-CAM can either:

- Operate independently, triggered periodically via internal timers or external GPIO pulses from the main ESP32 controller, or
- Be integrated as a peripheral module that receives capture commands via UART or HTTP over Wi-Fi, allowing full coordination with the main irrigation and lighting systems.

Captured images can be:

- Stored on the microSD card onboard the ESP32-CAM for later review, or
- Uploaded to a cloud server or viewed in real-time through a web interface, leveraging the ESP32's built-in Wi-Fi capabilities.

For physical integration, the ESP32-CAM can be mounted in a 3D-printed camera bracket or enclosed in a weather-resistant housing for environments with high humidity or water exposure. The camera should be positioned to capture a top-down or angled view of the plant pot, ensuring visibility of leaves and stem structure for health assessments.

No additional PCB is required for the camera subsystem, as the ESP32-CAM is a self-contained module. Communication is wireless, and control logic can be incorporated via lightweight HTTP server code or GPIO signaling from the main ESP32 board. If desired, a MOSFET-based power switch can be added to cycle the camera module's power during off-hours, further conserving energy in battery-powered applications.

In summary, the ESP32-CAM module offers a cost-effective and seamless way to integrate visual monitoring into the smart plant system. It complements the existing ESP32 infrastructure, shares the same power supply, and provides powerful capabilities for plant tracking, remote diagnostics, and automated photo documentation—all essential features in advanced smart gardening and environmental monitoring platforms.

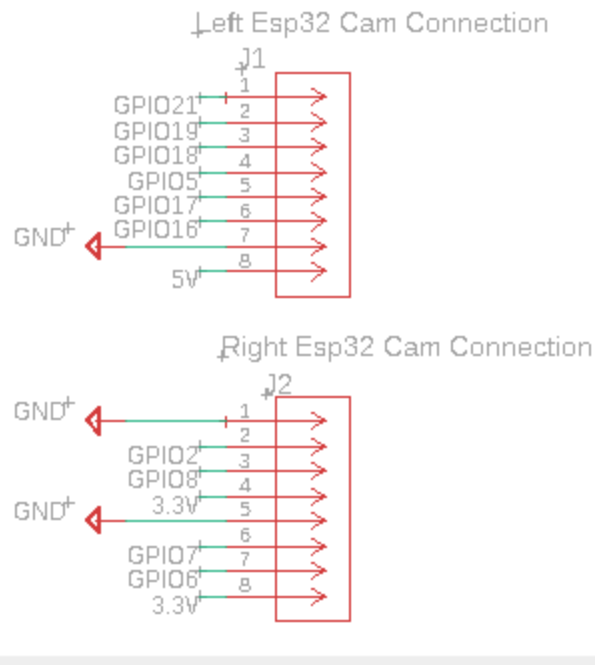


Figure 13: ESP32 Camera Connection Schematic

Chapter 7: Software

7.1 Core System Software Design

For the GreenSync system, the firmware will be architecturally split into two key domains: one dedicated to the individual soil moisture sensor nodes and the other for the Central Control Unit, or CCU, that acts as the system's brain. This separation is an intentional and critical design choice that will enhance modularity, simplify debugging, and allow each distinct hardware component to be optimized for its specific task. This distributed approach means that the sensor software can be perfected for low power operation, while the CCU software can be focused on complex data processing and network management. The CCU will serve as the system's intelligent nexus, responsible for advanced data analysis, managing the network connection with the mobile application, and executing precise, calibrated control over the garden's lighting and irrigation environments.

For instance, each sensor node will be equipped with specialized firmware focused exclusively on efficient and accurate data acquisition. Its primary

operational loop will involve reading the analog soil condition, performing an on chip analog to digital conversion, and then broadcasting this raw data packet over the wireless network. The CCU's more sophisticated firmware will receive these packets and will be responsible for all of the system's higher order logic. It will run a non blocking, real time control loop that continuously interprets the incoming sensor data, manages the Wi Fi connection to the user's mobile app, and makes calibrated decisions about when and for how long to activate the various environmental systems. The entire firmware design will emphasize responsive performance and robust power conservation strategies to ensure a seamless user experience and long operational life for battery powered components.

This state machine diagram illustrates the complete operational flow of an IoT sensor device. When the device first powers on through the Boot Sequence, it attempts to connect to a WiFi network in Network Client Module mode. If the connection fails due to a network timeout or wrong password, it switches to Access Point Module where it creates its own WiFi hotspot, allowing users to connect and provide correct network information. Once WiFi is successfully connected, the device proceeds to Hardware Init where it initializes and checks all sensors and components. If sensor readings are invalid (bad sensor values), it loops back to try initialization again; otherwise, when sensors are ready, it can either enter the Config Portal for user settings or proceed to the Monitor Loop. The Monitor Loop continuously checks sensors on a timer trigger schedule. When sensor changes are detected, the Data Handler processes this new information, leading to the Running State where the device is actively operating. If there's no activity, the device enters Power Savemode to conserve battery, from which it can be awakened by a wake signal to return to the Running State. Users can manually reset the device from the Access Point interface at any time. Eventually, all paths lead to System Halt when the device shuts down completely. This design ensures efficient power management while maintaining responsive sensor monitoring and network connectivity.

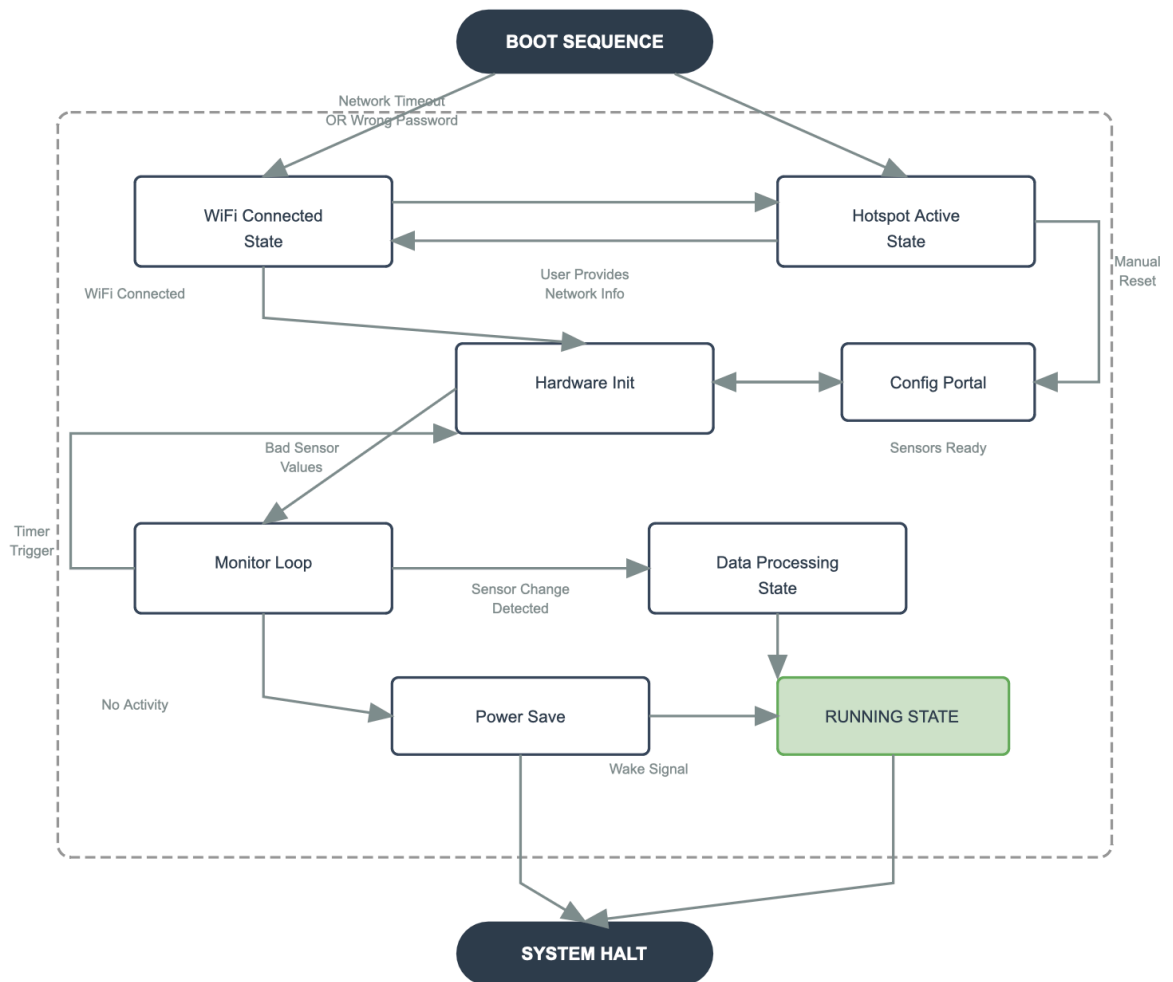


Figure 14: Software State Block Diagram

7.1.1 Development Environment and Tools

For the crucial task of firmware development, the project will leverage the widely supported Arduino IDE, with embedded C++ serving as the core programming language. This strategic decision is rooted in pragmatism and efficiency. By choosing an environment that is already deeply familiar to the development team, the project bypasses the significant time investment required to learn a more complex, professional toolchain like PlatformIO or the native ESP IDF. This accelerated learning curve allows engineering resources to be focused directly on developing the system's unique features rather than on mastering new tools. Furthermore, the immense global community behind Arduino provides an invaluable safety net, offering a vast repository of pre-vetted libraries, extensive documentation, and countless

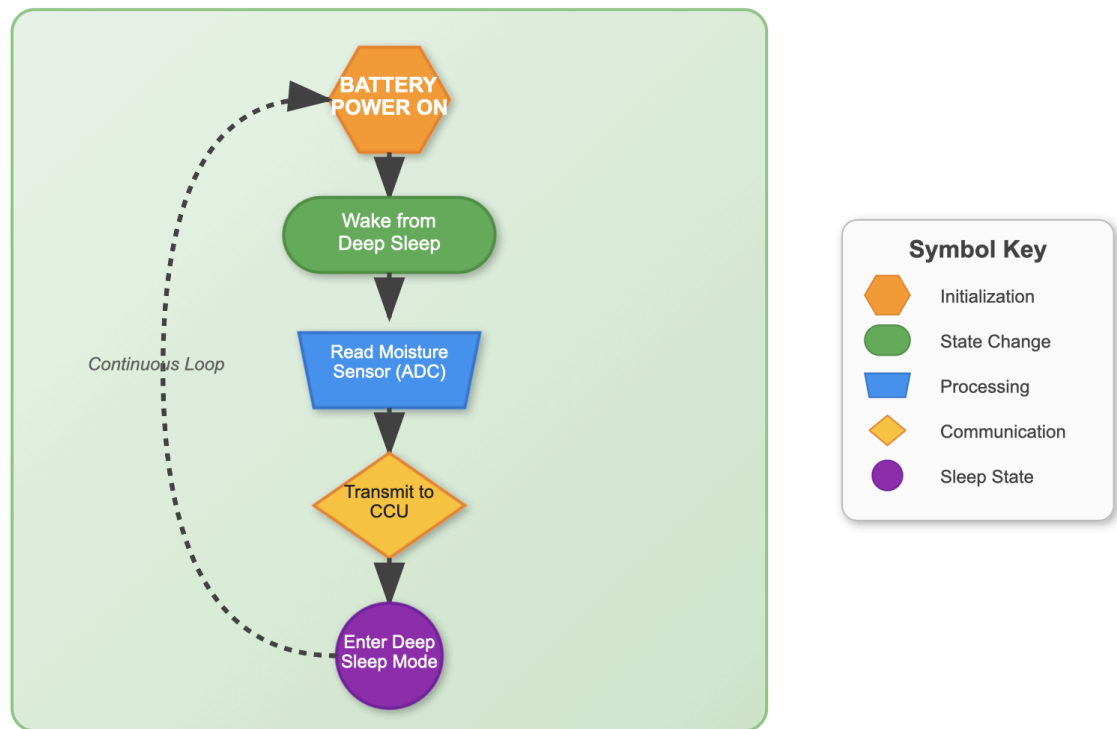
forum discussions that can dramatically speed up troubleshooting and development.

