PlantPulse

A Plant Health Monitoring System

Authors:

Kevin Shook
Computer Engineering

Joshua Barshay
Electrical Engineering

Fabrizio Martins
Computer Engineering

Review Committee:

Dr. Zakhia Abichar	ECE	Associate Lecturer
Dr. Mark Maddox	ECE	Adjunct Professor
Dr. Piotr Kulik	ECE	Assistant Professor

Table of Contents

TABLES

FIGURES

1 - EXECUTIVE SUMMARY

Is your home garden thriving or struggling? With PlantPulse, you'll never have to wonder again! Our innovative device uses advanced technology to monitor and analyze plant health metrics, providing you with a comprehensive picture of your plants' wellbeing. Say goodbye to trial and error - hello to successful gardening!

The art of home gardening is experiencing a resurgence in popularity, driven by an increasing desire for sustainability and self-sufficiency. However, monitoring plant health can be a daunting task, requiring manual measurements and interpretation of various metrics. This challenge can lead to frustration and abandonment of the hobby, even among those who are passionate about gardening.

PlantPulse aims to alleviate this issue by developing an innovative, all-in-one solution for monitoring plant health in home gardens. Our project combines cutting-edge technology with user-friendly design to provide gardeners with real-time insights into their plants' wellbeing. By leveraging advanced sensors and computer vision algorithms, PlantPulse will:

- *Monitor temperature, humidity, soil moisture, UV light, and overall plant health.*
- *Provide users with easy-to-understand visualizations of plant health metrics.*
- *Offer actionable suggestions for optimal plant care based on current health data.*

Through this project, we aim to create a comprehensive platform that empowers home gardeners to make informed decisions about their plants' needs. Our goal is to make PlantPulse the go-to tool for home gardening enthusiasts, enabling them to grow healthy and thriving plants with ease.

In this paper we will present our thought process throughout all the design stages as well as a detailed breakdown into the selection process for all the parts used. We will also provide the schematics, early prototype PCB layouts, and testing we have completed to verify our design.

2 - PROJECT DESCRIPTION

2.1 PROJECT BACKGROUND AND MOTIVATION

Home gardening is increasing in popularity among many Americans, whether it is for at home sustainability or simply as a hobby. With this type of endeavor, however, comes the challenge of monitoring the wellbeing of the plants which are being cared for. Without various tools, it is nearly impossible to keep track of all the necessary measurements required to successfully care for and grow a home garden. The goal of our project, PlantPulse is to alleviate the difficulty in managing a home garden by providing an all-in-one solution to monitoring plant health allowing for an easier time gardening.

One of the major goals of the Senior Design project is to demonstrate all our prior knowledge and gain a more real-world experience by collaborating with a group of potentially different backgrounds to solve or alleviate a real-world problem. When discussing ideas for the project initially, our group was trying to decide on an impactful problem to work towards which would leverage the skills of everyone in the group while also providing a good learning experience. We had several initial ideas, including telemetry systems which would help rocket trajectory, and other ideas involving the growing field of computer vision. Ultimately, we felt that these ideas were very interesting concepts, but did not have the kind of impact we were looking for.

We came across the idea of plant health monitoring as we had known many people who tried to dip their toes into home gardening as an interest but gave up due to little success without a proper monitoring system. Those who did have a monitoring system, still had no real way of determining the condition of the plant's health itself, which still resulted in failure. This gave way to the idea of a device that would monitor the conditions surrounding the plants themselves such as the temperature and transmit this information to the user along with a general measure of the plants overall health to ensure the user knows exactly how their plant is doing at any given time. After discussing this idea, we felt this matched exactly what we were looking for, something impactful, reasonable given a smaller group size of only three individuals and would give us a decent learning experience as we had to learn a lot of new technologies relating to plant health.

2.2 CURRENT TECHNOLOGY

When it comes to gardening, aside from external factors such as the weather and pests, there are several factors that contribute to the health of the plants. These include soil quality, sunlight, soil moisture, which many products in this market currently offer solutions for. The Normalized Difference Vegetation Index (NDVI) which is used to quantify the health and density of vegetation is also used currently to monitor plant health in many commercial settings to assist gardeners with maintaining optimal plant health. Both types of devices are quite prevalent with notable examples coming from companies such as Photon Systems Instruments and TheConnectedShop.

Our goal was to build upon this technology in a way that would make it convenient to utilize both technologies in tandem to provide an optimal plant health monitoring system. This provides general information such as moisture level and utilizes NDVI to provide an overall reading of the plants health which would allow the average consumer to easily determine how healthy their plants are and determine which measurements require attention and by how much. PlantPulse will include a temperature and humidity sensor, a moisture sensor, a UV index sensor, and a near infrared (NIR) camera that will allow the system to determine the NDVI. All these metrics will then be displayed to the user via either web app or mobile app on regular intervals.

2.3 OBJECTIVE AND GOALS

The main goal of the PlantPulse project is to create a plant monitoring device that incorporates basic measurement sensing with more advanced NDVI technology to provide a simple and effective way to determine plant health.

The subgoals and objectives required for this project can be seen below:

Goals:

- The device will provide the user with an overall view of their plant's health.
- The device will provide the user with periodic sensor measurements.
- The user will be able to interact with the device easily and effectively.

Stretch Goals:

- The device will be able to interface with the user via web application and mobile application.
- The device will be able to provide the user with various advanced suggestions based on current plant health metrics.
- Identification system of various plants, insects, and diseases.

Objectives:

- Autonomous humidity sensing of various diagnostic measurements including humidity, soil moisture, temperature, and UV light.
- Autonomous NDVI imaging of plants using a NIR camera to obtain readings at set times (at least three times a day).
- Resistance to inclement weather conditions while maintaining an acceptable level of functionality.
- Monitor power is run on solar power or a rechargeable back-up battery without deliberate effort from the end user.
- Transmit all received sensor data from the database to the end user.
- Relay the NDVI readings to the user in easy-to-understand fashion.

Autonomous humidity sensing will be achieved by a network of IoT sensors to obtain readings on temperature, humidity, soil moisture, and UV light. To complete this task our design will rely on the SHT45 dual temperature and humidity sensor, AS7331 UV light sensor, and PIM520 ground moisture sensor. These sensors will take readings periodically when woken up by a timer on the MCU. These readings will occur every hour for the light sensor and temperature sensor. The ground moisture sensor will run continuously but will be read when the MCU wakes up to read the other sensors. This data will then be transmitted to a web database via Wi-Fi.

Autonomous NDVI imaging will be achieved through the use of a Pi cam attached to a Raspberry Pi Zero 2 W. This system will be woken up three times a day to take the NDVI image. This wakeup routine will be triggered by a timer interrupt handled by the microcontroller unit which will signal each of the sensors to awake. Additionally, this wakeup routine will be determined by the light and temperature values seen by the temperature and ultraviolet light sensors as the plant will need to be in specific conditions to generate a good NDVI reading. The image and the sensor data will then be transferred to a web database via Wi-Fi and processed for user display.

Inclement weather resistance will be obtained by using housings for all the electronics that are of a high enough IPx rating to ensure resistance to water and dust. The PCBs themselves will also be designed with the higher-than-normal ambient temperatures in mind with specific attention to optimizing the thermal performance of the final layout. We will also make sure to spend an extra amount of attention on the actual design of both the camera and sensor packages to ensure that the exposed sensors do not interfere with the weather resistance of the system.

Solar panel will be achieved via the use of a 12V 5W solar panel and a 3000mAh 7.2V Li-ion battery pack. This will provide the power to the main camera package and will run without intervention from the end user. This system can deliver enough power to keep the system running indefinitely and the backup battery is large enough to provide power for multiple days without solar input. The sensor packages will achieve this via the use of high-capacity lithium primary battery cells. These will eventually die but their charge cycle should last for multiple months at a minimum and they will be easily serviceable by the end user.

All readings will be transmitted via a network of Bluetooth and Wi-Fi connections. The Bluetooth connections will allow the camera package, sensor package, and Raspberry Pi Zero 2 W to communicate without obstructing the end user's home Wi-Fi bandwidth and will allow the system to have an overall lower power consumption. We will ensure that the connections to Bluetooth and Wi-fi are optimized to have the system meet our specified battery life. The data shared from these packages will then be gathered at the camera package before being sent to a web database via Wi-Fi. The information stored in the database will then be relayed to the end user in an easy-to-read fashion via a web application.

Once the images are processed by the Raspberry Pi zero 2 and relayed to the database, we will compute the NDVI value on the backend to avoid extra-long processing time on the system itself. This information will then be interpreted and presented to the end user in a way that allows individuals who are unfamiliar with NDVI to understand the rating and determine their specific plant health from the value. The system will also provide recommendations based on the NDVI reading as well as the sensor data which is all collected at the same time. The actual captured images will also be displayed to the user with the NDVI reading so the user can fully understand how the information is gathered and what it means physically for their specific plant setup.

2.4 MARKETING REQUIREMENTS

Marketing Requirements:

Users don't want to spend a ton of money just to monitor their garden, so one of the primary marketing requirements for our system is low cost and expandability. A low barrier to entry is key, to allow the user to start small and build as they recognize the value the system provides, and their garden expands. We plan to achieve this by creating a mesh of interconnected devices, a camera package that acts as the primary hub for multiple plants, and a sensor package that sends localized sensor readings for a specific plant or plants.

The system should be able to function outdoors in a variety of elements and conditions. Most gardens, particularly home gardens, reside outside and thus are exposed to a myriad of challenges for electronic devices. Our devices should be able to reside outdoors for most of their working life. To achieve this we plan on designing our devices with weatherproofing techniques and resilient materials capable of withstanding high UV exposure as well as a wide range of ambient temperature.

If the data, the system provides is not useful there is no reason for a user to purchase our system. The system should provide real insight into individual plant health and suggest actions the user can take to improve their garden. We plan to achieve this with accurate localized data combined with a knowledge base capable of delivering an excellent user experience.

The user is trying to save time and effort in using our system and if the user must charge devices constantly the value created is quickly diminished and frustration sets in. The system should be very low maintenance overall, allowing the technology to fade into the background. This can be achieved with the combination of very efficient microprocessors, energy dense batteries, and solar charge capabilities. These should deliver the experience the user expects.

What use does the system have if it only works with a few plant types, planting methods, or plant sizes? The system should provide the expected results on a wide variety of plants in any configuration. This means the devices should be adaptable, allowing the user intuitive guided placement for optimal performance. A large knowledge base of plant types and their optimal care conditions should be available. The system should also have environmental knowledge of different regions, weather conditions, and potential insect issues. Finally, the system should have knowledge of different planting techniques and the necessary conditions to use in each case.

2.5 SPECIFICATIONS

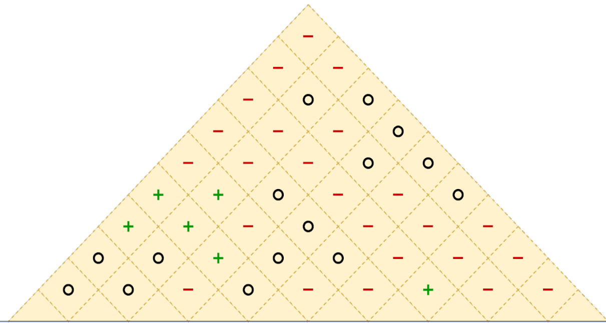
Table 2.6-1. Specifications

Overall Device	
Parameter	Specification
Housing Tilt Function	60 Degrees
Device Max Response Delay	~1 minute
Device Total Max Cost	$100
Operating Temperature	-20-65 °C
Power Supply	
Parameter	Specification

Solar Panel Peak Voltage	5 V
Solar Panel Peak Power	0.6 W
Battery Life	2 Days
Sensors and Camera	
Parameter	Specification
Light Sensor Accuracy	90%
UV Light Sensor Input Range	260-360 nm
Camera Required NIR Sensitivity	830-870 nm
Temperature Sensor Input Range	-45-125 °C
Humidity Sensor Accuracy	90%
Humidity Sensor Input Range	0-100% RH

2.6 HOUSE OF QUALITY

Our house of quality diagram shows our key focus points for our design and where we can optimize and what trade-offs we may need to make to meet our customer and design requirements. The focus points include affordability, ease of use, accuracy, compatibility, reliability, ease of maintenance, upgradability, and communication range. The major performance targets of the product include response time, housing tiltability, sensor accuracy, light sensor range, camera resolution, battery life, size, cost, operating temperature, and water resistance. This diagram with the product targets and the corresponding directions of desirability can be seen below in figure 2.6-1.



#	Customer Requirements \ Functional Requirements	Direction of Desirability	Performance	Housing Tiltability	Sensor Accuracy	Light Sensor Ranges	Camera Resolution	Battery Life	Size	Cost	Operating Temperature Range	Weather Resistance
			1	2	3	4	5	6	7	8	9	10
	Direction of Desirability		▲	▲	▲	▲	▲	▲	▼	▼	▲	▲
	Affordability	▲	↓↓	↓	↓↓	↓	↓↓	↓	↑	↑↑	↓↓	↓↓
	Ease of Use	▲	↑	↑	○	○	↑	↑↑	↑	○	↑	↑
	Accuracy	▲	↑	○	↑↑	↑	↑↑	↓	○	↓↓	↓	↓
	Compatibility	▲	○	↑	↑	↑	↑	↓	○	↓	↑	↑
	Reliability	▲	○	○	↑↑	↑	↑	↑↑	○	↓	↑	↑↑
	Ease of Maintenance	▲	○	○	○	○	○	↑	↑	↓↓	↑	↑↑
	Upgradability	▲	↑	○	○	↑	↑	↑	○	○	○	○
	Communication Range	▲	○	○	○	○	↓	↓↓	○	↓↓	↓	○
	Product Targets		Response < 1min	Min Tilt 60°	Accuracy 90%	UV Light 260-360nm Camera NIR Range 830-870nm	2MP	2 Days without Solar	Small enough for most household planters	Cost < $100	Operating Temp. -20-65 Degrees Sensor Temp. Range -45-125 Degrees Celsius	IP rating IP54

Correlation	
+	Positive
−	Negative
O	No correlation
Relationship	
↑↑	Strongly Positive
↑	Weakly Positive
○	No Relation
↓	Weakly Negative
↓↓	Strongly Negative
Direction of Desirability	
▲	High
▼	Low

Figure 2.6-1. House of Quality

2.7 DEVICE ILLUSTRATION

Below (figure 2.7-1) is a mockup design of the resulting product. The device enclosures are designed for a long-term outdoor environment. Design considerations include extended exposure to sunlight (UV radiation and high temperatures), exposure to weather (primarily rain, wind), and a range of ambient temperatures. We are using 3D printing technology to build the enclosures; we have a 3D printer ready to use. Materials we plan to use are ASA for the structure and TPU for waterproofing gaskets. The vent for the humidity sensor will have a fine mesh screen to prevent water ingress to the level required by IP5 standard. The camera hole will be sealed with a watertight bonding agent. The adjustable hinge will also allow the wires for the solar cell and UV light sensor to pass through it with physical safety stops to prevent potential damage as the panel is adjusted. We are also considering making the camera package stake removable in favor of varied attachable mounting options.

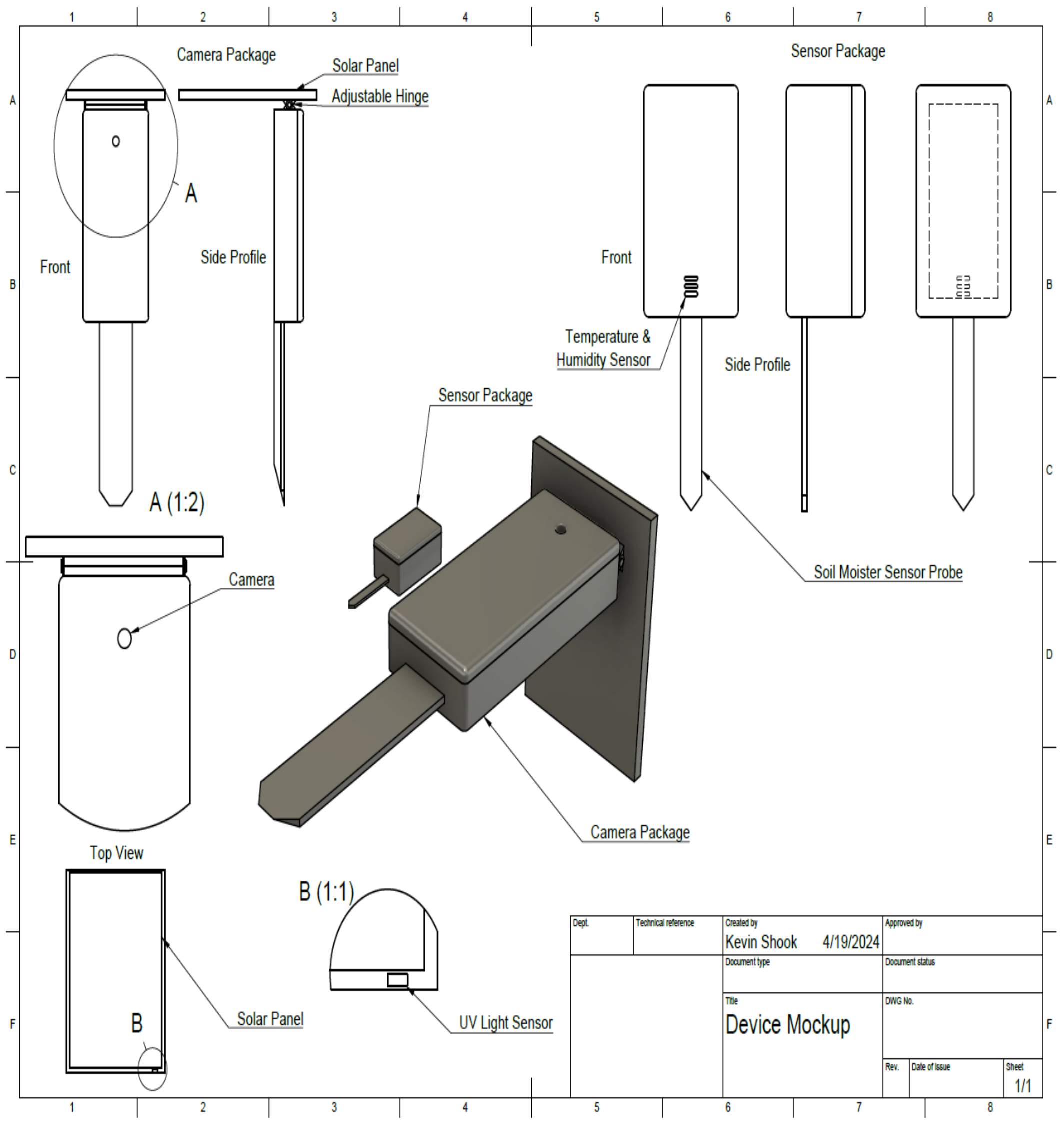


Figure 2.7-1. Camera Device Mockup

2.8 HARDWARE BLOCK DIAGRAM

The block diagram for both our camera package and sensor package are provided below in Figure 2.8-1. The figure also shows the workload breakdown in color coded modules. Each team member is responsible for their designated component of the diagram but not necessarily the only one working on it. Further breakdown of each module can be found further in the software flow section.