To facilitate collaborative development where multiple team members will work on different software modules in parallel, a centralized code repository will be established on GitHub. This version control system will be essential for tracking all changes, managing different feature branches, and providing the ability to roll back to previous stable versions in case of errors. This technical workflow, supported by active and disciplined communication on Discord, will be crucial for maintaining code integrity, preventing conflicting code merges, and ensuring all team members are working from a unified and up to date source.

7.1.2 Sensor Module Firmware

The GreenSync design will specify the deployment of multiple soil moisture sensor modules, all sharing a common hardware and software design for consistency and reliability across the entire garden environment.

GreenSync Sensor Module Operation

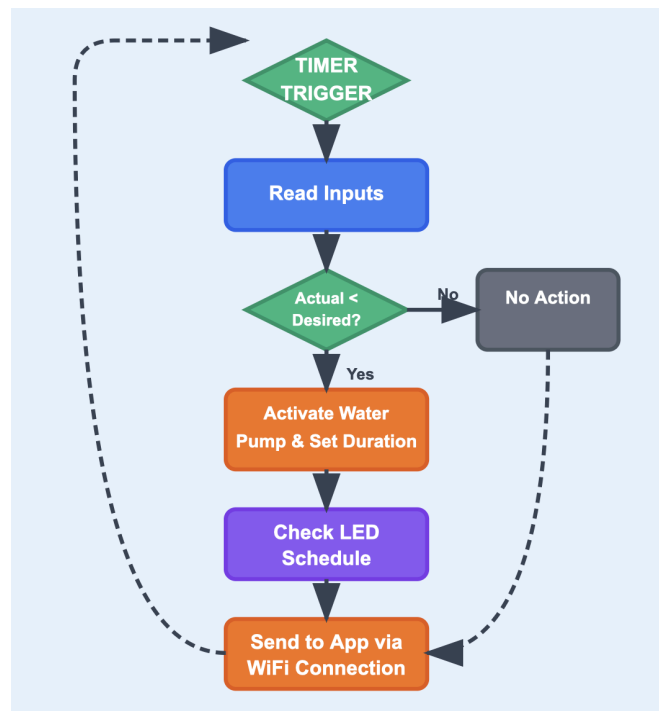


Power-optimized cycle: Wake → Read → Transmit → Sleep

Figure 15: GreenSync Sensor Module Operational Cycle

The guiding philosophy for the sensor module's firmware is extreme power efficiency. To achieve a long operational life on battery power, the software's logic will be kept as minimalistic as possible. The firmware's sole responsibility will be to accurately measure the moisture level of the soil. All computationally intensive tasks, such as data analysis, comparison to user defined thresholds, or historical data logging, will be deliberately offloaded to the more powerful, mains powered CCU. As the flowchart in the figure illustrates, the sensor's task will be a simple and focused recurring loop: wake from deep sleep, collect a single data point, transmit it, and immediately return to deep sleep. This approach makes the sensor a highly optimized, low energy data collection endpoint.

7.1.3 Central Control Unit Logic



Symbol Key

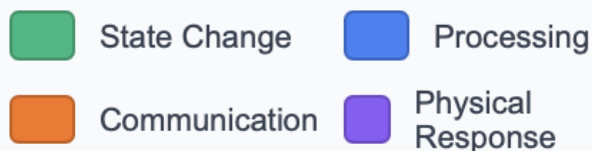




Figure 16: Central Control Unit Decision Logic Flowchart



Figure 17: System Initialization Process

The firmware for the main system's MCU will serve as the command center for the entire project, orchestrating every automated action the GreenSync system takes. This software, running on the main ESP32 processor, will be responsible for all intelligent operations. It will manage the two way communication link with the user's mobile application, receive and process

data broadcasts from all active soil sensors, and directly command the system's physical actuators, including the water pump and the LED lighting arrays.

The CCU's networking logic will be built to handle simultaneous Wi Fi connections, maintaining a persistent link to the mobile app while continuously listening for incoming data packets from the broader sensor network. The mobile app will serve as the source for all user defined "setpoints" and configuration data—such as target moisture levels and preferred lighting schedules—while the CCU will stream back real time telemetry. This will include current conditions, historical data charts, and overall system status, all for display on the user's device.

The core of the CCU's intelligence will lie in its advanced control algorithm. Rather than a simple on or off check, the firmware will calculate a "moisture deficit" by subtracting the current moisture value from the target value. The duration of the subsequent watering cycle will then be calculated to be directly proportional to this deficit. A small deficit might trigger a short, gentle watering, while a large deficit would result in a longer, more thorough cycle. This proportional control method is more sophisticated than a simple threshold trigger, as it will conserve water and actively prevent the system from overshooting the target moisture level, ensuring healthier soil conditions for the plants. The accuracy of this entire process will therefore be fundamental to the system's successful operation.

7.2 Mobile Application Framework

Flutter has been selected as the foundational framework for the user facing mobile application. This choice was based on Flutter's proven strengths in building performant, cross platform user interfaces from a single codebase, which is a significant advantage for development efficiency. Its ability to effectively manage real time data synchronization with external hardware like the ESP32 was also a key factor in its selection as the most suitable tool for this project. This approach allows for the creation of a consistent and high quality user experience on both iOS and Android devices without the need for separate development teams or efforts.

7.2.1 Application Development Stack

All application coding will be conducted within the VSCode IDE, enhanced with the official Flutter and Dart extensions for a seamless development and debugging experience. During the initial design and prototyping phase, Figma

has been used as a critical planning tool to construct a complete, full scale interactive mockup of the application's UI and logic. This blueprint, which includes detailed user flow diagrams and screen layouts, will guide the development team. The use of Figma's AI plugin, UXPilot, was also explored to rapidly prototype different UI concepts, ensuring a well considered and user friendly design from the project's inception.

To maintain a unified codebase and streamline project management, the application's source code will reside in the same GitHub Repository as the embedded firmware, organized in its own dedicated directory. This will provide a single source of truth for all software developed for the GreenSync project, simplifying the build and deployment process.

7.2.2 User Interface and Experience

The GreenSync application is designed as an intuitive and visually informative dashboard for the user's indoor garden. Access will be secured through a standard email and password authentication system to protect each user's data and settings. The primary user interface is a central "Dashboard" screen, which will display a dynamic icon for each plant registered to the system. These icons are designed to be informative at a glance, surrounded by status rings that will provide real time feedback on key metrics. For instance, a blue ring could represent the current soil moisture level, shrinking as the soil dries, while a yellow ring could indicate the cumulative light exposure for the day.

Tapping on a plant's icon will navigate the user to a detailed control screen for that specific plant. Here, they will be able to view historical data charts, precisely adjust target moisture levels, and manually override the automated lighting or watering schedules. Less frequently accessed sections, such as "System Settings" and "User Profile," are located in a slide out navigation drawer, a common mobile UI pattern that keeps the main interface clean and focused on the most important information. Push notifications will be reserved for critical alerts only, such as a low water level in the main reservoir, a design choice made to avoid overwhelming the user with non essential routine updates.

The "Add Plant" function within the app will allow users to register a new plant to an available slot in the GreenSync hardware. This straightforward process will involve naming the plant, uploading an optional image, and configuring its ideal environmental parameters. These user defined settings are the critical inputs that the CCU will use to inform its automated control logic.

7.2.3 User interface mobile application

The mobile application interface designs and system flowcharts presented in this section were developed using custom HTML and CSS prototyping techniques. Web based design methodologies were employed to create interactive mockups that accurately represent the intended user experience and system workflows. The mobile app screens were crafted using responsive CSS frameworks and modern design principles, allowing for realistic visualization of the GreenSync application's interface across multiple user scenarios including authentication, dashboard management, plant monitoring, and system settings. The accompanying flowcharts were designed using structured diagramming approaches with custom styling to clearly illustrate the operational cycles and decision logic of the sensor nodes and central control unit. This prototyping approach enabled rapid iteration and comprehensive visualization of both the user facing interface design and the underlying system architecture, providing a complete visual representation of the GreenSync project's software components.