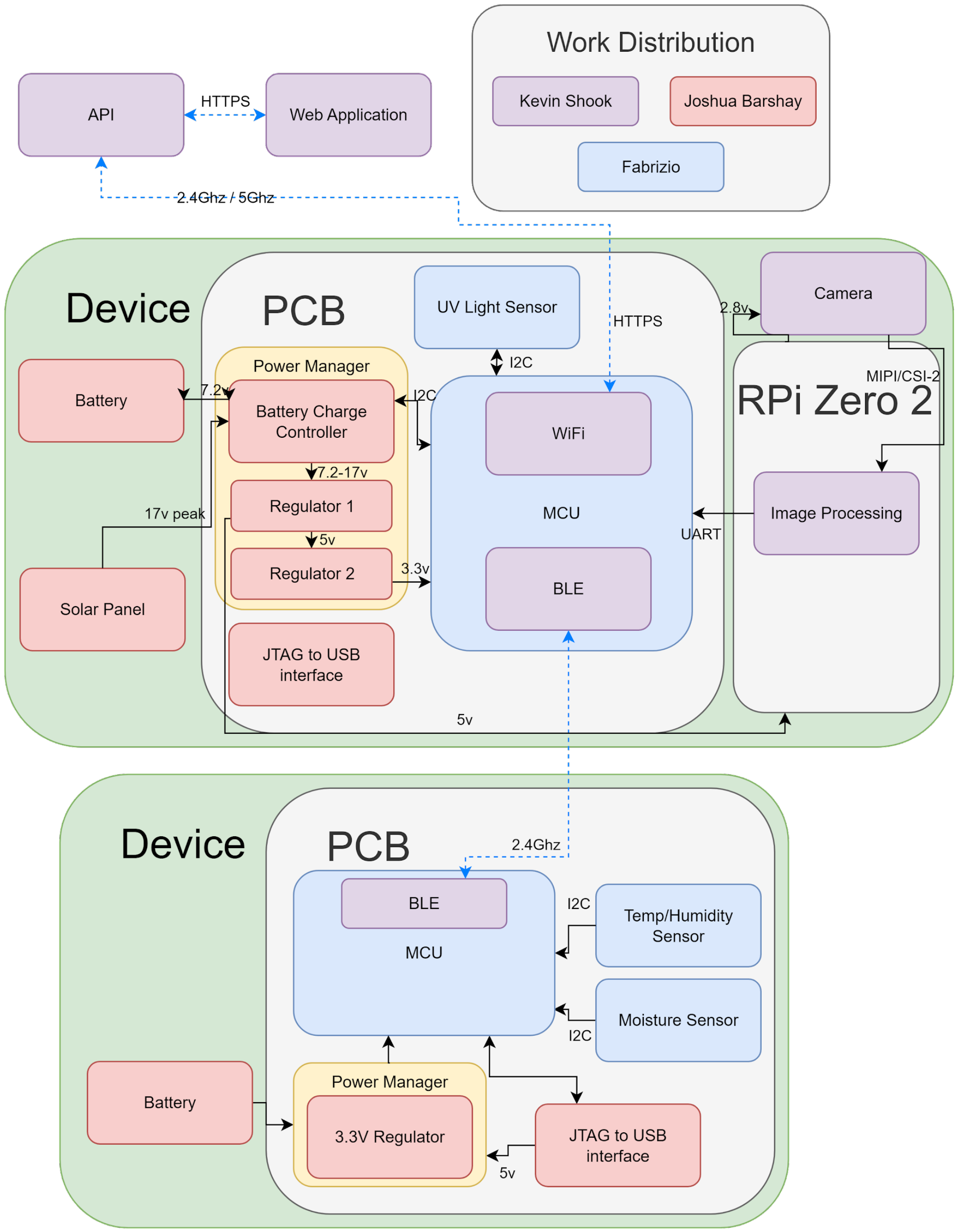


Figure 2.8-1. Device Hardware Diagram

2.9 SOFTWARE FLOW DIAGRAMS

The Figure 2.9-1. Illustrates the software flow and work distribution for the camera package. Figure 2.9-2 shows the software block diagram detailing how the device interacts with our web services and the work distribution involved with building the backend services that provide much of the functionality of the system.

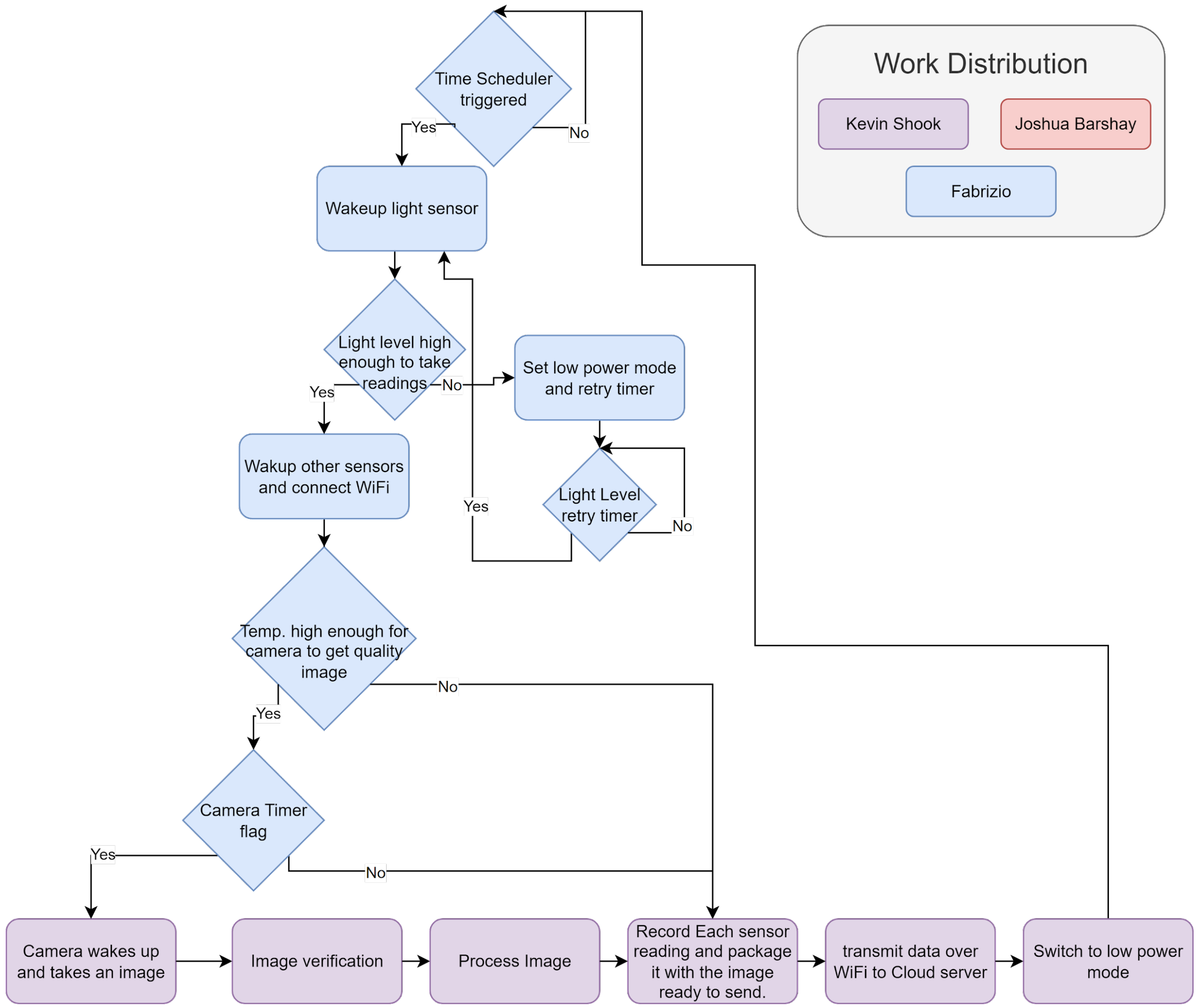


Figure 2.9-1. Device Software Diagram

The web services described below in Figure 2.9-2 provide much of the heavy lifting of the system including providing the wealth of knowledge needed to properly care for a wide variety of plants. Some of the services exposed by the API include plant classification, disease identification and remedy recommendation, insights into insect issues, as well as daily maintenance advice. The user interface is entirely driven by these services either via the web application or a mobile app identified in the diagram as the client.

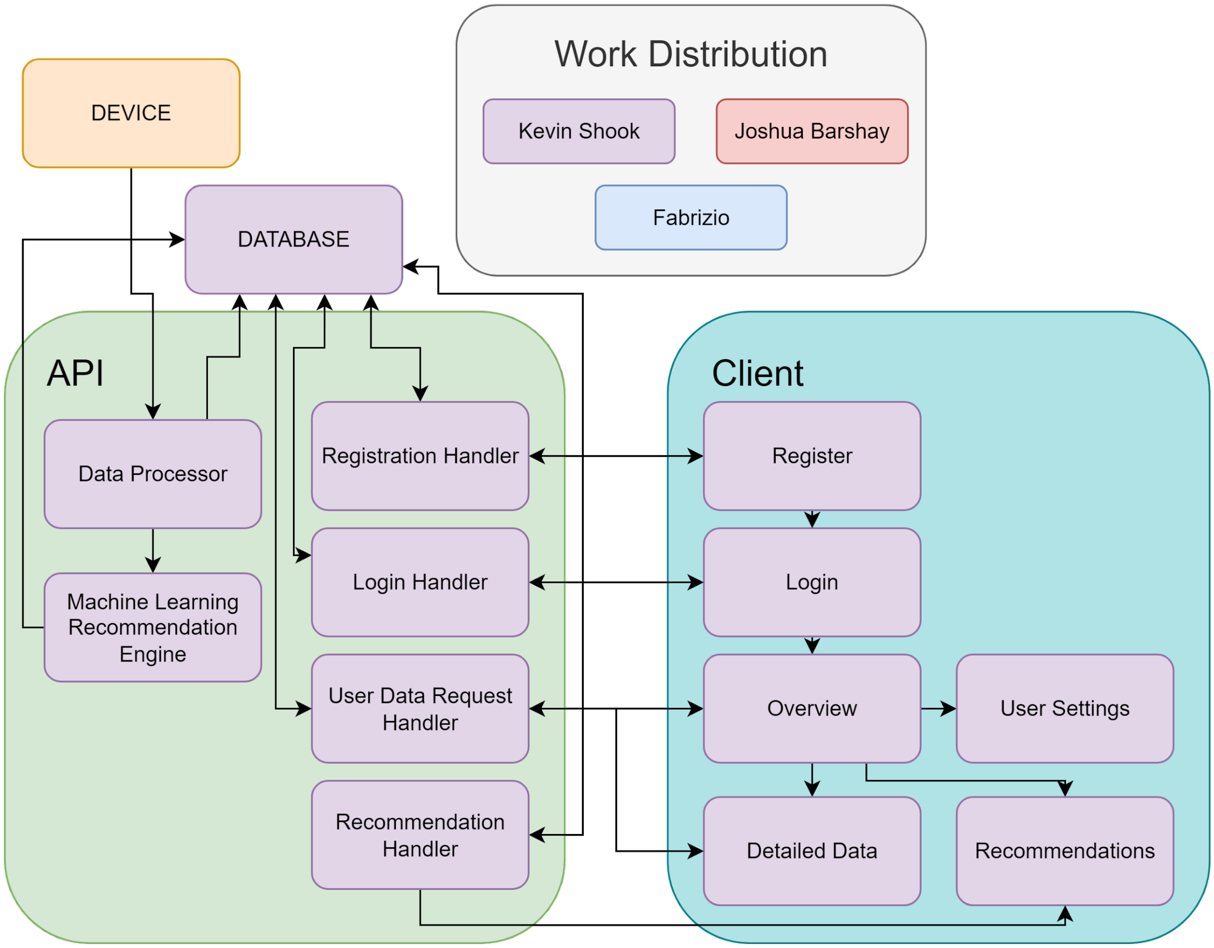


Figure 2.9-2. Web services and client application diagram

3 -RESEARCH AND INVESTIGATION

3.1 HARDWARE

3.1.1 Camera Technologies

Leveraging cutting-edge camera technologies facilitates the early detection of plant stressors, offering farmers proactive management strategies. As we transition these technologies into the consumer space, we embark on a comparative analysis of various camera technologies employed in commercial settings. Our focus lies on discerning the efficacy of different camera technologies for plant health monitoring. Employing the Normalized Difference Vegetation Index (NDVI) methodology, we evaluate plant health based on infrared reflectance. Increased infrared reflectance indicates heightened photosynthetic activity. We will be utilizing a NDVI technique for use with RGB cameras that are sensitive to near infrared (NIR). The algorithm approximates NDVI by the equation (1) where NIR is a pixel in the NIR wavelength and VIS is a pixel in the visible wavelengths. This combines the information available in the visible and NIR bands into a single and representative value between -1 and +1. [12]

$$NDVI = \frac{NIR - VIS}{NIR + VIS} \quad (1)$$

To effectively measure this, we require camera systems capable of detecting infrared wavelengths reflected by plants and correlating the measurements with NDVI values. Our exploration encompasses three distinct technologies: No IR filter cameras, Near IR + Red bandpass Filter cameras, and Infrared imaging systems. Through careful examination and comparison of their properties, we aim to align these technologies with the objectives of our project.

No IR Filter:

Cameras devoid of an IR filter capture light across the visible spectrum, including infrared wavelengths. These cameras allow the detection of subtle variations in plant health, leveraging the reflective properties of different wavelengths absorbed and emitted by plants.

Near-IR + Red Pass Filter:

Cameras equipped with a near-IR + Red pass filter selectively capture near-infrared and red light. This filtering mechanism enhances the differentiation between healthy and stressed vegetation, leveraging the chlorophyll absorption bands and the reflectance properties of plant leaves.

Infrared Imaging Systems:

Infrared imaging systems utilize sophisticated sensors to capture thermal radiation emitted by objects, including plants. These systems provide detailed thermal maps of plant surfaces, enabling the detection of temperature differentials associated with stressors such as water deficiency and disease.

In the table below (Table 3.1.1-1) we weigh the advantages and drawbacks of each camera system, providing valuable insights into their performance levels and inherent trade-offs.

Table 3.1.1-1. Camera Technology Comparison

Measure	No IR Filter	Near-IR + Red Pass Filter	Infrared Imaging Systems
Spectral Range	Visible + IR	Near-IR + Red	Infrared
Spatial Resolution	High	Moderate	High

Cost	Low	Moderate	High
Sensitivity	Moderate	High	High
Complexity	Low	Moderate	High
Applications	General Purpose	Agriculture	Precision Agriculture
Processing Requirement	Minimal	Moderate	High

Spectral Range: Cameras with no IR filter offer broad spectral coverage, while near-IR + Red pass filters target specific wavelengths conducive to vegetation analysis. Infrared imaging systems focus solely on thermal radiation, providing unique insights into plant health.

Spatial Resolution: No IR filter cameras typically offer high spatial resolution, crucial for detailed plant monitoring. However, near-IR + Red pass filter cameras strike a balance between resolution and spectral specificity. Infrared imaging systems also offer high resolution but are primarily thermal-focused.

Cost: No IR filter cameras are cost-effective, making them accessible for general monitoring purposes. Near-IR + Red pass filter cameras incur moderate costs, catering specifically to agriculture applications. Infrared imaging systems represent a significant investment due to their advanced technology.

Sensitivity: Near-IR + Red pass filter cameras and infrared imaging systems exhibit high sensitivity to subtle changes in plant health. Cameras without an IR filter may lack sensitivity to specific stress indicators.

Complexity: No IR filter cameras are straightforward to operate, requiring minimal processing. Near-IR + Red pass filter cameras and infrared imaging systems are more complex, necessitating specialized knowledge and software for data analysis.

Applications: While cameras without an IR filter serve general monitoring purposes, near-IR + Red pass filter cameras find extensive use in agriculture. Infrared imaging systems excel in precision agriculture applications, offering detailed thermal insights.