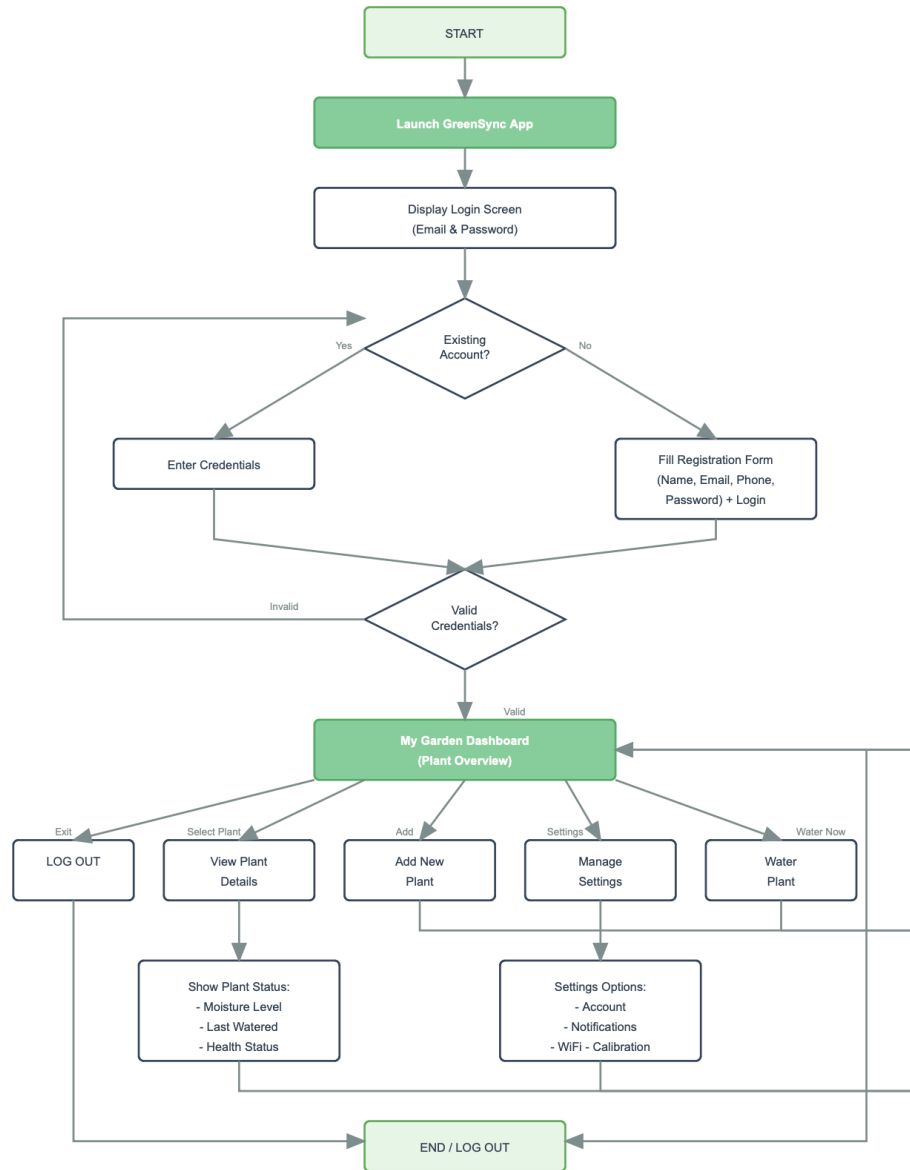


Figure 18: Mobile App Flowchart

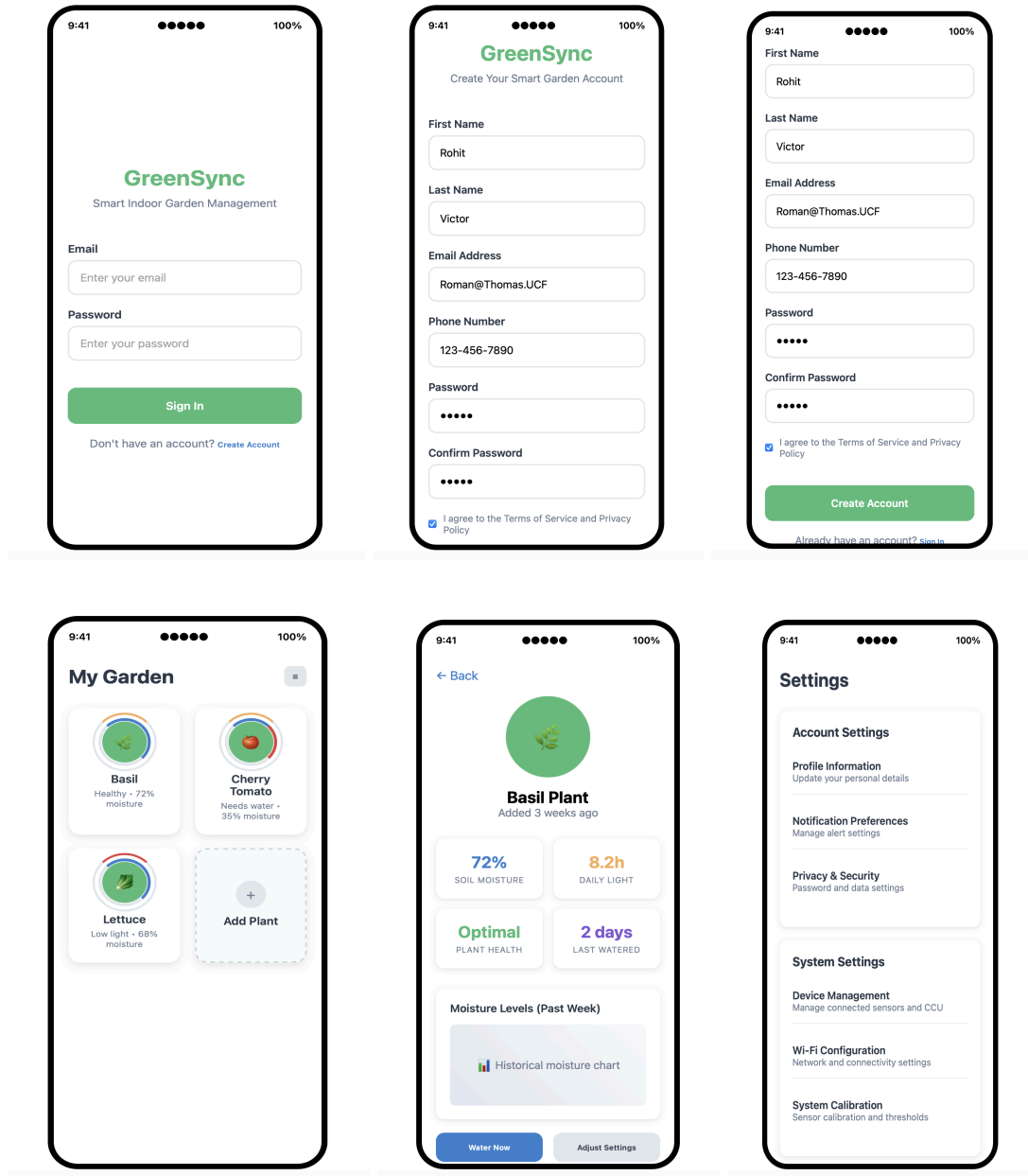


Figure 19: Users Mobile Interface

7.3 System Calibration and Constants

Adhering to modern software principles means avoiding hard coded values whenever possible, as they can make software brittle and difficult to maintain. However, for the GreenSync system, specific constants will need to be embedded in the firmware to serve as essential calibration benchmarks that

allow the software to accurately interpret messy, real world data from the physical environment.

7.3.1 Soil Moisture Calibration Constants

In this system, hard coded constants will be necessary to translate the raw analog readings from the soil moisture sensors into a meaningful and standardized data format. Rather than a simple linear scale derived from two points, the system is designed to use a more sophisticated multi point calibration curve. Through empirical testing, raw sensor values will be recorded for several known soil states (for example, "fully dry," "damp," and "ideally saturated"). These data points, representing the unique response of the sensor hardware, will be stored in the firmware as an array. During operation, the ESP32 will use mathematical interpolation between these known calibration points to calculate a more accurate, non linear moisture percentage from any given raw reading. This method will provide a much more precise interpretation of soil conditions across the full moisture spectrum.

While formal testing has yet to be completed, it is hypothesized that the raw analog value for fully dry soil might be near 3100, while the value for ideally saturated soil could be in the 700 to 900 range. These values will be finalized during the hardware testing and calibration phase of the project.

7.3.2 Network Communication Constants

The system will leverage the built in Wi Fi capabilities of the ESP microcontrollers to create a private, low latency local communication network. Rather than relying on a traditional router based network that would add complexity and potential points of failure, the system will use a direct device to device communication method. This approach utilizes the Wi Fi hardware to send small, efficient data packets from the sensors directly to the Central Control Unit. This method is ideal for sending small packets of data with very low latency, which is perfect for real time sensor monitoring.

To establish a secure and manageable network, the system will use a dynamic pairing process rather than hard coding static MAC addresses. This process will be initiated by the user from the mobile application. When "Pairing Mode" is activated, the CCU will broadcast a special "discovery" message over its Wi Fi interface. New, un-paired sensors will be programmed to listen for this specific message. Upon receiving it, they will respond by transmitting their unique MAC address directly to the CCU. The CCU will then

relay this information to the mobile app, which will display a list of newly discovered sensors. The user can then select a sensor and assign it to a plant, which securely adds its MAC address to a "whitelist" stored in the CCU's persistent memory. This method creates a robust and flexible system, allowing users to easily add or replace sensors without ever needing to modify and re-flash the firmware.

Chapter 8: System Fabrication

8.1 Layout of the PCB

The Printed Circuit Board (PCB) design for this project is organized into several dedicated modules, each serving a specific electrical or functional purpose. This modular architecture enables targeted testing, streamlined troubleshooting, and reduced hardware risk during iterative development. The system components—including the power regulation circuitry, the ESP32 microcontroller subsystem, and the sensor/actuator peripherals—are designed as independent yet interoperable units that integrate into the larger system framework.

All PCB modules remain in active development and are subject to revision pending performance testing and advisor approval. The following sections detail the design intent, layout considerations, and current development status for each major subsystem.

8.1.1 Board for the 3.3V Regulator

The dedicated board for the 3.3 V regulator provides low-noise power to the ESP32 and all logic-level subsystems. The design was created using Fusion360 and includes optimized routing for thermal efficiency and component density. Capacitor placement and ground plane continuity were prioritized to maintain regulator stability and reduce voltage ripple during high-load operation.

Thermal behaviors, load-transient response, and mechanical clearance constraints remain under review. Additional refinement is expected prior to fabrication to ensure robust, dependable performance for the system's digital electronics.

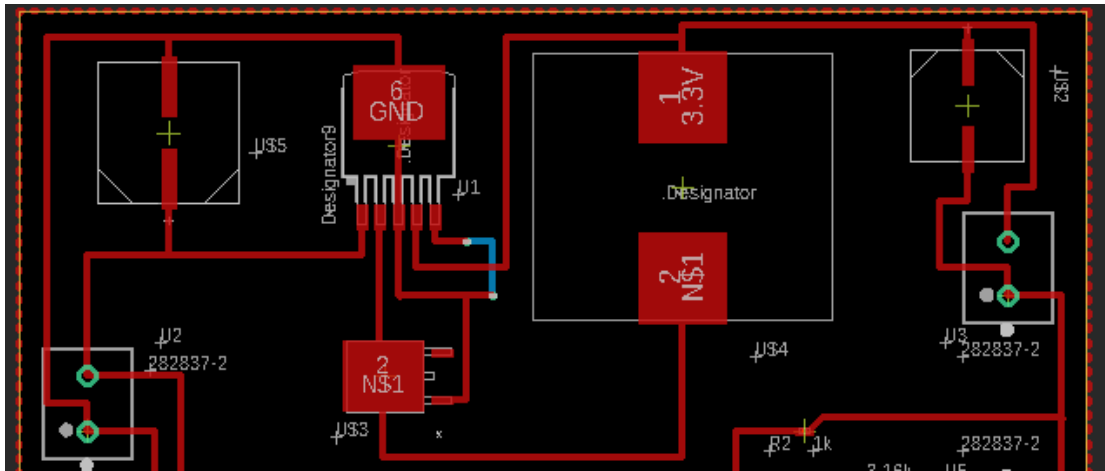
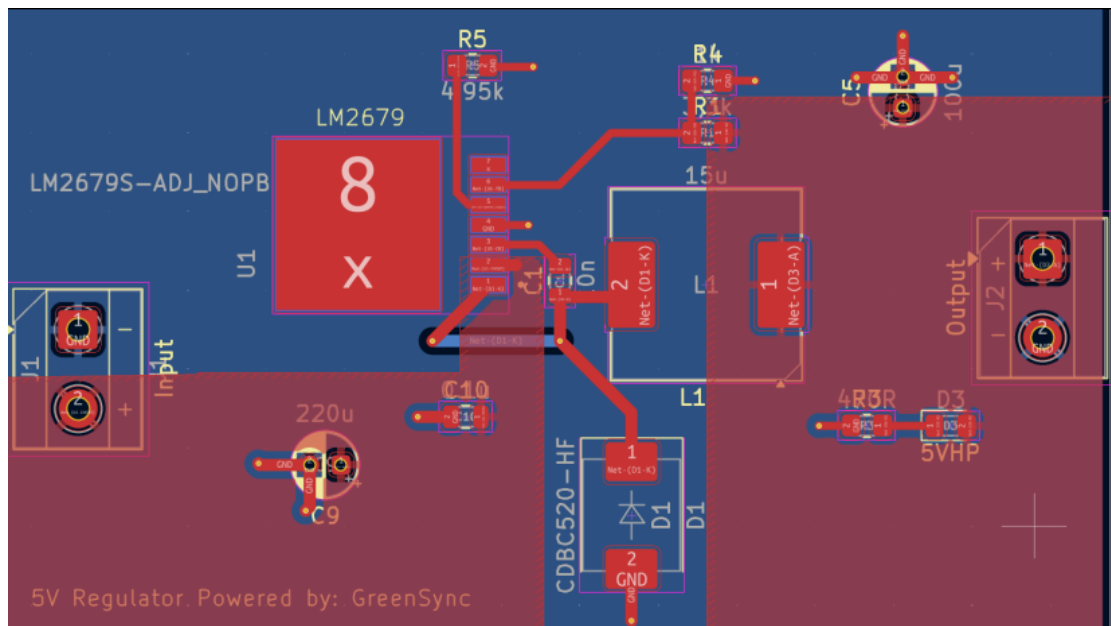


Figure 20 Version 2: Regulators Designed Through Fusion360 (1)



The 5V regulator board designed for GreenSync is built around the **LM2679S-ADJ**, a high-efficiency buck converter capable of supplying up to 5A of continuous current—more than enough to power the WS2812B LED strip, relay modules, sensors, and any auxiliary 5V peripherals. The PCB layout emphasizes proper switching-regulator design practices, including a short, low-impedance loop between the diode (CDBC520-HF), inductor, and output capacitor to reduce noise and maintain stable regulation. A 220 µF input capacitor and 100 µF output capacitor help smooth voltage ripple, while

the adjustable output network (R4/R5) allows fine tuning of the 5V rail during testing. Thermal performance is improved through wide copper pours on both the input and output planes, ensuring even heat dissipation across the board. By following the recommended layout guidelines from the LM2679 datasheet and incorporating clear labeling for input and output terminals, this regulator board provides a reliable and robust 5V supply tailored specifically to the power demands of GreenSync.