Processing Requirement: Cameras without an IR filter require minimal processing, suitable for real-time monitoring. Near-IR + Red pass filter cameras and infrared imaging systems demand more intensive data processing and analysis, delaying insights.

Due to the low target cost requirements we decided to go with the no-IR filter option. This choice not only aligns with our budget constraints but also enhances color capture in the image, thereby facilitating the identification process of plant species and potential plant diseases or insect issues.

3.1.2 Camera Parts Comparison

Based on our review of the different camera technologies available capable of detecting the wavelengths of light necessary for NDVI, we have narrowed our search for sensors down to three different camera sensors. In figure detailed breakdown and comparison of each sensor.

Table 3.1.2-1. Camera Parts Comparison

Measure	OV2640 no-IR	OV5640 no-IR	Pi Cam no-IR
Sensor Model	OV2640	OV5640	Raspberry Pi Camera
Maximum Resolution	1600x1200	2592x1944	2592x1944
Pixel Size	2.2µm	1.4µm	1.4µm
Field of View (FOV)	65°	60°	53.5°
Frame Rate	Up to 60 fps	Up to 15 fps	Up to 30 fps
Interface	DVP or SPI	DVP or MIPI CSI-2	MIPI CSI-2
IR Filter	No	No	No
Lens Compatibility	M12 Mount	M12 Mount	Fixed Lens
Low Light Performance	Moderate	Moderate	Moderate
Cost	Low	Moderate	Low to Moderate
Availability	Low-No Availability	Low-No Availability	Widely Available

Applications	General Monitoring	Mobile Devices	Raspberry Pi Projects

Sensor Model:

OV2640: Manufactured by Omni Vision, the OV2640 is a 2-megapixel CMOS image sensor widely used in camera modules for various applications.

OV5640: Also produced by Omni Vision, the OV5640 is a 5-megapixel CMOS image sensor known for its higher resolution and image quality compared to the OV2640.

Pi Cam: The Pi Camera is a camera module designed specifically for Raspberry Pi boards, featuring a 5-megapixel sensor, and optimized for use with Raspberry Pi projects.

Maximum Resolution:

OV2640: Offers a maximum resolution of 1600x1200 pixels.

OV5640: Provides a higher maximum resolution of 2592x1944 pixels.

Pi Cam: Shares the same maximum resolution as the OV5640 at 2592x1944 pixels.

Pixel Size:

OV2640: Has larger 2.2µm pixels, which may offer better sensitivity in low-light conditions.

OV5640 and Pi Cam: Both feature smaller 1.4µm pixels, which can lead to higher resolution but may result in reduced low-light performance.

Field of View (FOV):

OV2640: Offers a wider field of view of 65°.

OV5640: Provides a slightly narrower FOV of 60°.

Pi Cam: Has the narrowest FOV of 53.5°.

Frame Rate:

OV2640: Supports frame rates of up to 60 frames per second (fps).

OV5640: Offers lower frame rates of up to 15 fps.

Pi Cam: Capable of frame rates of up to 30 fps.

Interface:

OV2640 and OV5640: Support DVP (Digital Video Port) interface or MIPI CSI-2 (Camera Serial Interface) for communication with microcontrollers or processors.

Pi Cam: Utilizes MIPI CSI-2 interface specifically designed for Raspberry Pi boards.

IR Filter:

All cameras: Do not come with an IR filter, making them suitable for infrared imaging applications.

Lens Compatibility:

OV2640 and OV5640: Support M12 mount lenses, providing flexibility for different focal lengths and applications.

Pi Cam: Comes with a fixed lens, limiting options for lens customization.

Low Light Performance:

All cameras: Offer moderate low-light performance, with the OV2640 potentially having a slight advantage due to its larger pixel size.

Cost:

OV2640: Generally, has a lower cost compared to the OV5640 and Pi Cam.

OV5640: Priced moderately higher than the OV2640.

Pi Cam: Available at a similar price range as the OV5640.

Availability:

OV2640: Widely available from various suppliers and distributors.

OV5640: Has moderate availability, depending on supplier stock and demand.

Pi Cam: Widely available from Raspberry Pi distributors and retailers.

Applications:

OV2640: Suitable for general monitoring applications requiring moderate resolution and frame rate.

OV5640: Ideal for mobile devices and applications requiring higher resolution imaging.

Pi Cam: Specifically designed for Raspberry Pi projects, offering compatibility and ease of integration with Raspberry Pi boards.

Based on the comparison, the selection of the best camera for the project depends on factors such as resolution, frame rate, cost, and compatibility with the target platform. If high resolution and frame rate are critical, the OV5640 or Pi Cam may be preferred. However, if cost-effectiveness and wider FOV are more important, the OV2640 could be the better choice. Availability is also crucial for our project needs as we do not have the ability to work with manufacturers for custom camera parameters. We chose to go with the OV2640 due to it working with our low-cost microprocessor, the cost of the sensor being within our target budget requirements, and its market availability. However, after attempting to get the no-IR version of the camera we have found that the no-IR version is no longer sold as of last year. We were only left with the Pi Cam no-IR version and are using a raspberry pi zero 2 as an interface between our MCU and the Pi Cam due to camera interface incompatibility.

3.1.3 Ultraviolet Light Sensor Technology Comparison:

Photoresistors:

Photo resistors are a type of light sensor which utilizes a light sensitive material which changes resistance based on the lighting, increasing resistance as luminance decreases and decreasing resistance as luminance increases, typically exponentially. These sensors can measure visible light as well as light in different wavelengths including ultraviolet light depending on the material of the photoresistor, which impacts how sensitive the photoresistor is to light of specific wavelengths. Modern photoresistors are made of material like lead sulfide, lead selenide, indium antimonide, and most commonly cadmium sulfide and cadmium selenide. These types of light sensors are the simplest and the cheapest to manufacture, however they do suffer from what is called resistance recovery, which means that there is a delay between changes in illumination and the resistance of the photoresistor. This property makes photoresistors poor choices where light is fluctuating rather quickly as there can be up to a 1ms delay when going from total darkness to light and up to 10ms delay when going from total light to dark. [22][23][24]

Photodiodes:

Photodiodes are a semiconductor with a PN junction which converts photonic energy to electrical energy. Like photoresistors, they are made of materials like Silicon, Germanium, and Indium Gallium Arsenide, with the sensitivity to wavelengths being determined by the specific material used to manufacture the photodiode. Unlike photoresistors, photodiodes work off the photons hitting the atoms of the diode releasing electrons which creates an electron hole pair. The photons absorbed in the depletion region of the photodiode will move to opposite ends due to the electric field generated, creating the electrical current. Several other types exist such as PIN photodiodes, Avalanche photodiodes, and Schottky photodiodes, all of which provide different uses. Photodiodes in general are much faster than photoresistors and respond to changes in light much better as they do not suffer from resistance recovery. [22][23][24]

Phototransistors:

Phototransistors are also semiconductors which are used to convert photonic energy to electrical energy. When light hits their junction, currents flow in the opposite direction which are proportional to the luminance of the light. This happens because phototransistors have the base terminal of a standard transistor exposed which acts as sending current through the base when photons hit this terminal activating the transistor. As these current increases, it is concentrated an is converted into a voltage as well, which means phototransistors can produce both current and voltage unlike photodiodes and photoresistors. This difference from a standard transistor is due to the material, which is used to create a phototransistor, which is a bipolar semiconductor. Almost all phototransistors are responsive across all the wavelengths of visible light including ultraviolet and infrared. Phototransistors come in a few different configurations, with the most popular two being BJT and FET Phototransistors for different requirements. In general phototransistors are more sensitive than photodiodes but have a much narrower response range compared to the photodiode. Phototransistors also allow for a much higher current to pass through them and have a slower output response making them less suitable for low power applications compared to photodiodes. [22][23][24]

Other types of sensors:

Several other types of light sensors are available as well, including phototubes and photovoltaic cell technology. Photovoltaic cells are useful for detecting ultraviolet light and are used extensively in solar panels which are discussed further in the technology comparison for the solar panel component of our system.

Table 3.1.3-1. UV Sensor Technology Comparison

Sensor Type	Typical Response time	Typical Accuracy	Wavelength Range
Photoresistor:	Slow	Low	Moderate
Photodiode:	Very Fast	High	Highest
Phototransistor:	Fast	Very High	High

Considering the specifications for the system we needed a rather accurate light sensor with an input range for ultraviolet light. Considering the photodiode technology typically has the highest wavelength range and response time with good accuracy, we selected this type of sensor for our application.

3.1.4 Ultraviolet Light Sensor Part Comparison:

Table 3.1.4-1. UV Sensor Parts Comparison

Specifications	AS7331	LTR-390UV-01	VEML6075

Bandwidth Accuracy	N/A	N/A	±10nm
Bandwidth Range	260 nm – 360 nm	270 nm – 470 nm	300 nm - 365 nm UVA 300 nm - 330 nm UVB
Sensitivity	421 counts/(µW/cm²)	2300 Counts/UVI	UVA: 0.93 counts/(µW/cm²) UVB: 2.1 counts/(µW/cm²)
Voltage range	2.7V – 3.6V Typical: 3.3V	1.7V – 3.6V	1.7V – 3.6V
Average Current	970µA – 2mA Typical: 1.5mA	100µA – 110µA	480µA
Interfaces	I2C	I2C	I2C
Operating Temperature	-40ºC ~ 105ºC	-40ºC ~ 85ºC	-40ºC ~ 85ºC
Dimensions	3.6 mm x 2.6 mm x 1.09 mm	2.0 mm x 2.0 mm x 2.0 mm	2.0 mm x 1.25 mm x 1.0 mm
Weight	N/A	N/A	N/A
Price	$10.86	$3.11	$5.95

As discussed in section 3.1.3 on ultraviolet light sensor technologies, our choice ended up being a photodiode based ultraviolet light sensor. The specific sensor we selected was the ams OSRAM AS7311 3-channel UV-A/B/C Spectral Sensor Integrated Circuit. This sensor offers three channels to measure the different ultraviolet light wavelengths being ultraviolet A, ultraviolet B, and ultraviolet C utilizing photodiodes to convert the analog signal to digital ones in either a continuous mode or trigger mode. This sensor also provides a variable responsivity range and sensitivity using gain conversion and the integrated circuits internal clock. As a result of this feature the overall irradiance response has a range up to 3.4E10 which is an extremely high value for response and a high sensitivity value of up to 421 counts per µW/cm², which is also adjustable. The sensor is well within our specified temperature range as well, functioning from -40ºC to 105ºC and with this also has temperature compensation for the measurement results ensuring that the change in temperature doesn't impact the stability or accuracy of the sensor's readings. [25]

The other options on the market were very limited, as many of the sensors did not actually read the specific wavelength we needed for the ultraviolet light sensor, or they were ambient light sensors which did not measure ultraviolet radiation at all. We were able to find two other ultraviolet light sensors which met our requirements which were the LTR-390UV-01 and the VEML6075.

The LTR-390UV-01 sensor (will be abbreviated as LTR-390 for this section) is also a photodiode based integrated circuit, which offers a low power consumption mode and a programmable interrupt which improves efficiency. This sensor has a relatively good sensitivity with 2300 counts per µW/cm² in the 270 nm to 470 nm wavelength range. This sensor also provides a good dynamic range of 1.8E7 with a high spectral response as well. Overall, this is a good sensor for our applications meeting the communication requirement using I2C and the temperature range being between -40ºC and 85ºC. [26]

The VEML6075 sensor is also a photodiode based integrated circuit, which measures both Ultraviolet A and ultraviolet B in the wavelength range of 300 nm to 360 nm. The VEML6075 has a temperature compensation capability keeping the readings stable across temperature changes as well and good sensitivity having a value of 0.93 counts per µW/cm² for the ultraviolet A range and 2.1 counts per µW/cm² for the ultraviolet B range. This sensor also meets the requirements for communication utilizing I2C and has the same temperature range as the LTR-390UV-01 sensor mentioned previously. This sensor would have been a good choice but unfortunately has been discontinued by the manufacturer and is no longer available. [27]

Overall considering the three sensor options, the ams OSRAM AS7311 3-channel Ultraviolet light sensor was our choice as it offered the highest accuracy and flexibility with its high dynamic range and sensitivity for the three different ultraviolet light wavelengths, which was something other sensors lacked. This sensor did have the highest price out of our options at a price point of $10.86 per chip. With this sensor having an integral function of the monitoring system, enabling the camera when the light level is suitable to capture for the NDVI measurement, we decided this would be the best option despite the price.

3.1.5 Temperature Sensor Technology Comparison

Thermocouples:

Thermocouples are the most common type of temperature sensor, which like the other types we will discuss in this section have applications in automotive, consumer, and industrial areas. These types of sensors work by having two dissimilar metal wires joined together which generate a voltage difference between the two wires. This can be used to calculate temperature with the most common of these materials being Nickel Chromium and Nickel Aluminum. This set up results in a temperature sensor which is self-powered, has a quick response time, and a wide temperature range which is modifiable depending on the materials chosen. Some important things to keep in mind when using thermocouples is that the output voltage, they produce is very small, requiring a suitable amplification for accurate measurements. With this the nature of the sensor also means it can potentially interact with the copper traces transmitting the signal, generating another voltage which needs to be accounted for to ensure the accuracy is not compromised. [28][29][30][31]

Resistance Temperature Detectors (RTD):

Resistance temperature detectors (RTD for short) is a type of resistor which works on the principle that an object's resistance changes with temperature, having defined resistance to temperature characteristics to monitor the changes. The most common material used is platinum as they have linear responses to temperature changes along with pretty good accuracy and stability for a wide temperature range. Due to the nature of these sensors, they do suffer from higher thermal mass which gives them a slower response time to temperature changes and require an external excitation current. [28][29][30][31]

Thermistors:

Thermistors are very similar to RTD’s described above, but instead of platinum or other metals they tend to be made of materials such as ceramics or polymers. This makes them usually cheaper than standard RTDs but also lowers their overall accuracy as well. Unlike RTDs, thermistors do not have a linear temperature to resistance relationship with the most common configuration, Negative Temperature Coefficient (NTC), where resistance decreases as temperature increases. This results in most thermistors requiring a correction to properly interpret the changes in temperature, which is often accomplished by creating a voltage divider circuit with the thermistor and a fixed value resistor where the divider voltage is passed through an analog to digital converter. [28][29][30][31]

Semiconductor Based Sensors:

Semiconductor based sensors are integrated circuits which measure temperature. They utilize the voltage changes across the PN junction of forward bias diodes, which are sensitive to temperature changes. Typically, they consist of two identical transistors at different collector currents, with the difference between the emitter-base voltages denoting an absolute temperature. These types of sensors generally have a more limited temperature operating range and lower accuracy, which results in them needing some form of calibration to a specific temperature range. Modern sensors of this type do also employ signal processing and additional calibrations to yield a higher accuracy across a greater range if needed and can be programmed to have an alert should they run outside of these constraints. [28][29][30][31]

Table 3.1.5-1. Temperature Sensor Technology Comparison

Sensor Type	Average Cost	Response time	Typical accuracy	Typical Range
Thermocouples:	Low	0.1s – 10s	Δ0.5ºC ~ Δ5ºC	-200ºC ~ 2000ºC

RTD:	High	1s - 50s	Δ0.1ºC ~ Δ1ºC	-200ºC ~ 600ºC
Thermistor:	Low	0.12s – 10s	Δ0.05ºC ~ Δ1.5ºC	-100ºC ~ 325ºC
Semiconductor:	Low	5s – 60s	Δ1ºC ~ Δ5ºC	-70ºC ~ 150ºC

The necessary specifications of our system require an operating range of -20ºC to 65ºC with 90% accuracy, device response delay of at max one minute, and a relatively small size. Given these specifications, from the temperature sensor options available the semiconductor-based sensor was the best choice providing at the worst end our minimum specification requirements with very small package sizes available. With recent research in applying CMOS technology to temperature sensors, we were able to find a temperature sensor which significantly outperforms the average semiconductor sensor in accuracy and response time which will be discussed further in the parts comparison section.

3.1.6 Temperature Sensor Parts Comparison:

Table 3.1.6-1. Temperature Sensor Parts Comparison

Specifications	SHT45	SHT41	SHT40
Accuracy-Spec	Avg: ΔRH = ±1.0 %RH, ΔT = ±0.1 °C Worst: ΔRH = ±2.0 %RH, ΔT = ±0.1 °C	Avg: ΔRH = ±1.8 %RH, ΔT = ±0.2 °C Worst: ΔRH = ±3.0 %RH, ΔT = ±0.2 °C	Avg: ΔRH = ±1.8 %RH, ΔT = ±0.2 °C Worst: ΔRH = ±6.0 %RH, ΔT = ±0.2 °C
Voltage Range	1.08V – 3.6V	1.08V – 3.6V	1.08V – 3.6V
Avg current	0.4μA	0.4μA	0.4μA

Interfaces	I2C	I2C	I2C
Operating Range	0 – 100 %RH, -40 ~ 125ºC	0 – 100 %RH, -40 ~ 125ºC	0 – 100 %RH, -40 ~ 125ºC
Dimensions	1.5mm x 1.5mm x 0.5mm	1.5mm x 1.5mm x 0.5mm	1.5mm x 1.5mm x 0.5mm
Weight	N/A	N/A	N/A
Price	\$3.47 - \$7.81 per chip	\$1.72 - \$3.12 per chip	\$1.68 – \$3.28 per chip

For the temperature sensor, as mentioned previously we decided to go with a semiconductor-based temperature sensor for its low cost and relatively good performance. When researching different components for our system we came across the Sensiron SHT4x series of both temperature and humidity sensors combined in one small package utilizing true I2C communication with a Vin, SDA, and SCL on the chip. This sensor family also meets our requirements as far as operating range goes with the range of temperature being -40ºC through 125ºC, giving us more than desired of the -40ºC through 60ºC range specified for the system. [4]

When looking over the product specifications we noticed there were three main options for this family of sensors, the SHT40, SHT41, and SHT45 all of which had no differences as far as current draw, voltage range, size, and operating range. The main difference between the sensors in this family is their overall and worst-case accuracy. The SHT40 is the worst performing and the oldest available sensor with a worst-case temperature accuracy of ±0.2 °C off the expected value which already is well within our parameters of 90% accuracy and has an average response time of approximately two seconds, which is also well within out 60 second latency requirement for readings. When comparing the SHT40 to the SHT41 and SHT45, the temperature accuracy does not improve too much as we go up from the SHT40 to the SHT45 with just a difference of ±0.1 °C in both the average and worst case, but the accuracy significantly increases for the humidity readings which will be discussed further in the next section. [4]

With this information from the specifications in mind, we went ahead and opted to go for the SHT45 relative temperature and humidity sensor as it surpassed all our required specifications for response time, communication protocol, and operating range. Choosing the SHT45 over the SHT41 or SHT40 sensor did come with a price increase as well compared to the base sensor, however we decided that this price difference well worth the increased accuracy in the humidity sensor portion of the sensor as we determined that our monitoring system should be as accurate as possible given the cost of the product.

3.1.7 Humidity Sensor Technology Comparison:

Resistive Humidity Sensor:

Resistive humidity sensors work by utilizing the principle that non-metallic conducts typically have resistance which is dependent on the water content of the material. They are typically made of materials with lower overall resistivity such as ceramic or organic polymer materials which then have more significant changes in resistance with changes in humidity. This material is then placed on top of two electrodes which are configured in an interdigitated pattern which increases contact area. This allows the water to be absorbed which changes the resistivity between the two electrodes which can be measured by a connect circuit configuration. These types of humidity sensors offer a small size and low cost while requiring no calibration to begin measurements. The Distance between the sensor itself and the circuit which measures the humidity can also be rather large compared to some options discussed later, which makes them suitable in remote applications. Due to the nature of these sensors, they do not provide a linear relationship between humidity and resistance, as the resistance and humidity are inversely proportional. They also suffer from sensitivity to chemical vapors and contaminants interfering with the readings, as well as being susceptible to shifting readings if in the presence of water-soluble materials. [17][18][19][20][21]

Capacitive Humidity Sensor:

Capacitive humidity sensors are the most common type of humidity sensor on the market. They work by utilizing a capacitor with a dielectric made of some polymer, which absorbs or releases water depending on the humidity around the sensor. The change in the dielectric constant due to this water absorption impacts the capacitance of the capacitor and is measured using an external circuit. These types of humidity sensors offer a near linear output voltage which allows for easier measurements and are extremely stable over long use periods with a very wide range of RH levels as they can tolerate a higher humidity level. These sensors do require calibration and can be impacted by the distance of the sensor itself compared to the circuit, which is measuring the capacitance of the capacitor, but typically these drawbacks outweigh the advantages of using these types of humidity sensors. [17][18][19][20][21]

Thermal Conductivity Humidity Sensor:

The thermal conductivity humidity sensor, unlike the previous two types discussed, are absolute humidity sensors instead of relative humidity sensors. This means that instead of measuring the actual water content in the air relative to the total amount it can hold at the given temperature, it measures the moisture in the air regardless of the temperature. Thermal conductivity sensors utilize two thermistors with negative temperature coefficients in a bridge circuit, with one of them sealed in a chamber filled with nitrogen, and the other is exposed to the outside via venting holes. When the thermistors receive power, the difference between their resistances is directly related to the absolute humidity. These sensors are generally more suitable for higher temperature applications as they measure absolute humidity instead of relative humidity and provide a higher resolution compared to both resistive and capacitive based humidity sensors. These sensors are however sensitive to other gasses other than nitrogen which have thermal properties and reduce their effectiveness. Despite this, these sensors are very durable and used in more industrial applications compared to the other two. [17][18][19][20][21]

Less Common Humidity sensors:

Nano bricks Humidity Sensors:

These types of sensors utilize Mesoporous tungsten oxide nano bricks as they are highly sensitive to ammonia gasses, organic compounds, and humid environments at room temperature. [17]

Magnetoelastic Humidity Sensor:

These types of sensors utilize uncrystallized alloys made up of materials such as iron, nickel and cobalt which are sensitive to changes in magnetic fields resulting in shape changes. These properties can then be utilized to measure humidity. [17]

Optical Fiber Humidity sensors:

These types of sensors are relatively new and bring with them the benefits of optical fiber type sensors being lightweight, great in harsh environments, and generally good chemical stability. [17]

Table 3.1.7-1. Humidity Sensor Technology Comparison

Sensor Type	Average Cost	Typical accuracy	Typical Range
Resistive:	Low	Δ2%RH	-40ºC ~ 100ºC
Capacitive:	High	2%RH	Up to 200ºC
Thermal:	Moderate	Δ5%RH @ 40ºC Δ0.5%RH @ 100ºC	Up to 300ºC

Considering that the capacitive type of humidity sensor accounts for nearly 75% of all humidity sensors currently on the market and they also meet the requirements of our specifications, the choice we went with was the capacitive type of humidity sensor. The sensor we selected is further discussed in the parts comparison section and vastly outperformed the averages for these types of sensors as well.

<u>3.1.8 Humidity Sensor Parts Comparison:</u>

For the humidity sensor we will be referring to the table presented in the section discussing the temperature parts comparison (Table 3.1.6-1) and information directly from the SHT4x datasheet provided by Sensiron. As we discussed in the section above, most humidity sensors on the market are currently capacitive type sensors, including the SHT4x sensor family which we decided to use for this project for its high accuracy in both temperature and humidity, as well as surpassing all of our minimum requirements for the system. [4]

This family of sensors integrates the temperature sensor component into the same unit as the humidity sensor utilizing both capacitive humidity sensor technology and having two bipolar transistors to form the semiconductor-based temperature sensor. By integrating both components into the same unit it allows for the humidity sensor to have more precision in determining the dew point, avoiding the error associated with separated measurement signals., resulting in an overall more accurate sensor. Having both components near each other also allows analog to digital conversion units to be integrated nearby as well, resulting in more stability in the readings and repeatability as the signals are not sensitive to interference. This sensor family also provides excellent resistance to more extreme conditions including an optional PTFE membrane that can allow for extra filtration of particles and pollutants which can reduce the response time of the sensor. [4]

All the options in this sensor family provide excellent accuracy with the lowest performing sensor being the SHT40 which has a typical accuracy of ±1.8 %RH between 30% and 80% relative humidity in the 0ºC to 80ºC range. The SHT40 does get less accurate in the more extreme ranges with an accuracy of ±2.0 %RH between both the ranges of 10% to 20% relative humidity and 80% to 90% relative humidity, with a worst case being ±3.0 %RH in the ranges of 0% to 10% relative humidity and 90% to 100% relative humidity. The overall performance of this sensor at 25ºC was worst case accuracy of ±6.0 %RH with a typical measurement having an accuracy of ±1.8 %RH. [4]

The SHT41 had a significant overall improvement at 25ºC with worst case accuracy being at ±3.0 %RH and typical accuracy remaining at ±1.8 %RH just like the SHT40. This sensor does perform better in the range of 0ºC to 80ºC in more extreme relative humidity percentages, remaining at a typical accuracy of ±1.8 %RH between 30% and 80% relative humidity, and having an accuracy of ±2.0 %RH in the remaining ranges of 0% to 30% relative humidity and 80% to 100% relative humidity. [4]

The SHT45 had the best overall performance overall, with a slight improvement in the worst-case accuracy which is at ±2.0 %RH. The typical running accuracy made a significant improvement however, decreasing to just ±1.0 %RH compared to both the SHT40 and SHT41, making the worst-case accuracy and the typical accuracy relatively close in value at 25ºC. The performance of this sensor is also significantly better in the 0ºC to 80ºC range, with the typical running accuracy being ±1.0 %RH between 20% and 0% relative humidity, with only a decrease to ±1.5 %RH between 0% and 10% relative humidity and 80% to 90% relative humidity. The worst accuracy is between 90% and 100% relative humidity at ±1.75 %RH, which gives this sensor overall a worst-case accuracy which remains less than the typical accuracy of both the previously mentioned sensors. [4]

With this information in mind, as mentioned in section 3.6.1 regarding the temperature sensor, the main motivation for selecting the SHT45 over the SHT40 or SHT41 was its significant performance increase in regard to the typical running accuracy for the humidity sensor, which is just ±1.0 %RH making this sensor extremely accurate, and for our expected operating conditions being below 80% relative humidity, this sensor should almost always perform in its typical accuracy.

3.1.9 Soil Moisture Sensor Technology Comparison:

The technology behind soil moisture sensors is very similar to humidity sensors with both sensor types measuring an amount of water. The main difference being that humidity sensors measure the water contained in the air and the moisture sensor measures the liquid content in or on a particular surface such as soil. Soil moisture sensors typically do not actually measure directly like a humidity sensor, instead relying on some other metric related to the water content in the soil via probe to provide a very good estimate of the moisture level. [32]

The most common types of soil moisture sensors are Coaxial Impedance Dielectric Reflectometry soil moisture sensors, some form of capacitive soil moisture sensor, Time domain reflectometry soil moisture sensors, and Frequency domain reflectometry soil moisture sensors. [32]

Coaxial Impedance Dielectric Reflectometry (CIDR):

This type of soil moisture sensor works by utilizing some type of oscillator to create an electromagnetic signal which is sent through the soil. The sensor then measures the ratio of the reflected voltage compared to a different voltage signal which is based on the impedance of the probes within the soil. This is the most common type of soil moisture sensor for commercial agricultural applications.[36]

Capacitive soil moisture sensors:

This type of soil moisture sensor works very similarly to how the capacitive humidity sensor described in section 3.1.7 works. These sensors are typically made of materials which have a higher water absorption capability such as a metal oxide, which changes the capacitive value of the capacitor itself. This type of sensor when used in soil moisture sensing applications measures the actual capacitance itself. [33][34][35]

Time Domain reflectometry soil moisture sensors (TDR):

This type of soil moisture sensor works by sending high frequency signals transmitted across probes in the soil and superimposing the sent signal with the return signal resulting in a soil moisture reading. The signal measurement of the two signals is the dielectric constant which has a linear relationship with soil moisture. These types of sensors are very accurate as they are not impacted by factors such as temperature and soil density, due to the nature of these sensors they are extremely complex and expensive to manufacture. [33][34]

Frequency Domain reflectometry soil moisture sensors (FDR):

This type of sensor utilizes the capacitive moisture sensor technology described above, but instead of measuring the capacitance in the material, it measures the electromagnetic waves generated by a capacitive probe. This type of sensing also measures the dielectric constant but instead of measuring the time measures the frequency difference between the signals. This results in a faster measurement time and provides the same benefits as time domain reflectometry but at a cheaper price since it utilizes capacitors which are much cheaper than the complex equipment needed for time domain reflectometry. [33][34]

Table 3.1.9-1. Moisture Sensor Technology Comparison

Specification	CIDR	Capacitive	TDR	FDR
Accuracy	High	Low	High	High
Response Time	Fast	Fast	Slow	Fast
Stability	High	High	High	High
Price	High	Low	High	High

Considering the factors that are necessary for our monitoring system, we need something small, relatively accurate and stable, and cheap. Almost all the sensor types available for soil moisture sensing are commercial grade, which makes them very expensive and more than is necessary for the scope of what our system is doing regarding the soil moisture sensing. Capacitive soil moisture sensors were the most readily available that were cheap with a small size for our system, so we selected this type of soil moisture sensor.

3.1.10 Soil Moisture Sensor Parts Comparison:

Table 3.1.10-1. Moisture Sensor Parts Comparison

Specifications	PIM520	Adafruit soil sensor	Grove sensor
Input voltage	3.3V	3V – 5V	3.3V – 5V
Communication	PulseWidth Frequency	I2C	I2C
Dimensions	99 mm x 10 mm x 5 mm	76.2mm x 14.0mm x 7.0mm	92.1mm x 23.5mm x 6.5mm

Price	$4	$7.50	$6.50

For the soil moisture sensor as discussed in section 3.1.9 we went with a capacitive moisture sensor for its cheaper price compared to the other options available. Finding a sensor for our system was rather challenging with only three main options available that would work in the way we needed for this system. The three sensors we found were the Pimoroni PIM520, the Adafruit soil sensor, and the Grove soil sensor.

The Grove soil moisture sensor is a basic capacitive soil moisture which utilizes a NE555DR integrated circuit to read the capacitance difference in the sensor material and transmits the data via I2C. This sensor did not provide a detailed datasheet aside from the dimensions and input voltage. [39]

The Adafruit soil moisture sensor is like the Grove soil moisture sensor with the communication protocol being I2C and having a similar input voltage range of 3V to 5V. This sensor utilizes the ATSAMD10 integrated circuit for the capacitive difference measurement and gives a reading between 200 and 2000 indicating very dry and very wet. This sensor package also includes an ambient temperature sensor which was not a factor in selection as we already had a highly accurate temperature sensor selected. [38]

The Pimoroni PIM520 soil moisture sensor like the other two sensors was a capacitive type of sensor and did not specific exactly which type of integrated circuit it utilizes for the capacitance measurement and does utilize a different way of transmitting the data, instead of I2C it utilizes a pulse width frequency modulation. This system transmits a lower frequency smaller hertz value when the sensor detects a wetter condition and a high frequency larger hertz value when there is a drier condition. This sensor also does come with some built in calibrations but benefits from calibrating to the specific system case. [37]

Overall considering the limited information available for all these soil moisture sensors, for the price the Pimoroni PIM520 provides the best system for monitoring the soil the product is in. Since the soil moisture should not be constantly fluctuating and really a general idea of the condition is necessary for this aspect of the system, the lack of any accuracy specifications provided for this sensor should not impact the functionality. This sensor did not provide us with the desired I2C communication protocol, which is necessary for our system, instead providing the pulse width frequency output as mentioned previously. This means that going with this sensor may potentially increase the current drawn as instead of utilizing the two pins established with I2C communication (SDA and SCL) will not be used for this sensor. Instead, we will need to incorporate an extra general-purpose input / output pin to the system to send and receive data. Despite this fact, considering that the Pimoroni PIM520 sensor provides the cheapest price and does not have too much of a performance difference that we could tell between the sensors on the market, we decided to go with this sensor and will attempt to optimize our system so that the added general-purpose input / output pin does not result in a significantly higher power draw.

3.1.11 MCU Selection:

Table 3.1.11-1. MCU Comparison

Specifications	ESP32-S3	ESP32-C3	SAMA7G54	Raspberry Pi Zero
Frequency	2400MHz ~ 2500MHz	2400MHz ~ 2500MHz	~1GHz	~1GHZ
Operating Temp	-40ºC ~ 85ºC	-40ºC ~ 85ºC	-40ºC ~ 105ºC	-20ºC ~ 70ºC
Storage Temp	-40ºC ~ 125ºC	-40ºC ~ 125ºC	40ºC ~ 105ºC	N/A
Power Range	3.0V – 3.6V	3.0V – 3.6V	3.3V – 5.5V	5V
Interfaces	UART, GPIO, ADC, PWM, I2C, I2S, SPI, LCD, DVP, RMT, SDIO, USB, MCPWM, DMA, TWAI	UART, GPIO, ADC, PWM, SPI, I2C	USB, I2C, SPI, EBI	N/A
SRAM	512KB	400KB	128KB	512KB
PSRAM	32Mb	N/A	N/A	N/A
GPIO Pins	38	22	136	40
ROM	384KB	N/A	80KB	N/A
CPU	Dual Core Xtensa LX7	Single Core RISC-V	Single ARM Core	Single quad core ARM CPU
Wi-Fi	Wi-Fi 4.0	Wi-Fi 4.0	N/A	Wi-Fi 4.0
Bluetooth	Bluetooth 5.0	Bluetooth 5.0	N/A	Bluetooth 4.2
Sleep Amperage	8µA – 240µA	5µA – 130µA	N/A	80mA – 100mA
Price	$1.85 per chip	$1.00 per chip	$13.00 per chip	$15.00 per chip

For the microcontroller unit there were several options to choose from with many different pros and cons and specifications. This section will not have any technology comparison for reference as comparing the technologies for the microcontroller unit was out of the scope of this project and did not have any real weight in the end decision of which microcontroller we decided to use. Instead, the decision of the microcontroller was based on a few different criteria being the features we required for the monitoring system, the operating specifications / requirements, the communication protocols, and the price per microcontroller unit.

The specific features we needed for our microcontroller unit consisted of having Bluetooth capabilities and Wi-Fi capabilities as well as being able to communicate via I2C and having at least one GPIO pin which could accept a pulse width frequency signal. The Bluetooth and wi-fi capabilities were necessary as our system will need to be able to transmit the sensor data and the camera readings via wi-fi to our web application for the end user to see as well as transmitting sensor information between the two microcontroller units. The operating specifications were not too strict as the main consideration was an operating temperature between -20ºC and 60ºC, for most microcontroller units the operating range is well within this specification, and it wasn't difficult for us to find some options which met the requirement.

Looking over several of our options, our final group of potential microcontroller units came down to the ESP32 family of processors with a few different options including the ESP32-S3 and the ESP32-C3. Both microcontrollers ended up being a part of our system with different functions to meet the needs of the system.