8.1.3 Microcontroller PCB (ESP32-WROOM Module)

The microcontroller board serves as the primary computational unit for the system, providing processing, wireless connectivity, and control signal routing. The PCB layout includes carefully managed trace geometries and ground-plane partitioning to minimize electromagnetic interference—an important consideration due to the ESP32's high-frequency RF transmission characteristics.

Component placement was optimized to maintain short, direct signal paths for I²C, ADC, and UART lines, ensuring stable communication with all peripheral devices. Thermal considerations, particularly those related to Wi-Fi and Bluetooth operations at peak load, informed the spacing and copper pour design.

This board remains under active development, with several layout features awaiting verification through simulation and advisor review. Modifications may be necessary once the power module undergoes integrated testing.

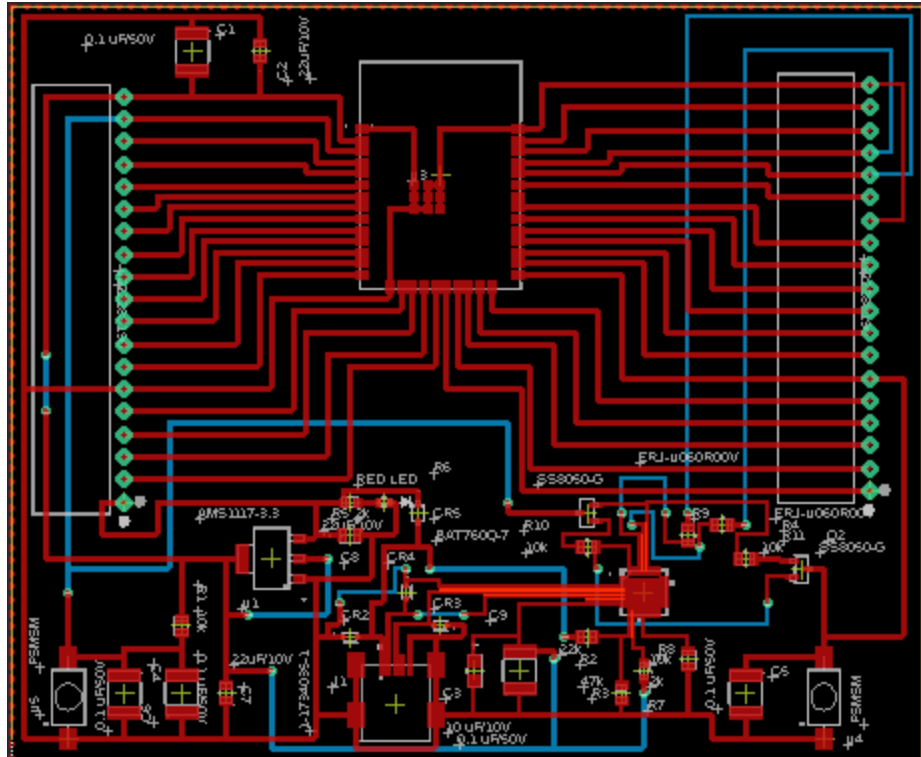


Figure ESP32 Version 2

8.1.4 Sensor and Peripheral Module PCB

The prototype PCB integrates multiple sensors and peripheral devices directly onto the board to streamline wiring and reduce external mechanical complexity. The following components are incorporated into this subsystem:

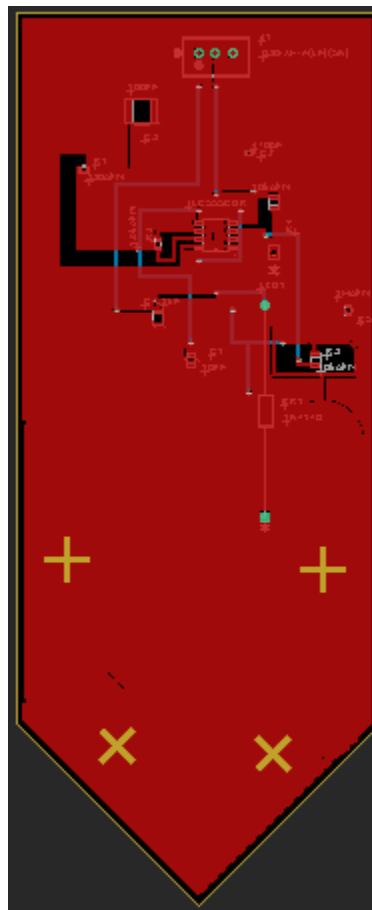
- **VEML7700 Ambient Light Sensor**
Mounted with short trace lengths to minimize noise on the I²C lines. Orientation and placement ensure proper light exposure for accurate environmental readings.
- **DFRobot Capacitive Soil Moisture Sensor Interface**
Input circuitry includes filtering and stabilization measures to ensure accurate ADC readings from soil moisture measurements.
- **AI-Thinker ESP32-CAM**
Provided with dedicated power routing and noise isolation to prevent RF interference affecting the main microcontroller or sensor signals.
- **5V Relay Module**
Integrated with proper flyback protection and spacing requirements to

isolate high-current switching noise from low-voltage digital circuits.

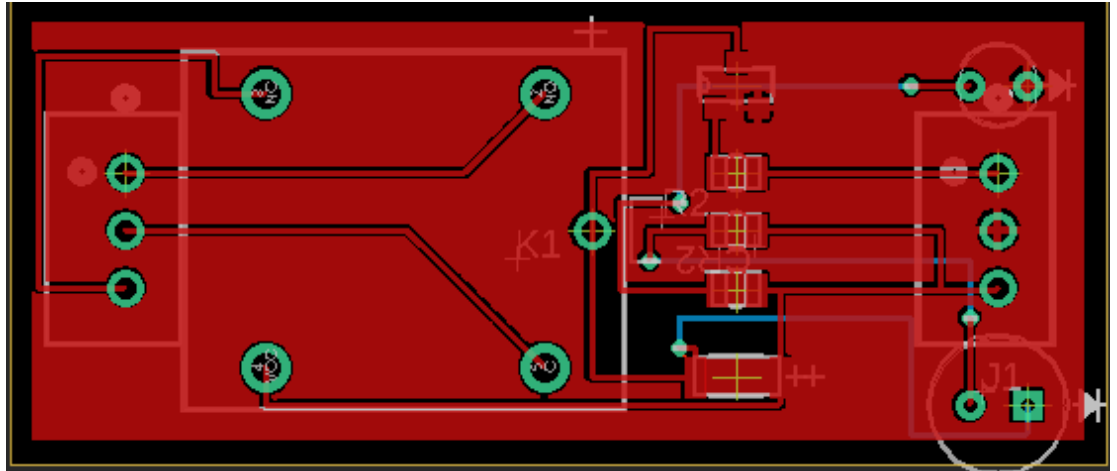
- **Battery Holder and Power Input Routing**

Designed to accommodate the system's portable power source and provide direct feed to the regulation module.

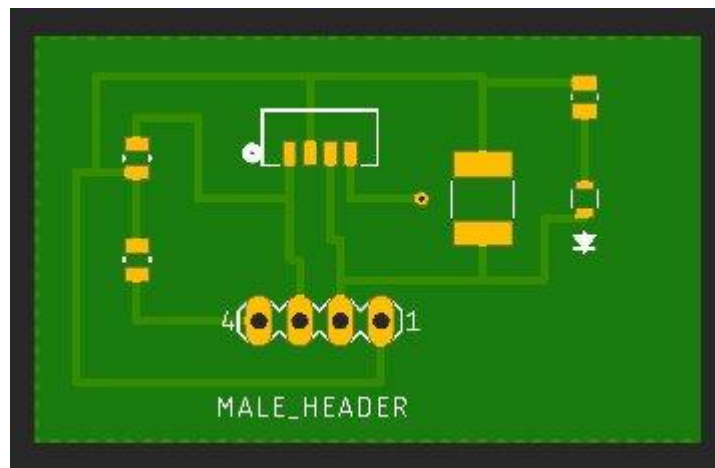
Each of these components is placed with consideration for signal integrity, mechanical stability, and electrical isolation. Further refinement will be completed as testing progresses and final mechanical constraints are defined.



Moisture Sensor



5V Relay



Light Sensor

Chapter 9: System Testing and Evaluation

With the core components of the GreenSync system now acquired, initial testing and integration efforts have begun to verify individual functionality and ensure system compatibility. The primary focus this semester has been on validating the basic operation of each major hardware unit: the ESP32 microcontroller, ESP32-CAM with onboard camera, capacitive soil moisture sensors, and VEML7700 ambient light sensors. Each of these subsystems plays a crucial role in enabling automated irrigation and environmental monitoring, forming the backbone of GreenSync's smart gardening framework. This chapter outlines the preliminary tests conducted on each component to confirm basic functionality and prepare for full system

integration. While this semester emphasized hardware verification and sensor communication, the following semester will focus on assembling the full system, refining logic control, and developing the software interface to tie all modules together.

9.1 ESP32 Testing

As one of the core components of the GreenSync system, the ESP32-WROOM-32 development board was among the first modules to undergo preliminary testing. This board is responsible for managing sensor inputs, executing logic for water delivery decisions, and facilitating communication with the mobile application. Given the critical role it plays in system coordination, verifying its operational status was a top priority.

Initial testing began by connecting the ESP32 to a laptop via USB and configuring it within the Arduino IDE. The correct board profile and COM port were selected, and the appropriate USB-to-serial driver (CH340 or CP210x, depending on the variant) was installed. A basic "Blink" program was uploaded to the board to toggle the onboard blue LED at regular intervals. This simple test served as confirmation that the ESP32 was successfully recognized by the host system, and that it could be programmed, reset, and powered without issue.

In addition to blinking the LED, the test verified serial communication through the Arduino IDE's Serial Monitor, which showed real-time debug output from the microcontroller. This confirmed the reliability of the USB interface and the proper configuration of the ESP32's UART pins. The internal timing of the ESP32 was also confirmed to be accurate, as the LED blinked at the expected 1-second intervals, validating the board's onboard clock source.

Although more advanced features such as Wi-Fi connectivity, sensor interfacing, and multitasking will be developed and tested in the next project phase, these initial checks confirmed the development board's readiness for further integration. This foundational step ensures confidence in the hardware and development environment moving forward, and lays the groundwork for upcoming work involving real-time sensor polling, actuator control, and wireless data transmission.

9.2 Moisture Sensor Testing

One of the most critical features of the GreenSync system is its ability to determine soil hydration levels in real-time. To test this capability, we wired a

capacitive soil moisture sensor to the ESP32 development board using a breadboard and jumper wires. The ESP32 was connected to a laptop via USB, and the Arduino IDE was used to upload basic test code that read the sensor's analog signal and printed the raw ADC output to the serial monitor.

The readings from the sensor responded as expected during various moisture conditions. When exposed to dry air or wiped clean with a paper towel, the sensor produced higher raw values, typically in the range of **2800–3100**. When the sensor was partially submerged in moist soil or water, the raw values decreased proportionally. At full submersion in water, the readings dropped to around **800**, indicating strong responsiveness to moisture presence. A half-damp condition (such as light misting or partial soil contact) yielded intermediate values around **2000**, demonstrating that the sensor can detect a clear gradient in soil moisture.