The ESP32-S3 was our main choice for the microcontroller unit as it is a system on a chip which provides us with all the necessary specifications for our system and includes built-in wi-fi and Bluetooth capability. Finding a microcontroller unit with Bluetooth and wi-fi capability was rather difficult as there are not too many options on the market, with two of the main options including the ESP32 being manufactured by the same company being Espressif. The ESP32-S3 is powered by a dual core Xtensa microprocessor and has a RISC-V coprocessor and provides 512KB of SRAM and 384KB ROM to work with and allows the use of an additional 32MB of PSRAM. With all the features and peripherals, this microcontroller comes to be only $1.85 per chip making it an easy choice for our applications. [40]

The ESP32-C3 is very similar to the ESP32-S3, providing the same wi-fi and Bluetooth capabilities with the main difference being the actual specifications of this microcontroller unit. The ESP32-C3 is powered by a single core RISC-V microprocessor and only has 400 KB of SRAM available with no additional ROM or PSRAM capabilities. [41]

Considering the capabilities of both of these chips, we utilized both in our application with the more powerful ESP32-S3 handling the camera package as it has the additional PSRAM and the ESP32-C3 handling the sensor package as it is lower cost, and it isn't necessary to have the extras that come with the ESP32-S3 for processing the simple sensor data. [40][41]

When considering the camera itself, we had some issues with finding cameras which did not have an IR filter, which resulted in us needing to rethink the camera options that were available to us at the time. The camera that met our requirements was the raspberry pi cam which only interfaces with MIPI CSI-2 which is not exactly compatible with the ESP32-S3 or ESP32-C3 as it only support MIPI CSI-1, which led to us looking at alternative options to either replace the ESP32-S3. Before settling on the pi cam, we had some other options we were considering but also did not work with the ESP32-S3, which led to us finding the SAMA7G54 and the raspberry pi zero as our options. The SAMA7G54 had a lot less to offer than the ESP32-S3 for our project purposes and had a dev-kit which was unattainable which led us to move to the raspberry pi which we decided will handle just the image capture and processing and send this to the ESP32-S3. [40][41][42][43]

3.1.12 Load Power Requirements for Power Subsystem

To give a clearer idea of the requirements for the power subsystem the following chart will break down the load current draw and voltage level for each component selected. These are the power levels that the power supply must supply.

Table 3.1.12-1. Load Power Requirements for Power Subsystem

Component	Current	Voltage
ESP32-S3 (TX)	300mA Peak Current Nominally Lower (around 100 mA)	3.3V
ESP32-S3 (RX)	100mA Peak Current Nominally Lower (around 30 mA)	3.3V
ESP32-S3 (Modem Sleep)	13-23mA	3.3V
ESP32-S3 (Light Sleep)	240uA	3.3V
ESP32-S3 (Deep Sleep)	7-8 uA	3.3V
Raspberry Pi Zero 2 W + Camera (Active)* [48][52]	220-260mA	5V
Raspberry Pi Zero 2 W + Camera (Idle)* [48][52]	80-100 mA	5V
AS7331 (Active)	1.5mA	3.3V
AS7331 (Idle)	970uA	3.3V

Theoretical Worst Case	561.5 mA	N/A
Practical Worst Case	181.5mA	

**The current consumption numbers for the Pi Zero 2 W are not from official Raspberry Pi documentation as those do not exist but rather from several independent tests.*

In the above table the load requirements for each part on the camera package are listed. The power source for this package will be a solar panel and backup Li-ion battery. The theoretical worst case refers to the possibility of the MCU getting stuck on a line of instructions and remaining in the active (TX) state while experiencing the peak current consistently with the Raspberry Pi Zero 2 W and camera active. The practical worst case refers to the MCU getting stuck in an active state with the Raspberry Pi Zero 2 W idle. This is the most likely worst-case scenario since the camera and Raspberry Pi will only be active three times during the day for a couple seconds.

Table 3.1.12-2. Sensor Package Load Requirements

Component	Current	Voltage
ESP32-C3 (TX)	280mA Peak Current Nominally Lower (around 100 mA)	3.3V
ESP32-C3 (RX)	80mA Peak Current Nominally Lower (around 30 mA)	3.3V
ESP32-C3 (Modem Sleep)	13-26mA	3.3V
ESP32-C3 (Light Sleep)	130 uA	3.3V
ESP32-C3 (Deep Sleep)	5uA	3.3V
SHT 45 (Active)	50uA	3.3V
SHT 45 (Idle)	.08uA	3.3V
PIM520 (Active)	TBD	3.3V

PIM520 (Idle)	TBD	3.3V
Theoretical Worst Case	280.05 mA	N/A
Practical Worst Case	100.05mA	

In the above Table 3.1.12-2 the load requirements for each component on the sensor package are listed. The power source for this package will be a single 9V battery. The theoretical worst-case scenario refers to a potential situation where the MCU and peripherals get stuck in an active state while experiencing peak TX current consistently. The practical worst case refers to a potential situation where the MCU and peripherals get stuck in an active state while experiencing nominal currents. The current load for the PIM520 will need to be determined through our own independent testing in the next semester as these figures are not provided by the manufacturer.

3.1.13 Solar Panel Technology Comparison

To provide an autonomous and easy power source the primary power generator for our system will be a solar panel. Our solar panel needs to be powerful enough to fully power the system when active during peak solar input. Our panel also needs to boast high efficiency so that the backup battery will remain charged and can power the system at night or periods of low solar input. When deciding on a solar panel to use for our project there are three main photovoltaic cell (PV) technologies to choose from which are monocrystalline, polycrystalline, and thin film.

Monocrystalline Cells:

Monocrystalline cells are the oldest form of silicon-based PV technology commercially available. They are a single silicon crystal lattice and require extremely pure silicon to manufacture. They are created by seeding a mass of molten silicon to create a cylindrical ingot that can then be cut into wafers. Monocrystalline cells tend to be more expensive to manufacture and more efficient than other technologies.

Polycrystalline Cells:

Polycrystalline cells are a cheaper, less efficient way of designing a crystalline silicon PV cell. Unlike their monocrystalline counterparts, polycrystalline cells have many different “grains” or “shards” of crystal lattices connected in one cell. They are created by simply casting molten silicon into a cube shape before cutting it into wafers. Polycrystalline cells tend to be the most popular PV cell technology.

Thin Film Cells:

Thin film cells are the last main PV technology widely available although they tend to be less popular than crystalline technologies. Thin film cells as a category refers to a few different technologies which all share roughly the same properties. Thin film cells drastically cut down on the actual silicon used in the manufacturing process reducing the silicon atoms to simply a thin film. This greatly reduces the cost of the cells but also greatly reduces the efficiency. Thin film cells also tend to be flexible and more durable than crystalline cells.

In the table below we will analyze the differences in the three topologies and their tradeoffs.

Table 3.1.13-1. Solar Cell Technology Comparison

Parameter	Monocrystalline	Polycrystalline	Thin Film
Price	High	Moderate	Low
Durability	Low-Moderate	Low-Moderate	High
Efficiency	High	Moderate	Low
Flexibility	Low	Low	High
Efficiency Decrease Over Time	Low	Moderate	High

Price: Monocrystalline is the most complex PV cell to manufacture and it requires the purest silicon. As such monocrystalline is the most expensive to create. Polycrystalline does not require the same level of purity or the same care to ensure a single crystal lattice when manufacturing and as such is cheaper. Thin film cells use the least amount of silicon and as such tend to be the cheapest to produce.

Durability: The crystalline structures of monocrystalline and polycrystalline cells make them less resistant to various forms of trauma. However, this can be mitigated by the packaging of the PV cell. Thin film cells are more resistant due to the lack of more defined crystal structure.

Efficiency: Efficiency refers to the percentage of solar power that can be converted into usable energy by the PV cell. Monocrystalline cells tend to be the most efficient with up to 20% efficiency. Polycrystalline cells tend to be less efficient because of their multiple crystal lattices with typical efficiencies of around 15-17%. Thin film PV cells lack the defined crystal structure of monocrystalline and polycrystalline cells and have a higher level of impurity. As such thin film cells sacrifice efficiency with typical efficiencies as low as 10-13%.

Flexibility: Mono and polycrystalline cells both require that the crystal structure is maintained to retain their functionality and as such are not flexible. Thin film cells have much more flexibility because they only have a thin film of silicon atoms.

Efficiency Decrease: The impurities of thin film cells lead to a decrease in efficiency over time at a much higher rate than the crystalline options.

Conclusion: For our project we will use a monocrystalline PV cell for our solar panel. For a panel of the size needed for our project the price difference between the technologies is not significant enough to justify the decreases in efficiency. Efficiency is incredibly important for our project since we need the solar panel to act as the power source for as long as possible. With these factors considered there is no reason to select the polycrystalline or thin film technologies.

3.1.14 Solar Panel Parts Selection

For our system the main power source for the camera package will be the solar panel. This panel needs to be able to fully power the system when active and charge the backup battery when the system is in idle. The solar panel must also be able to achieve a voltage beyond the given minimum input voltage of the charge controller (3.7V). Beyond these requirements we are very flexible with the part selection for the solar panel.

Table 3.1.14-1. Solar Panel Parts Comparison

Parameter	5856 [5]	5368 [59]	P105 [51]	NPA5S-12I [53]
Peak Voltage	6V	6V	6V	17V
Peak Current	120mA	200mA	940mA	350mA
Wattage	.6W	1.2W	5W	5W
Price	$8.95	$14.95	$35.00	$21.49

The 5368 solar panel was enough to charge the backup battery during idle times when we were considering batteries at a lower nominal voltage, but with our current design it will be unable to properly charge the battery since it is not powerful enough. The 5856 solar panel was being considered when the Ov5640 camera was being considered as the power draw for the overall camera package would be significantly lower. However, with the Pi Zero 2 W and Pi cam the power requirements for the system are far beyond what this panel would be able to provide. The P105 was the largest competitor for the selection of a solar panel for this design. If it was used a boost mode battery charger would need to be used to provide a high enough battery charging voltage. However, for our design since we want to utilize a NVDC DPPM charge controller it is not possible to use a boost mode charge controller. Given this consideration, the peak voltage of the P105 is too low to properly charge the batter we selected which requires 8.5-7.2V of charging voltage. To fully fill the needs for our design the NPA5S-12I solar panel was selected. This will provide a high enough voltage to charge the battery. Additionally, since it is a 5W panel it will provide enough current to the load after being stepped down by our buck regulators even though the max output current is only 350mA.

3.1.15 Power Regulators/Charge Controller

To provide proper load power to each of the IoT components and the MCU on the sensor package the voltage output battery must be regulated. Additionally, the MCU and IoT devices require the 5V output of the charge controller be regulated down to 3.3V. Our application is a low-power system focused on efficiency and minimal intervention from the end user. To achieve these requirements there are a variety of regulator architectures to choose between. The three most popular topologies are buck, boost, and LDO regulators.

LDO Regulators:

LDOs or low dropout voltage regulators are linear regulators that focus on providing the load with a constant output voltage. They are typically used when the load needs to be very low noise and the power supply is not steady or is noisy. However, since LDOs are linear regulators when the input voltage is not close to the output voltage the need to dissipate heat is much larger and as such the power lost in this dissipation is larger yielding a lower efficiency.

Buck Converters:

Buck converters are step-down switching converters that focus on high efficiency voltage regulation. Since buck converters are switched, they introduce noise in the form due to capacitor resistances as well as load inductance and capacitance. In low-noise applications this can reduce performance. However, since buck converters are switching regulators, they are capable of both stepping up current and increased efficiency (can easily achieve above 90% efficiency).

Boost Converters:

Boost Converters are step-up switching power regulators that are typically used when the input voltage is less than the required load. Like buck regulators they introduce switching noise at the load. Similarly, since boost converters step up voltage, they require that current be stepped down. Boost converters also tend to be high efficiency converters similarly to buck converters.

In the Table 3.1.15-1 below we will analyze the differences in the three topologies and their tradeoffs.

Table 3.1.15-1. Power Regulator Technology Comparison

Parameter	LDO	Buck	Boost
Efficiency	Moderate to Low	High	High
Noise	Low	Moderate to High	Moderate to High

Step-Up/Down	Down	Down	Up
Heat Loss	Moderate to High	Low	Low
Complexity	Low	High	High
Cost	Low	Moderate to High	Moderate to High
Input Range	Narrow (without efficiency loss)	Wide	Wide

Efficiency: The efficiency of LDO regulators relies heavily on the ratio of input to output voltage. Buck and boost converters both have high efficiencies even at large differences between input and output voltage.

Noise: Since LDOs are linear regulators they lack the switching noise introduced by buck and boost converters. In our use case noise is only a concern for one of the camera reference voltages.

Step-Up/Down: Boost converters focus on being step-up converters in contrast to LDOs and buck converters. For our use case the battery on the sensor package and the input to the camera supplies will be higher than the supply voltages needed so we do not need a step-up converter.

Heat Loss: The excess power dissipated when stepping down voltage in LDOs is dissipated through heat whereas the switching regulators do not need to dissipate energy through heat and most heat loss is a byproduct of low efficiency. For our design low heat loss is important since heat build-up is a concern.

Complexity: LDOs typically require very few external components to properly realize their design making them easy to design with. Buck and boost converters require many more supporting components to filter out noise and realize the topologies. Since we have a large amount of time to design our product complexity is not a major concern.

Cost: In terms of ICs all three topologies have similar costs. However, since buck and boost converters require a larger number of supporting components, they can cost more to fully realize. For us the cost of the regulators is relatively minor in comparison to other components and as such slightly higher costs can be considered.

Input Range: LDOs have a smaller input range in general but when seeking a high efficiency regulator, the range where LDOs can be optimally used is even narrower. Buck and boost converters both boast large input ranges which typically span tens of volts. For us the input range for the sensor package will be high since the battery on the sensor package will be a significantly higher voltage than the supply requirement. On the other hand, the camera supply voltages will have a relatively low input output delta.

Conclusion:

For our system we will be utilizing a buck regulator on the sensor package and camera package since we need a high efficiency solution with an input voltage much higher than the output. The charge controller we selected also doubles as a buck-boost converter (meaning it can be used as either a buck or boost converter) and it will regulate the supply voltage for the Pi Zero 2 W.

3.1.16 Power Regulator Parts Selection

Buck Regulators:

For our project we will use a buck converter on the sensor package to regulate the voltage for the MCU and IoT sensors and a buck regulator on the camera package to regulate the voltage for the MCU and UV sensor. We will use an additional buck regulator to regulate the output of the charge controller to the 5V needed for the Pi Zero 2 W. The main design goal with our selection is to have all outputs provided at a high efficiency with low quiescent current and cost. The emphasis for our design is on maximum efficiency and economy so simpler, more efficient regulators are more desirable than more complex or less efficient ones.

Table 3.1.16-1. Power Regulator Parts Comparison

Parameter	LMQ644A2-Q1 [50]	TPS566231 [49]	TPS62932[9]	MCP16502 [8]	LM64460-Q1 [64]
Number of Outputs	2	1	1	6	1
Quiescent Current	40uA (typ)	50uA (typ)	12uA (typ)	.12-16 mA (typ)	7uA (typ)
Input Voltage Range	3-36V	3-18V	3.8-30V	N/A	3-36V
Output Voltage Range	0.8-20V	0.6-7V	0.8-22V	1.2-3.7V (Buck Pin 1) .6-1.85 (Buck Pins 2-4) 1.2-3.7 (LDO Pins)	1-34.2V
Complexity	High	Moderate	Moderate	High	Moderate

Number of Additional Components Required	36	14	9	25	12
Approximate Efficiency (3.3V Out)	95%	97%	95-97% (Depending on load current)	20-95% (Varies greatly with load current)	88% (Heavily variable load)

For this project we will use the TPS62932 buck regulator for both 3.3V outputs. We chose this regulator because of its excellent efficiency and low quiescent current as well as its lower cost and complexity. The LMQ644A1-Q1 is more expensive with efficiency that does not justify the increase in cost and complexity. The LMQ644A1-Q1 was chiefly being considered early in the design phase when we were unsure how many different output voltage levels would be required and what those levels would be. The LMQ644A1-Q1 would be slightly easier to work with and more efficient if we required two different output voltages and the input voltage was higher. The MCP16502 was selected as it is the power management IC recommended for use with the SAMA7G5 series MCU. The six output pins provide easy regulation to the multiple different functions required by the MCU and the efficiency when used with this MCU is also quite high. However, since we ultimately did not need an MCU with CSI2 we are no longer able to justify the selection of this IC. As a standalone IC the MCP16502 is overkill for the scope of this design. The TPS566231 was the closest competition to the TPS62932. It also has very good efficiency with a lower level of complexity. However, the higher quiescent current and larger number of components make it slightly worse for our purpose. The larger BOM for this regulator would increase costs needlessly and the slight increase in efficiency may well be lost in the increased quiescent current under real-world conditions since the power draw is so low. Overall, the TPS566231 was just slightly too complex to justify use in this design. Ultimately the TPS62932 provides a very efficient solution with a relatively simple design and a very low cost. This fills our needs precisely since we are optimizing for a low-power low-cost system.

For our design we will use the LM64460-Q1 regulator for the highly variable 7.2-17V input which will provide a 5V output. This regulator was selected because of its excellent efficiency across the wide range of input voltages. This contrasts with the TPS62932 and the TPS566231 which provide a higher efficiency at the lower input variances associated with the 5V rail and the 9V battery. These two regulators tend to have poorer overall efficiencies when used across a wider range of input voltages. The LM64460-Q1 on the other hand tends to fare worse with the lower output voltages and currents associated with the 3.3V rails. As such the LM64460-Q1 will only be used for the 5V rail. The MCP16502 and LMQ644A2-Q1 will not be used for this rail for the same reasons addressed above when discussing the 3.3V rails.