This testing confirmed that the sensor is suitable for differentiating between dry, moist, and saturated conditions. These raw ADC readings will later be calibrated and used in the control logic for triggering watering events based on predefined thresholds.

Table 38: Moisture Test

CONDITION	SOIL MOISTURE RAW (ADC VALUE)	NOTES
Dry (clean/towel dry)	2900–3100	Sensor out of soil
Slightly Moist	2000–2300	Sensor partially inserted in soil
Fully Wet	800–1000	Sensor submerged in water

9.3 Ambient Light Sensor Testing

To verify GreenSync's ability to monitor environmental lighting conditions, we tested the functionality of the ambient light sensor using a standard bench setup. The sensor was wired to a breadboard and connected to the ESP32 microcontroller, which was in turn linked to a laptop via USB. We used the Arduino IDE to upload test code that read and displayed real-time light intensity measurements in lux.

Once the sensor was powered and initialized correctly, we conducted several simple lighting tests to observe how the sensor responded to varying levels of illumination. Under normal room lighting conditions, the sensor consistently reported lux values in the range of **200–300**. When the sensor was covered by a hand or piece of cloth, the readings dropped significantly to near **0–50**

lux, confirming its ability to detect low-light or shaded environments. To simulate a high-light scenario, a cell phone flashlight was pointed directly at the sensor, causing the values to spike to **5000–6000 lux**.

These tests demonstrate that the light sensor is functioning as expected and can reliably distinguish between varying lighting conditions. This capability will later be used in the system to make decisions such as toggling grow lights or logging environmental changes for plant health tracking.

Table 39: Light Testing

LIGHTING CONDITION	OBSERVED LUX VALUE	NOTES
Normal Room Light	200–300 lux	Ambient indoor lighting
Covered Sensor	0–50 lux	Sensor shaded or blocked from light
Flashlight Shined on Sensor	5000–6000 lux	Simulated bright, direct illumination

9.4 Camera Module Testing

The ESP32-CAM development board plays a supporting yet valuable role in the GreenSync system, enabling visual monitoring of plant health and environmental conditions. The module features an onboard OV2640 camera sensor and offers both Wi-Fi and microSD support, making it ideal for lightweight image capture tasks within an embedded system.

For initial testing, the ESP32-CAM was connected directly to a laptop using a USB-to-serial adapter with 5V and GND lines supplied from the laptop itself. Once connected, the board was flashed using the Arduino IDE with example code from the ESP32-CAM library, specifically the "CameraWebServer" sketch provided in the ESP32 examples. After successful upload and reset, the board hosted a local IP address accessible via any web browser on the same Wi-Fi network.

Navigating to the board's IP address opened a web-based user interface that streamed live video from the onboard camera. The video stream allowed for real-time adjustments to camera settings such as resolution, contrast, and brightness. A snapshot feature was also available, which was used to capture still images during testing. These snapshots served as a quick proof-of-functionality and confirmed that the image quality was suitable for the project's needs. The testing process was documented via screen recordings and shared among the team for internal review.

Although no integration with the main ESP32 board has occurred yet, the test confirmed that the ESP32-CAM module was fully operational, capable of streaming video, and configurable through a lightweight user interface. Future work will involve integrating this video feed into the mobile application or potentially saving snapshots to a microSD card for offline analysis or documentation.

This successful validation of the ESP32-CAM's core functionality ensures that the component is ready for more advanced implementation in the next semester, where it will serve as the visual intelligence layer of the GreenSync system.

9.5 🌐 Plan For Senior Design 2

With individual hardware components now verified for basic functionality, the next semester will shift the focus toward full system integration and performance testing of the complete GreenSync smart gardening platform. The primary objective is to combine the core modules—ESP32 microcontroller, ambient light sensor, soil moisture sensor, water pump, and camera—into a cohesive, reliable system capable of autonomously monitoring and responding to the plant's environmental needs. This stage will emphasize not only operational correctness but also system efficiency, data accuracy, and real-world usability.

ESP32 Integration:

The ESP32 will act as the system's central hub, orchestrating sensor readings, actuator commands, and wireless communication. The firmware will be expanded to handle continuous polling of soil moisture and ambient light data, executing conditional logic to activate watering routines or light adjustments as needed. In the next phase, we plan to evaluate the ESP32's multitasking performance using FreeRTOS to schedule tasks efficiently, ensuring that no sensor data is missed during concurrent operations. Additionally, memory usage and power management features will be tested to optimize long-term operation.

Soil Moisture and Water Pump System:

The moisture sensor and mini pump will be integrated as a closed feedback loop. If the soil moisture falls below a pre-defined threshold, the ESP32 will trigger the pump to deliver a calculated volume of water. Testing here will involve calibrating the system to find optimal moisture thresholds for various soil types, determining the pump's response time, and verifying that moisture

readings stabilize after irrigation. We'll log each event and analyze time-to-recovery metrics to evaluate watering efficiency. Furthermore, long-term reliability tests will be performed to ensure the sensor does not degrade from repeated moisture exposure.

Ambient Light Sensor and Lighting Control:

The ambient light sensor will be used to assess daily light exposure and determine whether artificial grow lights need to be activated. In the upcoming semester, we will integrate either a MOSFET relay or PWM-controlled LED driver to automate lighting response based on lux readings. Tests will be conducted to confirm smooth transitions, proper lux thresholds, and safe operation of the lights. We will also observe whether light levels affect moisture evaporation, introducing an opportunity for cross-sensor calibration.

Camera Module Testing:

The onboard camera will be used to periodically capture plant images for health monitoring or remote viewing via the mobile app. Future testing will involve confirming the image capture trigger works in sync with other events (e.g., after watering or at set intervals), storing or transmitting the data efficiently, and optionally compressing images to conserve memory. If bandwidth and memory allow, we'll explore lightweight image processing or AI tagging methods as stretch goals.

Mobile App Integration:

Although basic app development may remain minimal during this phase, we plan to ensure BLE or Wi-Fi-based communication is functional for syncing plant data and receiving user commands. UI/UX testing will be done using mock data from the ESP32 to confirm a smooth display of moisture, light, and status logs.

System-Wide Evaluation:

Once all modules are connected, a full-system dry run will be conducted over several days to observe reliability, performance bottlenecks, and possible sensor conflicts. Key metrics to evaluate will include system uptime, responsiveness, false triggers, power consumption, and long-term data consistency. Each subsystem will be stress-tested under edge cases (e.g., low power, rapid lighting changes, or fluctuating moisture conditions) to validate robustness.

In summary, the next semester will focus on finalizing the GreenSync system from a collection of independent modules into a fully functional, intelligent automation platform. All testing efforts will feed directly into iterative

improvements to hardware wiring, software logic, and real-time control algorithms.

Chapter 10: Administrative Content

10.1 Component Comparison

Below we will list the items and components we will need during this project but are subject to change as we personally have never built something like this before. Below will be the main components and we will add a plus and minus buffer range in case we need to add or remove any of the components.

Table 40: Bill of Materials

CATEGORY	COMPONENT	DESCRIPTION	QTY	ESTIMATED COST (USD)
 Control Unit	ESP32-WROOM-32 Dev Kit	Main microcontroller with Wi-Fi and Bluetooth	1	\$5.33
 Power Supply	12V Wall Adapter Power Supply	Powers pump, lights, and controller system	1	\$8.00
 Moisture Sensor	Capacitive Soil Moisture Sensor (AITRIP)	Raw soil moisture detection (connected to ESP32)	1	\$1.20
 Water Pump	Adafruit Peristaltic Liquid Pump	Precision water delivery with flexible tubing	1	\$14.95
 Release Valve	Adafruit Mini Liquid Solenoid Valve	Opens water path from reservoir on command	1	\$6.95
 Reservoir	Reliance Aqua-Tainer (7 Gallon)	Water storage container with tap	1	\$17.00
 Light Sensor	VEML7700 Ambient Light Sensor	Measures ambient light levels (in lux)	1	\$4.79
 Grow Light	Roleadro LED Grow Light Panel	Supplemental plant lighting (Red/Blue spectrum)	1	\$24.99
 Camera Module	ESP32-CAM (OV2640 + MB Board)	Captures plant images and stores to MicroSD	1	\$13.99
 Storage	MicroSD Card (16 GB)	Image storage and potential data logging	1	\$6.00
 Mobile App	Custom Android/iOS App	Displays sensor readings, light and watering control	1	Free (custom built)
 Enclosure	Wooden Box / Enclosure	Holds and organizes all hardware components	1	\$7.00
 Soil	Potting Mix	Medium for planting and soil moisture testing	1 bag	\$5.00
 Tools/Dev	KiCad PCB design and schematics (open source)	–	–	Free

\$ Total Estimated Cost: ≈ \$114.20

Note: These are a very rough estimate, and it is assumed that costs will end up being significantly more.

10.2 Project Milestones

The timeline below is an estimation of how things will go for our group. They are subjected to change in the case of certain due dates that are assigned during this semester and the fall semester for Senior Design 2. Below is a pseudo schedule that we hope to follow in order to progress smoothly through this project.

Table 41: Project Milestones Table

Semester	Week(s)	Milestone
Summer	Weeks 1–2	Define project scope, goals, and deliverables; assign team roles
Summer	Weeks 3–4	Conduct research on plant needs, soil sensors, microcontrollers, and touchscreen options
Summer	Week 5	Compare component options and finalize bill of materials
Summer	Week 6	Order essential components and test soil sensor readouts using ESP32
Summer	Week 7	Create block diagrams and software architecture plans
Summer	Week 8	Write pseudocode for core features and interface design layout
Summer	Week 9	Draft early documentation and prepare progress presentation
Summer	Finals Week	Submit Summer Research Summary + Preliminary System Design
Fall	Weeks 1–2	Begin full system integration: sensors, pumps, UI, and ESP32–Pi communication
Fall	Week 3	Add real-time scheduling (RTC module or Pi time)
Fall	Week 4	Implement sensor-driven watering logic with live testing
Fall	Week 5	Build touchscreen interface with manual override features
Fall	Week 6	Add grow light control (optional stretch goal)
Fall	Week 7	Assemble full prototype in waterproof housing
Fall	Week 8	Run multi-day system validation tests
Fall	Week 9	Optimize logic, calibration, and UI feedback
Fall	Week 10	Prepare Final Report, Poster, and User Guide
Fall	Week 11	Polish final demo and presentation
Fall	Finals Week	Deliver Final Presentation + Submit Completed System

10.3 Workflow Distribution

Table 42: Work Distribution Table

Chapter 11: Conclusion

The development of GreenSync marks a significant step toward integrating automation and sustainability into everyday plant care. From the initial concept to the detailed design and implementation phases, this project has focused on creating a reliable, scalable, and user-friendly smart garden system. By combining sensors, actuators, and microcontroller-based control with wireless connectivity, GreenSync offers users a practical solution for maintaining healthy plants without the constant need for manual intervention.

Throughout this project, careful attention was given to balancing functionality, cost-effectiveness, and ease of integration. The system's foundation lies in its ability to sense critical environmental factors such as soil moisture and ambient light and respond dynamically by activating watering systems and supplemental lighting when necessary. This approach not only promotes plant health but also ensures resource efficiency, minimizing water and energy waste. These features make GreenSync particularly well-suited for individuals who want the benefits of a smart garden without the complexity or high cost often associated with commercial solutions.

The research and design process underscored the importance of exploring multiple technological pathways before finalizing component choices. Each subsystem, whether it was water delivery, light control, or camera-based monitoring was evaluated against a range of alternatives to ensure the best fit for GreenSync's objectives. This structured methodology allowed the team to justify each decision with a clear understanding of the trade-offs involved. For instance, choosing a capacitive soil moisture sensor over resistive types ensured better long-term durability, while the selection of an ESP32-based microcontroller enabled seamless wireless connectivity and efficient control logic within a compact footprint.

While GreenSync was designed as a single-plant prototype, its architecture was built with scalability in mind. The use of modular components, wireless protocols, and flexible software frameworks positions the system to expand into multi-plant or even greenhouse-scale implementations in the future. Additionally, provisions for camera integration and data logging lay the groundwork for advanced features such as image-based health diagnostics and cloud-connected analytics, which could transform the system from a simple automation tool into a comprehensive plant management platform.

Beyond technical achievements, this project demonstrates the practical value of applying IoT principles to sustainable living. Automating irrigation and lighting not only saves time for users but also promotes responsible resource management, aligning with broader environmental goals. As smart technologies continue to evolve, projects like GreenSync illustrate how

accessible, low-cost solutions can make a tangible impact on everyday tasks, bridging the gap between convenience and ecological responsibility.

In conclusion, GreenSync successfully delivers on its mission to simplify and enhance plant care through automation. By leveraging modern microcontroller technology, precise sensing, and intelligent control systems, it offers a strong foundation for future innovation in home gardening. With potential for integration into larger ecosystems and smart home platforms, GreenSync represents not just a prototype, but a stepping stone toward the next generation of sustainable, tech-driven agriculture.

Appendix A: References and Data Sheets

[1] D. Patterson and J. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, 5th ed. San Francisco, CA: Morgan Kaufmann, 2013.

[2] Raspberry Pi Foundation, “What is a Raspberry Pi?” [Online]. Available: <https://www.raspberrypi.com/about/>

[3] Espressif Systems, “ESP32 Technical Reference Manual,” 2022. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

[4] S. Trimberger, “Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology,” *Proceedings of the IEEE*, vol. 103, no. 3, pp. 318–331, 2015.

[5] A. Tanenbaum and T. Austin, *Structured Computer Organization*, 6th ed. Boston, MA: Pearson, 2013.

[6] Arduino, “Arduino Uno Rev3,” [Online]. Available: <https://store.arduino.cc/products/arduino-uno-rev3>

[7] Raspberry Pi Foundation, “Raspberry Pi Pico,” [Online]. Available: <https://www.raspberrypi.com/products/raspberry-pi-pico>

[8] Espressif Systems, “ESP8266EX Datasheet,” [Online]. Available: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

[9] STMicroelectronics, “STM32F103C8T6 Datasheet,” [Online]. Available: <https://www.st.com/en/microcontrollers-microprocessors/stm32f103c8.html>

[10] Espressif Systems, “ESP32 Series Datasheet,” [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

- [11] Silicon Labs, "AN118: Crystal Oscillator Design Considerations for MCUs," [Online]. Available: <https://www.silabs.com/documents/public/application-notes/AN118.pdf>
- [12] Digi-Key Electronics, "Crystal Oscillator Modules," [Online]. Available: <https://www.digikey.com/en/products/filter/oscillators/159>
- [13] Murata Manufacturing Co., Ltd., "Ceramic Resonators Technical Guide," [Online]. Available: <https://www.murata.com/en-us/products/timingdevice/resonator/basic/techguide>
- [14] STMicroelectronics, "Internal Oscillators in STM32 MCUs," [Online]. Available: https://www.st.com/resource/en/application_note/an2867-internal-rc-oscillators-stmicroelectronics.pdf
- [15] Microchip Technology Inc., "Understanding and Using Internal Oscillators," [Online]. Available: <https://ww1.microchip.com/downloads/en/appnotes/00687b.pdf>
- [16] Adafruit Industries, "Power Supplies: Wall Adapters," [Online]. Available: <https://learn.adafruit.com/power-supplies>
- [17] Texas Instruments, "Battery Management Overview," [Online]. Available: <https://www.ti.com/power-management/battery-management/overview.html>
- [18] Energizer, "Battery Basics: Alkaline vs. NiMH Rechargeable Batteries," [Online]. Available: <https://www.energizer.com/battery-basics>
- [19] SparkFun Electronics, "Powering Your Project," [Online]. Available: <https://learn.sparkfun.com/tutorials/powering-your-project>
- [20] National Renewable Energy Laboratory, "Solar Photovoltaic Technology Basics," [Online]. Available: <https://www.nrel.gov/research/re-photovoltaics.html>
- [21] Texas Instruments, "Understanding Linear Voltage Regulators," [Online]. Available: <https://www.ti.com/lit/an/slva079/slva079.pdf>
- [22] Analog Devices, "Buck Converter Design Considerations," [Online]. Available: <https://www.analog.com/en/analog-dialogue/articles/buck-converter-design-considerations.html>
- [23] SparkFun Electronics, "Voltage Regulators," [Online]. Available: <https://learn.sparkfun.com/tutorials/voltage-regulators>

[24] Digi-Key Electronics, "Understanding Buck-Boost Converters," [Online]. Available:

<https://www.digikey.com/en/articles/understanding-buck-boost-converters>

[25] T. Yadav, "Automated drip irrigation system using IoT," *International Journal of Creative Research Thoughts*, vol. 8, no. 5, pp. 232–235, May 2020.

[26] SparkFun Electronics, "Soil Moisture Sensor Hookup Guide," [Online]. Available:

<https://learn.sparkfun.com/tutorials/soil-moisture-sensor-hookup-guide>

[27] Adafruit Industries, "Mini Liquid Solenoid Valve," [Online]. Available: <https://www.adafruit.com/product/997>

[28] R. Singh, "Passive watering techniques in smart agriculture," *International Journal of Agricultural Research*, vol. 13, no. 1, pp. 12–17, Jan. 2021.

[29] H. Xu, L. Zhang, and S. Zhang, "Design of mist-based irrigation system for greenhouse environments," *Procedia Engineering*, vol. 29, pp. 2761–2765, 2012.

[30] D. Banerjee, "Capacitive Soil Moisture Sensor Design and Application," *IEEE Sensors Letters*, vol. 3, no. 1, pp. 1–4, Jan. 2019.

[31] S. Krishnan, "Resistive Soil Moisture Sensors: Benefits and Limitations," *Journal of Embedded Systems and Applications*, vol. 7, no. 2, pp. 58–62, Mar. 2020.

[32] G. Vellidis, "Use of Tensiometers in Irrigation Scheduling," University of Georgia Extension, Bulletin 1301, 2016.

[33] D. Noborio, "Measurement of soil water content and electrical conductivity by time domain reflectometry: A review," *Computers and Electronics in Agriculture*, vol. 31, pp. 213–237, 2001.

[34] D. A. Richards and M. W. Gardner, "Electrical resistance units for measuring soil-moisture," *Soil Science Society of America Journal*, vol. 7, no. 2, pp. 68–72, 1943.

[35] M. G. Villalobos and R. Fereres, "Root zone soil moisture monitoring using multisensor arrays," *Agricultural Water Management*, vol. 62, no. 3, pp. 163–180, 2003.

[36] AITRIP, "Capacitive Soil Moisture Sensor for Arduino and ESP32," *Product Datasheet*, 2023. [Online]. Available: <https://www.amazon.com>

[37] DFRobot, "SEN0193 Gravity: Analog Capacitive Soil Moisture Sensor-Corrosion Resistant," *DFRobot*, 2023. [Online]. Available: <https://www.dfrobot.com/product-1385.html>

[38] M. Kertesz, "Chirp – a capacitive soil moisture sensor with I2C interface,"

- Tindie Marketplace*, 2022. [Online]. Available: <https://www.tindie.com/products/miceuz/i2c-soil-moisture-sensor/>
- [39] Vegetronix Inc., “VH400 Soil Moisture Sensor,” *Vegetronix Technical Specifications*, 2022. [Online]. Available: <https://www.vegetronix.com/Products/VH400/>
- [40] SparkFun Electronics, “Soil Moisture Sensor Kit - SEN-13322,” *Product Page*, 2023. [Online]. Available: <https://www.sparkfun.com/products/13322>
- [41] RS PRO, “Mini Submersible Water Pump, 3–12 V DC, JT-180A,” *RS Components*, 2023. [Online]. Available: <https://uk.rs-online.com/web/p/water-pumps/>
- [42] Adafruit, “Peristaltic Liquid Pump with Silicone Tubing,” *Adafruit Learning System*, 2023. [Online]. Available: <https://www.adafruit.com/product/1150>
- [43] Kamoer, “Micro Diaphragm Water Pump,” *Kamoer Fluid Tech*, 2022. [Online]. Available: <https://www.kamoer.com/micro-diaphragm-pumps>
- [44] SparkFun, “Brushless Centrifugal Water Pump - 5V,” *SparkFun Electronics*, 2023. [Online]. Available: <https://www.sparkfun.com/products/16304>
- [45] Seeed Studio, “Mini Vibration Water Pump (DC 3V),” *Product Datasheet*, 2022. [Online]. Available: <https://www.seeedstudio.com/Mini-Water-Pump.html>
- [46] Adafruit, “Peristaltic Liquid Pump with Silicone Tubing,” *Adafruit*, 2023. [Online]. Available: <https://www.adafruit.com/product/1150>
- [47] DFRobot, “Gravity: 12V DC Peristaltic Pump,” *DFRobot*, 2023. [Online]. Available: <https://www.dfrobot.com/product-1787.html>
- [48] Kamoer, “KFS-B Micro Peristaltic Pump,” *Kamoer Fluid Tech*, 2023. [Online]. Available: <https://www.kamoer.com/KFS-B-micro-pump.html>
- [49] U.S. Solid, “12V DC Electric Solenoid Valve Normally Closed,” *us.solidproducts.com*, 2023. [Online]. Available: <https://ussolid.com/>
- [50] DFRobot, “Mini Solenoid Valve for Liquid Control,” *DFRobot*, 2023. [Online]. Available: <https://www.dfrobot.com/>
- [51] Adafruit, “Latching Solenoid Valve - 12V,” *Adafruit*, 2023. [Online]. Available: <https://www.adafruit.com/product/1501>
- [52] Bürkert, “Type 2875 – Proportional Solenoid Valve,” *Bürkert Fluid Control Systems*, 2023. [Online]. Available: <https://www.burkert.com/>
- [53] OMWay, “Motorized Ball Valve 12V,” *OMWay Technology*, 2023. [Online]. Available: <https://www.omwaytech.com/>
- [54] Adafruit, “Mini Liquid Solenoid Valve - 12V - 1/2” Nominal,” *Adafruit*, 2023. [Online]. Available: <https://www.adafruit.com/product/997>
- [55] DFRobot, “Gravity: Mini Solenoid Valve - 12V,” *DFRobot*, 2023. [Online]. Available: <https://www.dfrobot.com/product-2124.html>

[56] U.S. Solid, “Mini DC 12V Solenoid Valve – Normally Closed,” *US Solid*, 2023. [Online]. Available: <https://ussolid.com/12v-dc-mini-solenoid-valve.html>

[57] Reliance Products, “Aqua-Tainer 4 Gallon Water Container,” *Reliance Products*, 2023. [Online]. Available: <https://www.relianceproducts.com/aqua-tainer>

[58] Adafruit, “2-Liter Reservoir with Cap - Compatible with Peristaltic Pumps,” *Adafruit*, 2023. [Online]. Available: <https://www.adafruit.com/product/1150>

[59] Active Aqua, “Hydroponic Reservoir, 2 Gallon,” *Hydrofarm Commercial*, 2023. [Online]. Available: <https://hydrofarm.com/p/AARES2>

[60] Sterilite, “1.5 Gallon Food Storage Container,” *Sterilite Corporation*, 2023. [Online]. Available: <https://www.sterilite.com>

[61] M. T. Tyree, “Light as an Environmental Factor for Plants,” *Botany Online*, University of Hamburg, 2022. [Online]. Available: <https://www1.biologie.uni-hamburg.de>

[62] J. Gerald, “LED Grow Light Placement Guide,” *Grow Light Central*, 2023. [Online]. Available: <https://growlightcentral.com>

[63] H. D. Rahman et al., “Smart Lighting Systems in Vertical Farming: Technologies and Challenges,” *Agricultural Engineering International*, vol. 25, no. 4, pp. 124–135, 2022.