Charge Controller:

For our project we will use a charge controller to regulate the output of the solar panel, charge the rechargeable battery, monitor, and supplement output of the solar panel when solar output is low. The controller should be able to act as a voltage regulator as well as a battery charger. The controller should monitor the load power draw and select the proper power source (solar panel or back-up battery) based on the power being output by the solar panel and the load required power.

Table 3.1.16-2. Charge Controller Comparison

Parameter	BQ25756E [1]	BQ24210 [58]	BQ25798 [57]
Input Voltage Range	0-60V	3.5-7V	3.6-24V
MPPT Support	Yes	Yes	Yes
Charger Topology	Buck-Boost Switch Mode	Linear	Buck-Boost Switch Mode
Controllability	I2C	Standalone (RC Settable)	I2C
Number of Series Battery Cells Supported	1-7	1	1-4
Power Topology	Non-Power Path	Non Power-Path	NVDC Power-Path
Price	4.95$	2.052$	4.21$

Our system is primarily powered by the solar panel and the battery will be charged via the solar panel. For the charge controller to accept the solar input without an unacceptable loss in efficiency it needs to support maximum power point tracking (MPPT) [67]. Additionally, to have the system run at maximum efficiency the MPPT algorithm should be able to communicate with the MCU to run calculations for its MPPT algorithm. For our system configuration the system and charge controller will be constantly powered throughout the day when solar power is above the minimum required level for the system. This means that our charge controller needs to be able to compensate for this constant power to avoid a reduction in battery lifespan. For this purpose, our charge controller should support power-path management. Power-path management allows the charge controller to isolate the input current to the system and the charge current to the battery meaning that when the battery is fully charged the charge controller will avoid directing extra current to it. Additionally, since we are using a solar input, and the solar source will dip below load requirements while still providing some level of output. Because of this behavior narrow-voltage DC (NVDC) power-path management is desired. NVDC power-path management allows the battery to act as a supplemental power source when the primary source falls below load requirements. This contrasts with a simple OR management system which would simply disable the primary power source when it falls below load requirements. Finally, the charge controller will also regulate the output voltage and current to the system. This makes a switched mode topology more desirable because of the higher efficiencies discussed in the technology comparison section.

Taking the factors mentioned above into account we selected the BQ25798 charge controller. This is because its implementation with NVDC power-path managements will allow it to fill our design requirements. The BQ25789 controller meets all our needs exactly. The only slight downside to this controller is that we don’t need support for a 4-cell Li-ion battery meaning a topology with less functionality at a lower price point could be manufactured. The BQ25756E controller was being considered because of its phenomenal input voltage range which is especially important due to the variability in supply voltage inherent to solar panels. However, this controller does not support power-path management meaning that it would erode the lifespan of the backup battery far faster than the selected controller. Additionally, it only supports a simple OR charge control system which would simply disable the solar input when the solar supply power drops below load. This is not desirable because with the variability in power output from a solar panel this would result in lower power efficiency. Finally, the BQ24210 controller was our low-cost economic option. It is about half the price of the other two charge controllers. However, this reduction in cost comes with a trade-off in efficiency. The regulator topology is a linear LDO, and it lacks the same features that the BQ25756E lacked. The downsides of the LDO topology is discussed in more depth in the technology comparison section, but a linear regulator would result in a large loss of power in our system since the solar panel’s nominal voltage falls well above the system's operating voltage.

3.1.17 Batteries Technology Comparison

To ensure power is constantly supplied to the final design a backup battery is needed to supplement the solar panel. For our application this backup can be recharged each day by the solar panel and will mostly act as a supplement during dark hours and periods of low solar input. The sensor packages also need to be powered from a battery with a sufficient lifespan. When choosing a battery for our voltage level and size requirements there are four main types to consider which are Li-ion, NiMH, lithium, and alkaline.

Li-Ion:

Li-ion batteries are one of the most popular types of rechargeable battery. They tend to have a very high energy density with higher output voltages. Li-ion batteries can vary quite largely on their lifespan depending on the output voltage and load current.

Lithium:

Lithium batteries differ from Li-ion batteries by being a primary cell meaning they are not rechargeable. Lithium batteries tend to have a very high energy density.

NiMH:

NiMH (nickel-metal hybrid) batteries are another one of the most common types of rechargeable battery. They tend to have lower energy density and shorter lifespans. NiMH batteries typically support lower voltages than other popular rechargeable battery types.

Alkaline:

Alkaline batteries are a modern type of primary cell battery. They have become the replacement for zinc-carbon batteries. Alkaline batteries are the easiest to find in consumer stores and come in many voltage levels with moderate to lower energy density.

In the table below we will analyze the differences in the three topologies and their tradeoffs.

Table 3.1.17-3. Battery Technology Comparison

Parameter	Li-ion	NiMH	Lithium	Alkaline
Lifespan	Low-High	Low-Moderate	N/A	N/A
Energy Density	Moderate	Low	High	Moderate
Available Voltage Range	Moderate	Wide	Wide	Wide
Internal Resistance	Low	Low	Low	Moderate
Price	High	Moderate	High	Low
Usability by the End User	Low	High	High	High

Lifespan:

Lifespan in rechargeable batteries generally refers to the number of times that a battery cell can be recharged. As such this category does not apply to non-rechargeable battery technologies. Li-Ion batteries have a highly variable lifespan that can range from low (around 100 cycles) to long (>2000 cycles). Similarly, NiMH batteries can have a similarly short lifespan but tend to have low maximum lifespans of around 1000 cycles.

Energy Density:

Energy density refers to the amount of energy in a battery per unit of size. Higher energy density generally allows for higher mAh and mWh ratings for equal package size. Li-ion batteries generally have moderate energy density and tend to be the most energy dense of the rechargeable cells. NiMH tends to be less energy dense and require recharging more often. Lithium batteries have excellent energy density and tend to have the highest energy density of all the popular battery technologies. Alkaline batteries also have moderate energy density somewhat comparable to Li-ion batteries.

Voltage Range:

Voltage range refers to the range of voltage levels that each battery technology can be found in. For instance, Li-ion batteries typically don't have voltages below around 2.4V. This voltage range metric is specifically regarding voltage levels that apply to our project. In a broader sense Li-ion has a much larger range of potential voltages than alkaline batteries but those voltages fall far outside of levels that we require for this project.

Internal Resistance:

Internal resistance refers to the internal resistance of the battery. Higher levels of internal resistance can reduce the expected discharge time of the battery at lower loads. Both rechargeable batteries and the lithium batteries all tend to have very low internal resistance. Alkaline batteries have a notably higher internal resistance than the other battery technologies.

Price:

Price refers to the average price of each technology for package sizes/voltage levels. Both the Li-ion and lithium batteries tend to be the most expensive options. NiMH is the lower-cost rechargeable option and alkaline batteries tend to be very cheap.

Usability:

Usability refers to the ability for the end user to obtain and replace each battery technology. Li-ion batteries typically do not come in standardized formats like "AA." As such it is difficult for the end user to replace or maintain a Li-ion battery cell. Alkaline NiMH both come in all the typical formats. Lithium batteries don't generally come in the lower voltage formats, but they do come in the standard 9v format.

Conclusion:

For our system we will use a Li-ion rechargeable battery for the solar panel backup battery and a lithium battery for the sensor packages. The Li-ion type was selected because the backup battery must have a good lifespan and be able to store a larger amount of power. A longer lifespan ensures that the end user won't have to consider replacing it for a long time and a larger energy density means the battery can act as a better solar backup while still meeting our space requirements. We selected the lithium battery for the sensor packages because we want a battery which will last for a very long time. The idea behind the power management for this system is to create power supplies that the user does not need to think about. For this purpose, a lithium battery works best. This could be replaced by an alkaline or NiMH battery by the end user once the lifespan of the original battery is up.

3.1.18 Battery Parts Selection

For our project the exact specifications of the Li-Ion backup battery are not actually of major import. The main requirement is that the system runs on backup power for 2 days. For our components that is a relatively easy target to achieve since the system is at idle most of the time and has a very low current draw even when active. The main requirement for the battery is instead that it functions with our charge controller. Since the battery has far less in terms of high-level functionality than the charge controller, we decided to allow our choice in charge controller to dictate the selection of the back-up battery. The main specifications that will determine our choice of battery are cost, maximum charge voltage, and capacity. Maximum charge voltage should fall above the charge controller's minimum charge current and minimum charge voltage which is 3V.

Table 3.1.18-4. Backup Batteries Parts Comparison

Parameter	LIPO 552530	US18650VTC6 [55]	PRT-13851 [56]	BL0750F503048 1S1PCMC [54]
Capacity	350mAh	3000 mAh	400mAh	700mAh
Max Charging Current	175mA	5000mA	400mA	700mA
Standard Charging Current	70mA	1000-3000mA (depending on temperature)	80mA	140mA
Max Charging Voltage	4.2V	4.2V	4.2V	4.2V
Lifespan	500 Cycles at .5C discharge and charge current and 80% discharge	Not provided by the manufacturer.	300 Cycles at .5C discharge and charge current and 80% discharge	300 Cycles at .5C discharge and charge current and 80% discharge

Price	$5.95	6.49	$5.50	$11.18

Our main requirement for the battery is simply the 2-day discharge specification. The output voltage of any single cell battery will be lower than what is required to power the 5V rail so any of the batteries being discussed will be treated as a 2-cell battery pack consisting of two of the single cells shown above placed in series. We went with the US18650VTC6 battery because with the alteration of design to use the Pi Zero 2 W we require a higher power draw even during idle periods. The BL0750F5030481S1PCMC battery would provide sufficient power for a typical use if we experienced no periods of solar downtime or did not have the 2-day discharge time specification. However, with our design this cell would only last the idle system around 6-18 hours depending on when it was activated and the number of MCU wakeups that occurred during that time. The LIPO 552530 and PRT-13851 batteries were initially being considered for the Ov5640 camera as it would not require the higher power draw Pi Zero 2 W and would not require a 5V power rail. This, however, wound up being out of stock and as such we had to adjust. In doing so these batteries became mostly inefficient. We could use either one of them as a 4-cell battery pack with two pairs of two parallel batteries in series, but that would ultimately be more expensive than just using the US18650VTC6.

We will not have a parts selection section for the primary cell in the sensor package. This is because we will be using a 9V battery and any battery in this format will be acceptable. Our initial selection will be a high-capacity lithium primary cell 9V battery, but this selection is not a necessary component of our design. An alkaline 9V battery would be just as acceptable or even a 9V Li-ion battery assuming the size was acceptable (although recharging it would be impossible). With the selected power regulator for the sensor package a wide range of voltages can be used, albeit with an efficiency drop when voltage differs from the designed voltage, meaning a highly accurate battery voltage is not essential either. This essentially means that the parts selection for this battery is meaningless since any comparable battery of the same voltage level is acceptable with the only parameter that changes being the battery life. Additionally, when comparing 9V batteries of the same chemistry the differences are minute and most of the batteries examined would be just as valid for our final design. Because of these factors a parts comparison would present an inaccurate view of the design process for this component as the choice we ultimately wind up making will be based on availability.

3.2 SOFTWARE

3.2.3 Image Processing Technology Comparison

In the pursuit of efficient plant health assessment, the selection of image processing technologies plays a pivotal role in accurately handling NDVI data, identifying the plants we are monitoring, and detecting disease/insect issues. In this section, we delve into three distinct image processing technologies and evaluate their applicability for these tasks.

Thresholding Techniques:

Thresholding techniques are fundamental in image processing for segmenting regions of interest based on intensity thresholds. They are simple and computationally efficient, making them suitable for binary segmentation tasks in NDVI analysis and plant health assessment. However, their adaptability to varying environmental conditions is limited, and they are prone to noise and variations in lighting conditions. Manual adjustment of thresholds may be necessary to achieve optimal segmentation results.

Feature Extraction and Classification:

Feature extraction and classification methods involve the extraction of discriminative features from images followed by the classification of these features into predefined categories. These methods offer flexibility in handling complex image structures and enable the classification of plant diseases and insect issues based on the extracted features. However, they heavily rely on high-quality training data for accurate classification. The computational complexity increases with the size of the feature space, and they are sensitive to variations in imaging conditions and image quality.

Deep Learning Architectures:

Deep learning architectures, particularly convolutional neural networks (CNNs), have revolutionized image processing tasks by automatically learning intricate patterns from data. They demonstrate high adaptability to diverse image characteristics and have the potential to accurately detect plant diseases and insect issues based on learned features. However, the requirement of substantial computational resources for training and inference poses a significant challenge. Deep learning models demand extensive labeled datasets for effective training, and the fine-tuning of parameters and model architectures can be complex and time-consuming.

In the table below we will analyze the differences in the three topologies and their tradeoffs.

Table 3.2.3-1. Image Processing Technologies Comparison

Technology	Pros	Cons
Thresholding	Simple and computationally efficient	Limited adaptability to varying environmental conditions
	Straightforward implementation	Prone to noise and variations in lighting conditions
	Suitable for binary segmentation tasks	May require manual adjustment of thresholds
Feature Extraction &	Ability to extract discriminative	Reliance on high-quality training

Classification	features	data for accurate classification
	Flexibility in handling complex image structures	Computational complexity increases with the size of feature space
	Enables classification of plant diseases	Sensitivity to variations in imaging conditions and image quality
Deep Learning Architectures	Capability to learn intricate patterns	Requirement of substantial computational resources for training
	Adaptability to diverse image characteristics	Need for extensive labeled datasets for effective training
	High potential for accurate disease detection	Complexity in fine-tuning parameters and model architectures

In our system we chose to separate the tasks of NDVI processing, plant identification, and the detection of disease/insect issues. We selected thresholding techniques not only to process the image for NDVI but also to visualize the areas of active photosynthesis within the plant for the end user. For plant identification, we chose to leverage deep learning CNNs to discern both the plant species and potential diseases or insect issues affecting the plant. Detecting disease and insect issues being a stretch goal for this project.

3.2.4 Image Processing libraries

Below is a comparison between image processing libraries that support thresholding techniques for NDVI and Convolutional Neural Networks (CNNs). The necessary technology, discussed in our technology research section (3.1.2), for classifying plant types, plant diseases, and insect issues with the plants our device will be monitoring. Below in table 3.2.4-1 is a comparison of three widely adopted libraries and their features that we have selected to accomplish the required tasks.

Table 3.2.4-1. Image Processing Libraries Comparison

Measure	OpenCV	TensorFlow	PyTorch
Thresholding Support	Yes	No	No
Convolutional Neural Networks	No	Yes	Yes
Ease of Use	High	Moderate	Moderate
Community Support	High	High	High
Performance	Moderate	High	High
Flexibility	Moderate	High	High
Documentation	Extensive	Comprehensive	Comprehensive
Integration with Other Tools	Good	Excellent	Excellent

Thresholding Support:

OpenCV: Supports thresholding techniques for image processing, including NDVI calculation.

TensorFlow and PyTorch: Do not natively support thresholding techniques but can be integrated with OpenCV for such tasks.

Convolutional Neural Networks (CNNs):

OpenCV: Does not provide built-in support for CNNs.

TensorFlow and PyTorch: Offer robust support for building and training CNN models for image classification tasks, including plant type classification and disease/insect detection.

Ease of Use:

OpenCV: Known for its simplicity and ease of use, making it suitable for beginners.

TensorFlow and PyTorch: Slightly more complex due to their deep learning frameworks, requiring some familiarity with neural network concepts.

Community Support:

OpenCV, TensorFlow, and PyTorch: Enjoy extensive community support with active forums, documentation, and tutorials available online.

Performance:

OpenCV: Offers moderate performance suitable for real-time image processing tasks.

TensorFlow and PyTorch: Provide high-performance computing capabilities, especially when leveraging GPU acceleration for deep learning tasks.

Flexibility:

OpenCV: Provides moderate flexibility with a wide range of image processing functions and algorithms.

TensorFlow and PyTorch: Offer high flexibility, allowing customization of deep learning models for various image analysis tasks.

Documentation:

OpenCV: Features extensive documentation covering a wide range of image processing techniques and functions.

TensorFlow and PyTorch: Offer comprehensive documentation with tutorials and examples for deep learning tasks.

Integration with Other Tools:

OpenCV: Integrates well with other libraries and tools for image processing and computer vision tasks.

TensorFlow and PyTorch: Also integrate seamlessly with OpenCV and other libraries, providing flexibility in building complex image processing pipelines.

Based on the comparison, the choice between these libraries depends on the specific requirements of the project. Because thresholding techniques for NDVI calculation are crucial, OpenCV would be the preferred choice. However, for the deep learning-based classification tasks, such as plant type classification and disease/insect detection, TensorFlow or PyTorch would be more suitable, providing robust support for building and training CNN models. In our project we decided on TensorFlow due to prior experience with the library.

3.2.5 Web Application and API

The selection of appropriate web application technologies plays a pivotal role in ensuring an enhanced user experience and seamless interaction with monitoring devices and APIs. In comparing industry-standard web application technologies, we consider mobile app integration and the ability to retrieve data and recommendations from APIs.

React.js:

React.js, a renowned JavaScript library, is widely acclaimed for its component-based architecture and efficient rendering capabilities. With React, developers can craft dynamic and interactive UIs for plant health monitoring applications. Reacts virtual DOM enables swift updates and smooth user experiences, while its ecosystem includes Redux for state management and Axios for API integration. Furthermore, React Native facilitates seamless mobile app development, ensuring uniform experiences across web and mobile platforms.

Angular:

Angular, a robust web application framework maintained by Google, provides a comprehensive toolkit for building scalable and feature-rich applications. Angular's declarative templates, dependency injection, and two-way data binding simplify the development of intricate plant health monitoring systems. Leveraging Angular Material, developers can fashion responsive and visually appealing UI components. Angular's Ionic framework simplifies mobile app integration, enabling developers to construct native-quality mobile apps using web technologies.