[64] S. Young, “Best Clip-On Grow Lights for Indoor Plants,” *Indoor Gardening Reviews*, 2023. [Online]. Available: <https://indoorgardening.com>

[65] R. Villareal et al., “Dynamic LED Lighting in Smart Agriculture: A Review,” *IEEE Access*, vol. 10, pp. 101342–101358, 2022.

[66] A. R. Hossain et al., “Analysis of Light Dependent Resistors in Embedded Monitoring Systems,” *International Journal of Scientific Research in Science and Technology*, vol. 9, no. 4, pp. 122–126, 2022.

[67] T. W. Kim et al., “Photodiode-Based Light Monitoring Systems: Circuit Design and Application,” *IEEE Sensors Journal*, vol. 21, no. 15, pp. 16542–16550, 2021.

[68] B. Zhang, “Phototransistor Applications in Analog Light Detection Circuits,” *Journal of Microelectronics and Optoelectronics*, vol. 13, no. 2, pp. 33–39, 2023.

[69] TSL2561 Digital Light Sensor Datasheet, AMS-OSRAM, 2021. [Online]. Available: <https://ams-osram.com>

[70] A. Shih et al., “Design and Characterization of Multispectral Sensors for Precision Agriculture,” *IEEE Access*, vol. 9, pp. 80567–80575, 2021.

[71] Vishay Intertechnology, “VEML7700 Ambient Light Sensor Datasheet,” 2022. [Online]. Available: <https://www.vishay.com/docs/84323/veml7700.pdf>

[72] ROHM Semiconductor, “BH1750FVI Digital Ambient Light Sensor

- Datasheet,” 2011. [Online]. Available: <https://www.rohm.com/datasheet/BH1750FVI>
- [73] ams-OSRAM, “TSL2591 Light-to-Digital Sensor,” 2023. [Online]. Available: <https://ams-osram.com/products/sensors/light-sensors/tsl2591>
- [74] Texas Instruments, “OPT3001 Ambient Light Sensor Datasheet,” 2022. [Online]. Available: <https://www.ti.com/lit/ds/symlink/opt3001.pdf>
- [75] Gardener’s Supply Company, “Fluorescent Grow Lights: An Overview,” 2023. [Online]. Available: <https://www.gardeners.com/how-to/grow-light-guide>
- [76] Sunlight Supply, “Understanding HID Grow Lights,” 2022. [Online]. Available: <https://sunlightsupply.com/blogs/hid-grow-light-guide>
- [77] P. Masson, “Advantages of LED Grow Lights,” *Horticulture Tech Review*, vol. 8, no. 4, pp. 22–29, 2021.
- [78] T. Watson, “Are Incandescent Grow Lights Still Useful?” *Indoor Plant Journal*, vol. 5, no. 3, pp. 14–16, 2020.
- [79] Gavita Holland, “Plasma Lighting for Plant Growth,” 2022. [Online]. Available: <https://www.gavita.com/plasma-grow-lights>
- [80] M. Bernier, “Infrared Light and Plant Morphology,” *Journal of Plant Physiology*, vol. 32, no. 1, pp. 48–52, 2021.
- [81] Roleadro, “Roleadro LED Grow Light, 75W Full Spectrum Panel,” Amazon, 2023. [Online]. Available: <https://www.amazon.com/dp/B01IVQ96KY>
- [82] GE Lighting, “GE BR30 Full Spectrum LED Grow Light Bulb,” GE Lighting, 2022. [Online]. Available: <https://www.gelighting.com/grow-lights>
- [83] Spider Farmer, “SF-1000 Full Spectrum LED Grow Light,” 2024. [Online]. Available: <https://www.spider-farmer.com/products/sf-1000-led-grow-light>
- [84] Logitech, “C270 HD Webcam,” Logitech, 2023. [Online]. Available: <https://www.logitech.com>
- [85] Wyze, “Wyze Cam v3 with Color Night Vision,” Wyze Labs, 2023. [Online]. Available: <https://www.wyze.com>
- [86] Amcrest, “Pan Tilt Zoom (PTZ) Outdoor Security Camera,” Amcrest, 2024. [Online]. Available: <https://www.amcrest.com>
- [87] Espressif Systems, “ESP32-CAM Development Board,” 2023. [Online]. Available: <https://www.espressif.com/en/products/devkits/esp32-cam/overview>
- [88] Canon, “EOS Rebel T7 DSLR Camera,” Canon USA, 2024. [Online]. Available: <https://www.usa.canon.com>
- [89] Ai-Thinker, “ESP32-CAM Development Board,” 2023. [Online]. Available: <https://docs.ai-thinker.com/en/esp32-cam>
- [90] Arducam, “Arducam Mini Module Camera Shield with OV2640 for Arduino,” 2024. [Online]. Available: <https://www.arducam.com/product/arducam-mini-2mp-plus>

- [91] OmniVision Technologies, "OV5640 – 5MP Camera Sensor," 2023. [Online]. Available: <https://www.ovt.com/sensors/OV5640>
- [92] A. S. Tanenbaum and D. Wetherall, *Computer Networks*, 5th ed., Pearson, 2011.
- [93] J. H. Davies, *Bluetooth Low Energy: The Developer's Handbook*, Prentice Hall, 2012.
- [94] Zigbee Alliance, "Zigbee Specification," 2023. [Online]. Available: <https://csa-iot.org/all-solutions/zigbee/>
- [95] Semtech Corporation, "LoRa Technology," 2024. [Online]. Available: <https://www.semtech.com/lora>
- [96] Nordic Semiconductor, "nRF24L01+ Product Specification," 2023. [Online]. Available: <https://www.nordicsemi.com>
- [97] D. Harris and S. Harris, *Digital Design and Computer Architecture*, 2nd ed., Morgan Kaufmann, 2012.
- [98] NXP Semiconductors, "UM10204 I2C-bus specification and user manual," Rev. 6, Apr. 2014.
- [99] A. M. Sodagar, *Serial Communication Protocols in Embedded Systems*, IEEE Press, 2020.
- [100] Espressif Systems, "ESP32 Technical Reference Manual," 2024. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
- [101] T. Wilmshurst, *Designing Embedded Systems with PIC Microcontrollers*, 2nd ed., Elsevier, 2010.
- [102] Microchip Technology Inc., "Understanding EEPROM endurance for PIC microcontrollers," 2012.
- [103] Espressif Systems, "ESP32 Flash Memory Management Guide," 2023. [Online]. Available: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/storage/index.html>
- [104] M. Margolis, *Arduino IoT Cloud Cookbook*, Packt Publishing, 2022.
- [105] Fujitsu Semiconductor, "FRAM Technology Overview," 2020.
- [106] J. Davies, *USB Complete: The Developer's Guide*, Lakeview Research, 5th ed., 2015.
- [107] M. Banzi and M. Shiloh, *Getting Started with Arduino*, 3rd ed., Maker Media, 2014.
- [108] Espressif Systems, "ESP-IDF Programming Guide," [Online]. Available: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>
- [109] PlatformIO Labs, "PlatformIO: The Next Generation IoT Development Ecosystem," [Online]. Available: <https://platformio.org/>

[110] J. Ganssle, *The Art of Designing Embedded Systems*, 2nd ed., Newnes, 2008.

[111] G. Halfacree and B. H. Lutz, "Getting Started with MicroPython on the ESP32," *HackSpace Magazine*, Issue 30, pp. 36–45, 2021.

[112] Espruino, "JavaScript for Microcontrollers," [Online]. Available: <https://www.espruino.com/>

[113] MIT App Inventor, "Create Apps for Android," [Online]. Available: <https://appinventor.mit.edu>

[114] T. Margalef, *IoT Projects with ESP32*, Packt Publishing, 2022.

[115] Arduino, "Serial Monitor," [Online]. Available: <https://docs.arduino.cc/software/ide-v2/tutorials/ide-v2-serial-monitor>

[116] A. Banzi and M. Shiloh, *Getting Started with Arduino*, 3rd ed., Maker Media, 2014.

[117] KiCad, "Official KiCad EDA Website," [Online]. Available: <https://www.kicad.org>

[118] EasyEDA, "EasyEDA Online PCB Design & Simulation," [Online]. Available: <https://easyeda.com>

[119] Altium, "Altium Designer," [Online]. Available: <https://www.altium.com/altium-designer>

[120] Autodesk, "Fusion 360 Electronics," [Online]. Available: <https://www.autodesk.com/products/fusion-360/features/electronics>

[121] Autodesk, "EAGLE PCB Design Software," [Online]. Available: <https://www.autodesk.com/products/eagle>

[122] All3DP, "PLA vs ABS: What's the Difference?" [Online]. Available: <https://all3dp.com/1/pla-vs-abs-filament-3d-printing>

[123] Ponoko, "Choosing Between Acrylic and Polycarbonate," [Online]. Available: <https://www.ponoko.com/blog/make/acrylic-vs-polycarbonate>

[124] Adafruit, "Project Enclosures and Cases," [Online]. Available: <https://www.adafruit.com/category/17>

[125] CircuitDigest, "Wooden Enclosure for Electronics Projects," [Online]. Available: <https://circuitdigest.com/diy-projects/wooden-box-enclosure-for-arduino-project>

[126] OKW Enclosures, "Aluminum Enclosures for Electronics," [Online]. Available: <https://www.okwenclosures.com/aluminum-enclosures>

[127] Miracle-Gro, "Potting Mix vs. Potting Soil: What's the Difference?" [Online]. Available: <https://www.miraclegro.com/en-us/library/container-gardening/potting-mix-vs-potting-soil>

[128] Planet Natural, "Garden Soil vs. Potting Soil," [Online]. Available:

<https://www.planetnatural.com/garden-soil-vs-potting-soil>

[129] The Spruce, "How to Make Cactus Potting Soil Mix," [Online]. Available: <https://www.thespruce.com/cactus-soil-plant-care-5192978>

[130] Gardeners Supply Company, "Understanding Peat Moss and Sustainability," [Online]. Available: <https://www.gardeners.com/how-to/peat-moss-sustainable-alternatives>

[131] Epic Gardening, "What Is Coco Coir?," [Online]. Available: <https://www.epicgardening.com/coco-coir>

[132] IEC 60335-2-51, *Household and Similar Electrical Appliances – Safety – Part 2-51: Particular Requirements for Stationary Centrifugal Pumps for Heating and Service Water Installations*. International Electrotechnical Commission, Geneva, 2002.

[133] IEC 60335-2-55, *Household and Similar Electrical Appliances – Safety – Part 2-55: Particular Requirements for Electrical Appliances for Use with Aquariums and Garden Ponds*. IEC, Geneva, 2017.

[134] National Electrical Manufacturers Association, *NEMA 5-15 Configuration, Dimensional, and Performance Specifications for 125 V, 15 A Grounding Plugs, Receptacles, and Connectors*. Rosslyn, VA, 2015.

[135] ASTM D1785-21, *Standard Specification for Poly(Vinyl Chloride) (PVC) Plastic Pipe, Schedules 40, 80, and 120*. ASTM International, West Conshohocken, PA, 2021.

[136] IEC 60529, *Degrees of Protection Provided by Enclosures (IP Code)*. International Electrotechnical Commission, Geneva, 2013.

[137] NFPA 70, *National Electrical Code (NEC®)*. National Fire Protection Association, Quincy, MA, 2023.

[138] ISO 50001:2018, *Energy Management Systems—Requirements with Guidance for Use*. International Organization for Standardization, Geneva, 2018.