Flask:

Flask, a lightweight and versatile web framework for Python, is renowned for its simplicity, extensibility, and rapid development capabilities. Flask's minimalist design and flexible architecture make it ideal for crafting APIs and web applications for plant health monitoring. Flask seamlessly integrates with popular libraries and extensions, such as Flask-RESTful for building robust APIs. While Flask does not offer native mobile app integration out of the box, it supports RESTful APIs, facilitating seamless communication between web and mobile applications.

Table 3.2.5-1. Web Technology Comparison

Technology	User Experience	Mobile Integration	API Support	Scalability	Ease of Use
React.js	Dynamic UI, Fast Rendering	React Native for native mobile apps	Axios for API integration	Highly Scalable	Moderate
Angular	Feature-rich, Two-way Data	Ionic for cross-platform	HttpClient for API	Highly	Moderate

	Binding	mobile apps	integration	Scalable	
Flask	Simplified Development, Versatility	Supports RESTful APIs for mobile integration	Flask-RESTful for API development	Highly Scalable	Beginner-friendly

The selection of web application technology for our plant health monitoring system depends on various factors, including user experience, mobile integration, API support, scalability, and ease of use. React.js offers dynamic UI with React Native for mobile apps. Angular provides a feature-rich framework with Ionic for cross-platform mobile apps. Flask, known for its simplicity and versatility, supports RESTful APIs, making it suitable for building a web application for our use case. It is imperative to comprehend the strengths and weaknesses of each technology to make informed decisions for our application.

4 - STANDARDS AND DESIGN CONSTRAINTS

4.1 RELEVANT STANDARDS

4.1.1 IEEE 802.11-2020:

IEEE 802.11-2022 is a standard for wireless local area networks (WLANs) also known as Wi-Fi. The standard also includes enhancements to existing medium access control (MAC) and physical layer (PHY). The purpose of the standard is to provide wireless network operations between many portable/remote devices. The standard redefines an assumption in the design in the wired LAN system that an address is equivalent to a physical location. This is not the case for wireless networks. The standard defines the addressable unit as a station (STA). These stations are further broken down into fixed STA, portable STA, and mobile STA. A portable STA is one that is moved from location to location but used in a fixed location. Mobile STAs can access the LAN while in motion. The interaction with other network layers requires the standard to operate within the logical link layer. This means that the handling of the STA mobility happens within the MAC sublayer. This is why the 802.11 standard modifies the defined MAC standard to accommodate the needs of wireless operation. The standard in cooperation with 802.1X standards also provides functions to protect Data frames, authentication, and key management.

The architecture laid out in 802.11 consists of several components that interact to provide a WLAN that supports STA mobility transparently to upper layers. The standard describes a basic service set (BSS) understood as the coverage area within which member STAs of the BSS can remain connected and communicate with connected members. STA membership in a BSS is dynamic. STAs can turn off, turn on, exceed the range limitations, and return within range. To become a member of a BSS, a STA must perform a synchronization procedure. To access all the services provided by the BSS, a STA becomes “associated.” This is a dynamic process and involves the use of distributed system service (DSS).

Distributed system (DS) is the architecture component used to connect infrastructure BSSs. The DS and extended service set (ESS) are mechanisms for expanding connectivity for non-general link (non-GLK) operations. The DS enables mobile device support, providing the necessary logical services to handle address to destination mapping. An access point (AP) is a device that has STA functionality as well as access to the DS. The DS and BSSs allow the standard to create a wireless network of arbitrary size and complexity and refers to this network as the ESS.

The key concept of the ESS is that it appears the same to an LLC layer as an independent BSS (IBSS). STAs within an ESS can communicate as well as allowing mobile STAs to move between BSSs (within the same ESS) transparently. This is what allows devices to connect to one access point and roam into range of another access point and the user sees no difference.

WLAN radio measurements allow STAs to assess their radio environment, observing radio link performance and gathering data. STAs can conduct these measurements locally, request them from other STAs, or be asked to perform them. This data informs STA management and upper protocol layers for various applications and adjustments to optimize STA operations in their environment.

The radio measurement service enhances WLANs by offering standardized measurements across vendors and sharing this data with upper communication stack layers. Certain applications like VoIP, video over IP, location-based services, and those needing to mitigate challenging radio environments benefit from quantifiable radio measurements. These measurements help address issues in using unlicensed radio spectrum for emerging technologies.

For mobility-focused technologies like VoIP and video streaming, specific radio measurements like Channel Load and Neighbor requests help gather transition information. Utilizing this information enables faster handoffs between BSSs within the same ESS, aiding STAs in making informed decisions, whether in APs or non-AP STAs.

Wireless network management (WNM) allows STAs to share data to enhance the wireless network's performance. Through WNM protocols, STAs exchange operational information, keeping each aware of network conditions and network topology. These protocols also help STAs identify collocated interference and adjust RF parameters in response to network conditions.

This allows devices such as our plant monitoring system the ability to securely connect to a local network wirelessly. It also allows multiple devices to be in operation within a BSS range or in operational terms a Wi-Fi router. These devices can not only communicate with each other but also remote server systems by means of connected internet services. This is the primary means by which we will be collecting plant health data from our devices to be processed and delivered back to the end user.

In our project we are using a pre-certified Wi-Fi module to ensure we meet the IEEE 802.11-2020 standard. This ensures proper tuning of RF signals to meet the 2.4Ghz and 5Ghz spectrum allocations with the appropriate channels. The module allows the Wi-Fi communication for our project to be abstracted and exposed through the manufacturers drivers. This also means we are not required to go through the certification process ourselves or register our device with the FCC.

4.1.2 IEEE 802.15.1-2002:

IEEE 802.15.1-2002 is a standard for MAC control and PHY specifications for wireless personal area networks (WPANs). The high-level description of this standard is that this is the IEEE integration of the Bluetooth specification into the IEEE 802 group of standards. Portions of this standard are unaltered Bluetooth specifications, and this standard was produced at the request of and with a license from the Bluetooth special interest group. This standard's sections on the PHY layer, baseband specifications, link manager protocol, and logical link control, are all derived primarily from the core Bluetooth specification. The main differences are the removal of language about specific implementations as well as the introduction of specific information as to how this specification communicates with other IEEE 802 specifications.

PAN architecture refers to networks that deal with the immediate area around a group of personal devices owned by a particular user. This contrasts with a LAN which associates a group of devices within a geographic area which typically is not as personal and further reaching. WPAN refers to the untethering of a PAN connection. WPAN can be viewed as a sort of communications "bubble" around a particular user that allows multiple of their devices to communicate with each other or for each device to communicate with a different user's device within this bubble [11]. The main differentiations between WPAN and WLAN are network lifespan, medium control, and power levels/coverage. WPANs have a lifespan tied to the master of the network connection whereas WLANs do not have any specified lifespan as their existence is separate from the devices that utilize them. WPANs do require the same level of controllability over the network that WLANs do since they do not require to maintain an information base or exist as part of a larger network. WPANs trade the longer coverage of WLANs for lower power usage typically only having around 10m of coverage but also only having around 1mW of transmit power [11].

The PHY architecture of the standard refers to the RF receiver and transmitter requirements and test conditions. As per regulatory requirements Bluetooth devices operate in the 2.4GHz ISM band. This standard also gives requirements for the transceiver including RF tolerance, spurious emission, sensitivity, interference performance, out of band blocking, and signal strength requirements. These requirements are to be tested under the prescribed conditions which ensure that they are met under standard use-case conditions by the end user. Finally, the section on PHY architecture lists the required transmit frequency regulations for this standard.

The baseband architecture of this specification describes how the communication band functions. The communication band allows for point-to-point connections (one master to one slave) or point to multipoint connections (one master to multiple slaves). Additionally, this communication framework allows for a "scatternet" which links together multiple networks with different masters. Each network can only have one master but a master of one network can participate as a slave of another network. Within these networks there are two main types of links: ACL links and SCO links. SCO is a symmetric point-to-point link between one master and one slave which performs time critical operations like voice. ACL is a point-to-multipoint link for irregular traffic between a master and one or more slaves [11].

Overall, standard IEEE 802.15.1-2002 allows for essential local communication between personal mobile devices. For our project this means that we can ensure communication between our camera package with the camera and UV sensor and the sensor packages with the peripheral temperature, humidity, and moisture sensors. This allows us to create a distributed network of interconnected devices which can then collect data at the main MCU for processing and then transmission to a web database.

4.1.3 IEEE 1562-2021:

IEEE 1562-2007 is a standard for sizing batteries and solar arrays in stand-alone photovoltaic systems. Provides a standardized approach for sizing of solar panels and batteries. This allows users to properly size their solar array and backup battery to ensure the load receives constant power without significant interruption when solar power is the sole power source in the system. For our project we can use this standard to calculate the minimum size required for our solar panel and backup-battery. Additionally, the standard includes methods to account for solar downtime and methods to calculate load demands.

By ensuring compliance with this standard, we can guarantee that our camera package can always have power. This standard will be essential in the design of the main power subsystem on the camera package.

4.1.4 IEEE 1149.1-2013:

IEEE 1149.1-2013 is a standard for boundary-scan architecture on ICs more commonly known as JTAG. JTAG is a system built into ICs that allows for testing of interconnects as well as flashing of system memory [45]. JTAG ensures that users will be able to program and test their PCBs regardless of the specific packages or architectures that they select. JTAG presents a standardized option that will allow us to properly test and prototype the ICs that we select for our design. JTAG will also allow us to maintain our finalized design and detect any errors that could occur as the result of trauma (exposure to dust, water, high, or low temperatures etc.) post-production while testing the finalized design.

Additionally, JTAG is one of the two serial download boot modes supported by the ESP32 series of chips. This means that by designing a USB serial port we can allow the ESP32 to be programmed and powered via USB which will give us more options when testing and finalizing our system.

4.1.5 IEEE/ASHRAE 1635-2022:

IEEE/ASHRAE 1635-2022 is a standard for the thermal management of batteries in stationary applications. The guide includes standards for the management of the thermals of stationary Li-ion batteries. This will be relevant to our project since the solar panel will use a Li-ion backup battery and will face a large amount of heat build-up during hot days. Li-ion batteries can become dangerous when exposed to excess heat so management of heat in a system using Li-ion batteries is a safety concern. Specifically, Li-ion batteries can become flammable when their heat exceeds certain values. While most of the standard focuses on the creation of HVAC systems to regulate the thermals of the various battery technologies, the standard does provide relevant guidance to us on the expected effect on capacity and lifespan for Li-ion batteries. The standard also provides basic principles on the correct housing to use to ensure contaminants do not adversely affect the heat performance of the battery.

This standard will provide essential guidance for our finalize design since the thermal regulation of the camera package will be extremely important to the system's overall functionality. As such we will design the housing of our camera package with the principals and understandings gained from this standard. While like we mentioned above, not everything in this standard is directly related to our design, the guidance and design philosophy shown in this standard will be vital to our device's thermal performance.

4.1.6 IEEE 1625-2004:

IEEE 1625-2004 is a standard for rechargeable Li-ion or Li-ion polymer batteries for portable computing. The standard includes requirements for manufacturers to ensure that the developed Li-ion batteries function within the set specifications and have the required safety considerations. This includes ensuring that there is proper vent activation during overcharge events as well as proper thickness of the separator to ensure no internal shorts occur. This standard also includes guidelines to follow when designing "host devices" which are devices that will charge or discharge a Li-ion or Li-ion polymer battery [63]. Within these guidelines are the proper ways to implement an over voltage protection and an over current protection system. It also lays out how to properly set up battery detection for the charge control sub-system. Finally, the standard provides some guidance to end users as to the functionality of the various protection systems required by the standard.

Most of the protections for our Li-ion battery will be taken care of by the charge controller. However, this standard will allow us to make use of the error and fault data that the charge controller provides and make decisions as to the viability and safety of continuing to use the battery after a potential upset is detected. Additionally, by complying with this standard we can ensure that the finalized design is safer for end users.

4.1.7 ISO/IEC 9899:2018 (C18):

ISO/IEC 9899 is the international standard for the C programming language and ISO/IEC 9899:2018 is the most current revision of the C standard known as C18. The ISO/IEC 9899:2018 international standard aims to specify the proper form and establishes the interpretation of programs written in the C language. The ISO/IEC 9899:2018 standard document contains four major subdivisions: Preliminary elements, the characteristics of environments that translate and execute C programs, language syntax constraints and semantics, and library facilities. These sections together aim to specify: the representation of C within specific applications, the proper syntax and constraints of the language, the semantic rules of interpreting code written in the language, the representation of the data to be processed by the programming language in both input and output, and the restrictions and limits by utilizing the C language in specific programs.

Considering that the C programming language is the most widely used language for Embedded systems design and implementation, these standards are highly important for the scope of this project as well as many others on the market. The ISO/IEC 9899:2018 standard described above in short, ensures that the code written is portable to other systems should it be required, but also allows for various compilers to understand and compile the same code as they all follow the C standard as specified. It is worth noting that some specific functionalities are left ambiguous in the C standard to allow for individual compilers to decide how to run a program most efficiently.

There are only 4 major revisions of the ISO/IEC 9899 international standard: ISO/IEC 9899:1990 (C89), ISO/IEC 9899:1999 (C99), ISO/IEC 9899:2011 (C11), and ISO/IEC 9899:2018 (C18), which makes this standard rather stable a far as changes go, however it is important to note that standards do undergo revisions periodically which introduce new features or remove old ones.

4.1.8 ISO/IEC 60529

ISO/IEC 60529 is the international standard that concerns ingress protection or IP ratings for enclosures. Ingress protection refers to the ability for some enclosure or casing to protect the enclosed electronics from the ingress of water or solid foreign objects. These ratings come on a scale from 0 to 8 for water and 0 to 6 for solid foreign objects [62]. The standard includes the testing procedures required to verify an enclosure for each rating.

For our design we will use this standard to ensure that the enclosure we use for the camera package and sensor package will be sufficient to protect the enclosed electronics from the elements. To ensure our compliance with this standard we will ensure that the products we use or design to enclose our electronics fit this standard. This standard will also serve as a guide when considering the products or enclosures we consider in our research phase which will save us time that would otherwise be spent verifying these products. This standard will also give us a methodology to test and verify our own housings if we decide to use those instead of consumer grade housings.

4.1.9 IEEE 946-2020

IEEE 946-2020 is the IEEE standard relating to the design of DC power systems for stationary applications. The focus of this standard does primarily relate to high voltage systems delivering a much larger amount of power than our system will deal with. However, this standard provides guidance on the proper protections needed for DC power systems including indicators for over voltage and over current events. It also provides guidance for the location of these protections to provide the best response to these faults and make servicing them easier.

For our design we will not use this standard exactly as the power systems it is designed for are much higher power than our system. However, we will use the philosophy of this standard to implement our own fault detection and monitoring protections on our power system. For our design the charge controller will monitor the status of the power system on the camera package and will provide information to the MCU for protections to be implemented. Additionally, we will use an LED to monitor the status of this power system through the pins on the charge controller.

4.2 DESIGN CONSTRAINTS

4.2.1 Time Constraints

Time is a serious concern for this project in a variety of ways. We are limited by the deadlines set for documentation, prototyping, and the finalization of the design. To ensure that we can keep these constraints from affecting our ability to deliver a complete project we must ensure that our research and design has a level of complexity that is reasonable to complete within the deadline. This means ensuring that the selection of components ensures that the implementation is possible in a timely manner.

Additionally, a major concern for prototyping and manufacturing is the availability of components. The components selected need to be obtainable within a reasonable timeframe so that they do not delay the completion of key deadlines such as prototyping and testing. This includes dev boards related components not used in the final design that aid in testing. An adjacent issue to this is the manufacturing time for the PCB designs. The completed PCBs will also need to be tested and verified so the time to manufacture and assemble the PCBs must be reasonable.

Another time constraint placed upon our design is the limitation of the format of senior design. Our final design must be presentable in the final 10-minute meeting held at the end of senior design two. This means that any systems we design must be demonstrably functional within this time. That means fully comprehensive demonstrations of certain subsystems like the power subsystem are mostly impossible with the demonstration being more focused on the system being functional and not the system functioning to our exact specifications in terms of efficiency, battery charge time, battery life etc. This means when creating our final design, we need to take steps to ensure that we can properly demonstrate its features in our final presentation.

Finally, one of the biggest time constraints is the deadline set for the completion of the semester 1 document. This deadline is the point at which we need to have a finalized design that is ready to be manufactured and assembled for implementation and testing. This deadline is a constant pressure that limits the amount of research we can do into different implementations, components, and software frameworks.

To mitigate the effect of time on our project we are paying constant attention to the amount of time dedicated to each task. We have planned time set aside for unforeseen issues or errors in the design that will result in time losses. We have also made sure to regularly meet to discuss our usage of time and how to best utilize it moving forwards. These practices should mitigate the effect of time and ensure that our design is of as high a standard as possible while still being completed within a reasonable timeframe. Ultimately, time, like cost, is one of the major constraints that every project must deal with. Time constraints will continue to make the other constraints in this section worse and worse. There is no complete removal of time as a constraint only a mitigation of its effect.

4.2.2 Cost Constraints

Another major constraint on the flexibility of our design is cost. Cost here refers to both the budget we have set aside for prototyping parts and dev boards as well as the final cost of the design. Cost is one of the most important constraints as it creates and worsens other constraints. For instance, if budget and price were not an issue then safety would have little impact on any actual decision regarding the dimensions or sizing of our design.