[139] IPC-2221B, *Generic Standard on Printed Board Design*. IPC – Association Connecting Electronics Industries, Bannockburn, IL, 2012.

[140] Bluetooth SIG, *Bluetooth Core Specification – Version 5.3*. Kirkland, WA, September 2021.

[141] IEEE Std 802.11-2020, *IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific*

Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE, Piscataway, NJ, 2020.

[142] ISO/IEC 20922:2016, *Information Technology—Message Queuing Telemetry Transport (MQTT) v3.1.1*. ISO/IEC JTC 1, Geneva, 2016.

[143] ISO/IEC 27001:2022, *Information Security, Cybersecurity and Privacy Protection—Information Security Management Systems—Requirements*. ISO/IEC JTC 1, Geneva, 2022.

[144] National Institute of Standards and Technology, *NISTIR 8259A: IoT Device Cybersecurity Capability Core Baseline*. Gaithersburg, MD, 2020.

[145] IEC 60598-1:2020, *Luminaires—Part 1: General Requirements and Tests*. International Electrotechnical Commission, Geneva, 2020.

[146] ISO 9241-303:2011, *Ergonomics of Human-System Interaction—Part 303: Requirements for Electronic Visual Displays*. ISO, Geneva, 2011.

[147] U.S. Environmental Protection Agency, *WaterSense® Program Guidelines*. Washington, DC, 2021.

[148] ISO 18400-102:2017, *Soil Quality—Sampling—Part 102: Selection and Application of Sampling Techniques*. ISO, Geneva, 2017.

[149] ISO 11074:2015, *Soil Quality—Vocabulary*. International Organization for Standardization, Geneva, 2015.

[150] ISO 772:2011, *Hydrometry—Vocabulary and Symbols*. ISO, Geneva, 2011.

[151] IEC 61960-3:2017, *Secondary Cells and Batteries Containing Alkaline or Other Non-Acid Electrolytes—Secondary Lithium Cells and Batteries for Portable Applications—Part 3: Prismatic and Cylindrical Lithium Rechargeable Cells and Batteries Made From Them*. International Electrotechnical Commission, Geneva, 2017.

[152] Alitove, “ALITOVE 12V 2A Power Supply Adapter,” ALITOVE, [Online]. Available: <https://alitove.com/products/alitove-12v-2a-dc-power-supply>

[153] Mean Well, “GS25A12-P1J 12V 2.08A Desktop Adapter,” Mouser Electronics, [Online]. Available: <https://www.mouser.com/ProductDetail/MEAN-WELL/GS25A12-P1J>

[154] UXCELL, “9V 2A Power Supply Adapter,” SunFounder Store, [Online]. Available:

<https://www.sunfounder.com/products/power-adapter-9v-2a-ac-adapter-power-supply-adapter>

[155] DROK, “Adjustable Buck Converter with 12V Adapter,” Amazon, [Online]. Available:

<https://www.amazon.com/DROK-Adjustable-Converter-Transformer-Protective/dp/B07JZ2GQJF>

[156] HiLetgo, “5V 2A USB Power Supply,” PTBN Electronics, [Online]. Available:

<https://www.ptbn.us/shop/Step-up-Cable-USB-To-DC-5-5x2-1mm-3-5x1-35mm-USB-Power-Supply-1034967>

Appendix B: Software Code

Section B1: Testing the Soil Moisture Sensor - Code from Arduino

```
#define SOIL_PIN 32          // Analog pin for soil sensor
#define LED_PIN 2           // Onboard blue LED on ESP32

void setup() {
  Serial.begin(115200);
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  // Read soil sensor
  int soilValue = analogRead(SOIL_PIN); // Range: 0 - 4095
  Serial.print("Soil Moisture Raw: ");
  Serial.println(soilValue);

  // Blink onboard LED for feedback
  digitalWrite(LED_PIN, HIGH);
  delay(100); // LED ON for 100ms
  digitalWrite(LED_PIN, LOW);

  delay(1000); // Wait 1 second before next reading
}
```

Section B2: Testing the ESP32-CAM development board - Code from Arduino

```
#define CAMERA_MODEL_AI_THINKER
#include "esp_camera.h"
#include <WiFi.h>

// Replace with your WiFi credentials
const char *ssid = "SpectrumSetup-*****";
const char password = "*****";

// Pin configuration for AI Thinker ESP32-CAM (OV2640)
#define PWDN_GPIO_NUM    32
#define RESET_GPIO_NUM   -1
#define XCLK_GPIO_NUM     0
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM     27

#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM       5

#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22

void startCameraServer(); // Forward declaration
void setupLedFlash();      // Optional, if needed

void setup() {
    Serial.begin(115200);
    Serial.setDebugOutput(true);
    Serial.println();

    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
```

```

config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sccb_sda = SIOD_GPIO_NUM;
config.pin_sccb_scl = SIOC_GPIO_NUM;
config.pin_pwn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;

if (psramFound()) {
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
    config.grab_mode = CAMERA_GRAB_LATEST;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
    config.fb_location = CAMERA_FB_IN_PSRAM;
}

Serial.println(psramFound() ? "PSRAM OK" : "PSRAM MISSING");

// Initialize camera
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
}

sensor_t *s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_QVGA); // Lower resolution for faster streaming

// Connect to WiFi

```

```

WiFi.begin(ssid, password);
WiFi.setSleep(false);

Serial.print("Connecting to WiFi");
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}

Serial.println("\nWiFi connected");
Serial.println("Starting camera server...");
startCameraServer();

Serial.print("Camera Ready! Use this URL to connect: http://");
Serial.println(WiFi.localIP());
}

void loop() {
    delay(10000); // Nothing to do here
}

// Web server code (built-in example from ESP32 library)
#include <WebServer.h>
#include "camera_pins.h"
#include "esp_timer.h"
#include "img_converters.h"
#include "Arduino.h"
#include "fb_gfx.h"
#include "esp_http_server.h"

```

Section B1: Testing the Ambient light sensor - Code from Arduino

```

#include "Adafruit_VEML7700.h"

Adafruit_VEML7700 veml = Adafruit_VEML7700();

void setup() {
  Serial.begin(115200);
  while (!Serial) { delay(10); }
  Serial.println("Adafruit VEML7700 Test");

  if (!veml.begin()) {
    Serial.println("Sensor not found");
    while (1);
  }
  Serial.println("Sensor found");

  // == OPTIONAL ==
  // Can set non-default gain and integration time to
  // adjust for different lighting conditions.
  // =====
  // veml.setGain(VEML7700_GAIN_1_8);
  // veml.setIntegrationTime(VEML7700_IT_100MS);

  Serial.print(F("Gain: "));
  switch (veml.getGain()) {
    case VEML7700_GAIN_1: Serial.println("1"); break;
    case VEML7700_GAIN_2: Serial.println("2"); break;
    case VEML7700_GAIN_1_4: Serial.println("1/4"); break;
    case VEML7700_GAIN_1_8: Serial.println("1/8"); break;
  }

  Serial.print(F("Integration Time (ms): "));
  switch (veml.getIntegrationTime()) {
    case VEML7700_IT_25MS: Serial.println("25"); break;
    case VEML7700_IT_50MS: Serial.println("50"); break;
    case VEML7700_IT_100MS: Serial.println("100"); break;
    case VEML7700_IT_200MS: Serial.println("200"); break;
    case VEML7700_IT_400MS: Serial.println("400"); break;
    case VEML7700_IT_800MS: Serial.println("800"); break;
  }

  veml.setLowThreshold(10000);
  veml.setHighThreshold(20000);
  veml.interruptEnable(true);
}

void loop() {
  Serial.print("raw ALS: "); Serial.println(veml.readALS());
  Serial.print("raw white: "); Serial.println(veml.readWhite());
  Serial.print("lux: "); Serial.println(veml.readLux());
}

```

```
uint16_t irq = veml.interruptStatus();
if (irq & VEML7700_INTERRUPT_LOW) {
    Serial.println("** Low threshold");
}
if (irq & VEML7700_INTERRUPT_HIGH) {
    Serial.println("** High threshold");
}
delay(500);
}
```

Appendix C: ChatGPT Prompts and Outcomes

[AI-1] Water Pump Comparison

- **Prompt:** "Evaluate compact 12V DC water pumps suitable for a smart plant watering system, specifically comparing a peristaltic pump (e.g., Kamoer KLP01), a small diaphragm pump (e.g., Shurflo 2088), and a brushless centrifugal pump (e.g., generic mini submersible). Focus on flow rate accuracy (ml/min), noise levels (dB), power consumption (Watts), self priming capability, and integration complexity with an ESP32 microcontroller for a target flow of 1.5 L/min."
- **Response:** "For your 1.5 L/min flow target in a compact 12V DC smart plant watering system, the Shurflo 2088 diaphragm pump presents a very strong candidate. It offers robust flow rates, typically achieving around 1.8 L/min at 12V... Crucially, it is a self priming pump, a significant advantage... However, diaphragm pumps inherently produce pulsatile flow... Peristaltic pumps... offer superior precision and consistency... but compact models often have lower maximum flow rates... Brushless centrifugal pumps are typically the quietest... A critical drawback is that they are generally not self priming... ESP32 integration for all three pump types typically involves a MOSFET switch... For diaphragm and centrifugal pumps, consider adding a flyback diode and an RC snubber circuit across the motor terminals to suppress EMF spikes..."

[AI-2] Sensor Battery Life Optimization

- **Prompt:** "For a capacitive soil moisture sensor PCB powered by a CR2032 coin cell battery and communicating with an ESP32 microcontroller via ESP NOW protocol, detail specific hardware and firmware strategies to achieve a minimum battery life of 180 days. Include recommended power down modes for both sensor and ESP32, optimal sampling intervals, and the impact of ESP NOW communication frequency on total power consumption. Provide a sample power calculation."
- **Response:** "To achieve a minimum battery life of 180 days with a CR2032 coin cell... the primary focus must be on aggressive power management. 1. ESP32 Deep Sleep:... It should spend the vast majority of its time in Deep Sleep mode... 2. Sensor Power Gating (Hardware Strategy): The capacitive soil moisture sensor must be completely de-powered (power gated) when not in use... Implement a low side MOSFET switch... 3. Optimal Sampling Intervals (Firmware Strategy):... sampling every 15 to 30 minutes, or even hourly, would be ideal... 4. ESP NOW Communication Efficiency:... Minimize the amount of data transmitted... Sample Power Calculation (Conceptual): ... Active current... 50mA (for ~500ms). Deep Sleep current = 20μA. Sample interval = 30 minutes... Estimated Battery Life = 220 mAh/0.816 mAh/day≈269 days."

[AI-3] PCB Corrosion Prevention Strategy

- **Prompt:** "Propose a multi layer protection strategy for exposed copper sensor traces on a PCB that will be buried in typical potting soil, ensuring long term stability of capacitive measurements. Consider material compatibility, ease of manufacturing, potential impacts on sensor accuracy, and cost effectiveness. Specifically, what would ChatGPT, DeepSeek, and Claude each propose, and how might their distinct solutions be combined for an optimal, robust strategy?"
- **Response (Synthesized from ChatGPT, DeepSeek, and Claude):** The combined research from the three platforms led to a synthesized strategy that would be more robust than any single AI's suggestion. The final theoretical design would incorporate:
 - **(from Claude)** Using a high TG FR 4 board material as a resilient foundation to resist moisture and warping.

- **(from DeepSeek)** Encapsulating the main body of the PCB in marine grade epoxy potting for a robust physical barrier against moisture.
- **(from ChatGPT)** Creating a precise sensing window where the exposed traces are protected with ENIG (Electroless Nickel Immersion Gold) plating for superior, localized corrosion resistance.
- **(from DeepSeek)** Using 316 stainless steel standoffs for any external metal contacts to prevent galvanic corrosion throughout the assembly.