The final cost of the design is critical because our project aims to create a plant health monitor that is accessible to an average consumer. To do so the cost of the finalized design cannot be prohibitively expensive. To achieve this goal, we have a self-imposed cost target of 100$ for the finalized design to attempt to ensure that the final design is not too expensive. This target is not a hard target as the price of comparable products tends to be quite high (in some cases upwards of $10,000) but rather a target decided upon when looking at other "enthusiast level" products in the gardening space. However, while this does make our target somewhat flexible the design should still cost as little as possible while maintaining a high degree of accuracy to ensure that there is a place for it on the market.

The budget constraint is another self-imposed target of 300$. We only have a limited amount of money to put towards components, development boards, PCBs, and additional parts for potential issues that arise. This constrains the types of components we can select as well as the quality of components that we can select. This will also limit the number of design iterations we are able to create if there are some unforeseen issues with the initial design. Finally, the budget constraint will limit our options for PCB manufacturers which can impact the amount of time it takes to have a design manufactured.

There is no way to fully remove the effect of cost on the final design. The ways we are seeking to mitigate its effect is by allocating a proper amount of time into research and development to ensure that the components we select, and the design of our subsystems remains as high-quality as possible while still falling within our self-imposed budget and cost constraints. This necessarily will compete with all the other constraints related to our project such as safety and time. Additionally, we may have to make certain tradeoffs like a larger buck regulator circuit footprint because of a cheaper inductor to hit our cost target. Ultimately, cost is a never-ending balancing act which amplifies the effect of every other constraint.

4.2.3 Safety Constraints

Our design is meant to operate outdoors and withstand inclement weather. When exposure to water is a concern with electronics safety becomes an issue. Both the power used to supply the main board and sensor package with power are potential safety concerns when water is introduced. To ensure that the Li–ion battery and the Lithium battery used for the main system and sensor package respectively do not cause harm we must ensure proper isolation of these components. This limits our options for which enclosure we can choose as well as the acceptable size of the final design so that we do not go over budget.

The only way to mitigate this safety constraint is by ensuring that the enclosure of the electronics is properly sealed. We must put time into ensuring the enclosures we order contain the proper IPX ratings and will independently test the final device to ensure that it is safe. Additionally, we will implement protections through the charge controller to ensure that any system upsets that may be caused by exposure to water will not result in catastrophic failure of the power system.

We must also make sure that our power sub-system is properly designed to not allow for a thermal run-away of the Li-ion battery. This will involve providing the proper protections for an overcurrent or overvoltage event as well as building in thermal protections directly to the design of the power sub-system on the main PCB. Currently our design does include a thermal monitoring thermistor on the camera package but as we test and advance our design into the next semester, we will take additional efforts to ensure the safety of our device.

4.2.4 Environmental Constraints

Like with any design that will remain outdoors for an extended period our design will be subject to environmental constraints. These constraints generally take four different forms. The first form is the personal environmental needs of the end user of our design and the second form is the environmental constraints placed on the design by the laws and procedures of the cities, municipalities, and countries where it will be used, the third form is the heat produced ambiently by the environment that the system operates in, and the final form is the constraint on the power delivered to our device via the solar panel.

In terms of the first aspect our design must be sure to not cause any unintended harm to any plants it is meant to be monitoring. This harm could occur in a variety of ways. Some of the most common would be a leak of an alkaline battery into the soil of a plant causing its death. Other potential harms are the potential for the Li-ion battery to cause a fire which would damage the plants that the device is supposed to be monitoring. This will require that we select housings with a proper IPx grade when designing the final device.

In terms of the second aspect our design must not violate any regulations regarding proper isolation and containment of electrical circuitry. This is especially important for the two batteries and the higher voltage lines coming from the solar panel. Our device must also comply with any pollution or waste regulations for it to be legal in the districts where it is used. Any non-compliance with these regulations could be devastating for our final system as it would make it unable to be used. This makes this aspect of the constraint one of the most important.

The third aspect of this constraint is the heat ambiently generated by the environment. Our system relies on the solar panel backup battery and power regulators all being relatively close to each other to ensure optimal power performance. Additionally, our system is meant to be left outside during the day where solar radiation and heat are at their highest. All these factors cause concerns as to the thermal management of our finalized design. The components we have selected only have operating temperatures as high as 125 degrees Celsius. This means that we need to manage the heat generated by all our components to ensure that this value is on exceeded. Doing so will pose significant design constraints and will be a large consideration for our PCB layout.

The final aspect is the solar constraint. Our device's camera package relies heavily on the ability of the solar panel to deliver meaningful power. This is naturally constrained by the environment that it is used in. Certain locations get less solar energy over the course of a year and every location gets less solar input during certain periods of the year. Our design must take these aspects into account when creating the power supply for the camera package and we must select a backup battery that is large enough to supply the package when the solar panel is unable to produce the needed power.

4.3 RELEVANT SPECIFICATIONS

The following sections are not accredited standards. However, these specifications still prove vital to the design of our system and its ability to communicate. As such we felt it was important to include the information about these specifications and how we will be holding ourselves to them in our design.

4.3.1 UM10204

UM10204 is the specification for the I2C bus. This specification provides information for users who wish to use I2C communication in their system and provides them with the tools to ensure the proper functionality of their I2C bus. It provides information on the configuration of the I2C communication protocol including the most common modes (those being standard fast and fast plus). The specification provides guidance on the voltage level recommended for use on the voltage rails as well as the values required for the pull-up resistors. Specifically, the specification provides a methodology for determining the minimum and maximum size of the pull-up resistors used based on the bus capacitance and communication rate.

For our design we will use this specification to ensure that the sizing of the I2C pull-up resistors selected is acceptable. This will ensure that any faults will be avoided and the I2C bus on our system will function without error. This can also ensure that we chose the maximum size I2C pull-up resistors to avoid an increase in the leakage current.

4.3.2 USB

While there exists no accredited standard for USB there is an industry standard and a de-facto standard to USB architecture and communication. For our design we will specifically use USB for both power and communication. For power we will use a micro-USB port to power the device during testing and programming and for communication we will use a USB to UART bridge to communicate between our personal computers and the MCU. As such USB will play a vital role in our design. As per USB specifications our design will support the proper 5V input from the USB port as well as the proper D+ and D- communication lines for USB to UART conversion. Compliance with this standard will allow us to program our MCU aftermarket and will allow us to test our finalized design separate of the tests of the power sub-system. This will allow for a more methodical approach to testing and can help us isolate and problems that may be caused by the design of the power sub-system.

5 - HARDWARE AND SOFTWARE DESIGN

5.1 HARDWARE DESIGN

5.1.1 MCU

The MCU is the brain of our design. The MCU is what will allow the camera and sensors to communicate with the web server as well as the primary control component for the sensor and power management subsystems. The MCU is the highest-level system in our design. For our design we will utilize two versions of the ESP32 series of chips. More precisely we will use the ESP32-S3-WROOM-1 for the camera package because of its PSRAM allowing it to store and manage the camera images from the Pi cam, and we will use the ESP32-C3-WROOM-02 for the sensor package due to its excellent low power operation and Bluetooth compatibility.

The MCU design section will also include the relevant peripheral components of the MCU design (UART to USB bridge, I2C, etc.) as these either directly relate to the MCU or its communication with other components. To reference the communication flow of the MCUs please view image 6.1.2-1 and for power flows for the MCUs please view images 6.1.3-1-1 and 5.1.3-2-1 for the camera package and sensor package respectively.

5.1.1-1 Camera Package

For the camera package our MCU of choice was the ESP32-S3-WROOM-1 as it has the PSRAM needed to store images. This MCU will communicate with the UV light sensor via I2C, with the web database via Wi-Fi and with the ESP32-C3 module via Bluetooth. The circuit schematic for this MCU can be seen below.

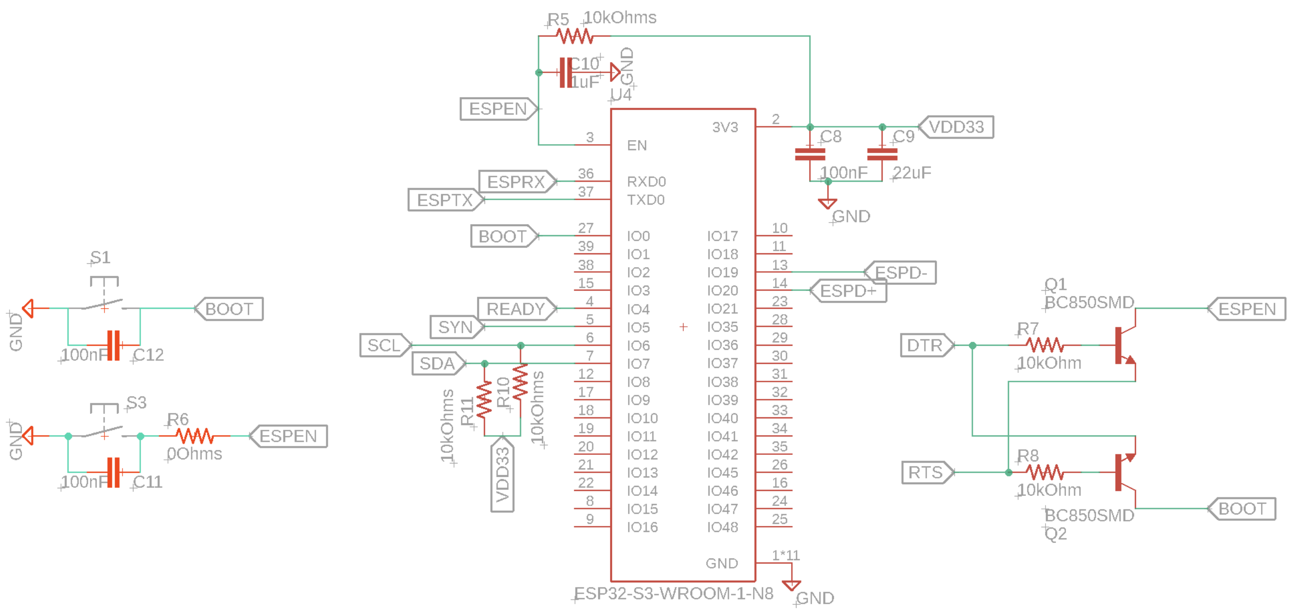


Figure 5.1.1-1-1. Camera Package MCU Circuit Diagram

As seen above the MCU will be powered via a 3.3V input from the on-board buck regulator. This is then connected to an RC delay circuit recommended by the ESP32 design team to reduce potential upsets on start up. The enable pin is also connected to a button which is tied to ground to allow for a manual reset to be triggered. Additionally, GPIO 0 is connected to a button which is also tied to ground to allow for manual selection of boot mode. GPIO 46 is specifically not used because a value of logical 1 during boot could potentially cause an upset and we do not have a need for the GPIO. The enable and GPIO 0 pins are also connected to the collector side of a pair of NPN BJTs whose base and enable sides are connected to the DTR and RTS signals of the UART to USB bridge. This will allow for automatic programing of the MCU.

The important GPIOs used for communication and data are GPIOs 4, 5, 6, and 7. GPIOs 4 and 5 will be used as digital outputs to communicate and program the UV light sensor's SYN and READY pins. They will play an essential role in the timing of readings from the UV light sensor. GPIOs 6 and 7 will be used as the I2C SCL and SDA pins respectively. 10k pull-up resistors were selected as they fall within the I2C specification for resistance values with our voltage level and line capacitance while also minimizing the current lost through the resistors when not in use. We will use the 3.3V output from the on-board buck regulator to serve as the pull-up value for our I2C bus.

The TXD0 and RXD0 pins are fed into the CP2102 UART to USB bridge for communication and programing via USB. The GPIO pins 19 and 20 labeled ESPD- and ESPD+ respectively are fed into another USB port for JTAG serial debugging. Any of the GPIOs not mentioned here or above are not needed for our design and will be left unconnected and will not be used in our design any strapping pins this applies to will be left at default values.

Much of this design is inspired by and with reference to the ESP32-S3-DevKitC-1 which is publicly available on the Espressif site [47].

We will now continue the discussion of the support peripherals to the main MCU with the circuit schematic for the UART to USB bridge. For this design we have chosen the CP2102 from Silicon Labs as this is the chip used in the ESP32-S3-DevKitC-1 so we can utilize that as a design reference, and we can have no concerns about any compatibility issues. Additionally, since our team has been using the ESP32-S3-DevKitC-1 for the prototyping we can ensure that the drivers needed for a UART to USB bridge IC are already installed and set up properly on our team's computers making integration of the new design seamless. This circuit schematic can be seen below.

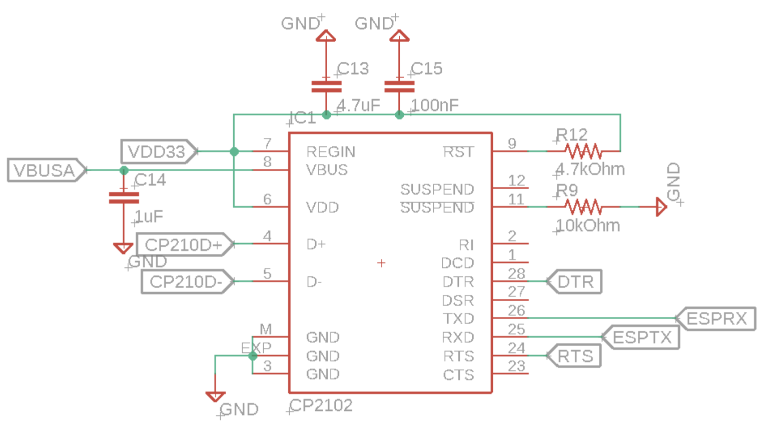


Figure 5.1.1-1-2. Camera Package UART Bridge Circuit Diagram

Above is the circuit schematic for the UART to USB bridge. The CP210 D+ and D- pins connect to the D+ and D- pins of the micro-USB connection, and the VBUSA pin connects to the VBUS pin of the micro-USB connection used for USB to UART. The 10kOhm resistor on the SUSPEND pin is recommended by the CP2102 chip designers to increase the noise resistance of the chip. The 4.7kOhm resistor is connected for a similar reason and since we do not need to disable the chip this pin is connected directly to VDD33. The DTR and RTS signals connect to the BJTs shown in the prior image with the MCU. These signals will indicate when data is being sent through the USB to program the MCU and will initialize a restart followed by a boot in joint download mode allowing for the MCU to be automatically programmed.

5.1.1-2 Sensor Package

The MCU on the sensor package has a much more minor role to play than on the camera package but it will still play an essential role in our design. The MCU on the sensor board will allow for the Bluetooth communication between the camera package and the sensor package and will allow for the control and communication of the sensors on the sensor package. To fill this purpose, we chose the ESP32-C3-WROOM-02. The circuit schematic for the implementation of this module can be seen below.

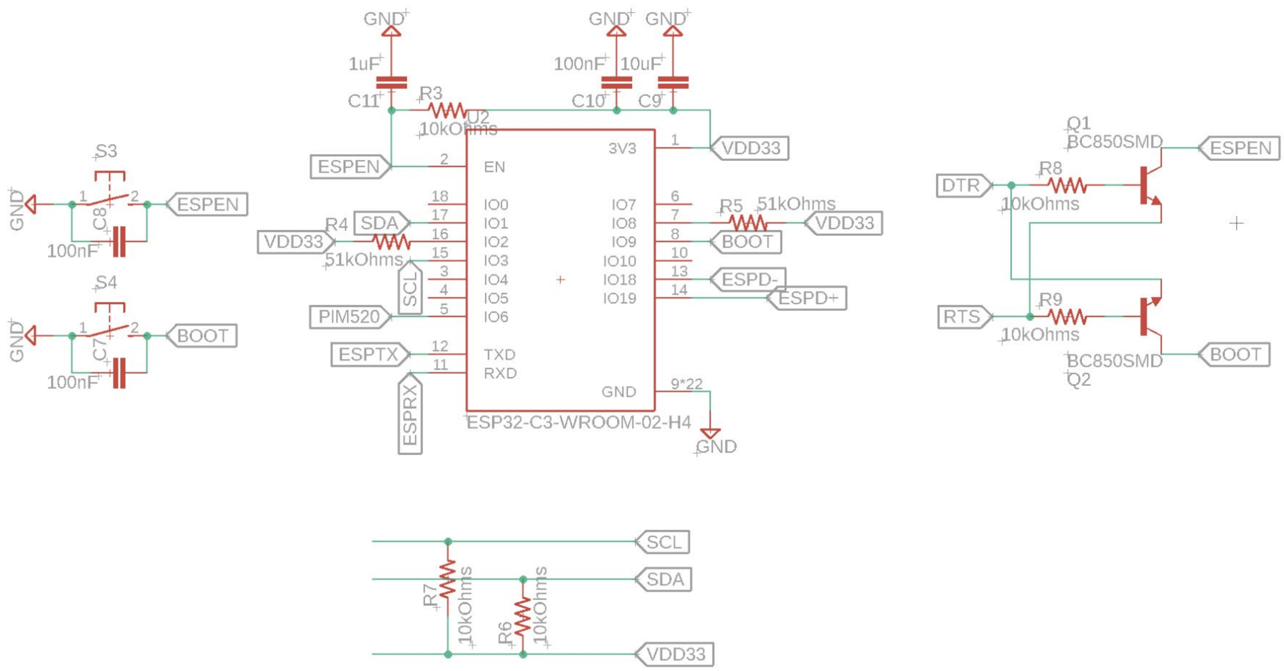


Figure 5.1.1-2-1. Sensor Package MCU Circuit Diagram

As seen above the enable pin is tied to an RC delay circuit to prevent potential upsets on boot caused by instability from VDD33. This pin is additionally connected to a switch which is tied to ground to allow for a manual reset. This switch and the switch connected to GPIO 9 both have a parallel capacitor to prevent potential EMI interference when pressed. These two pins are additionally connected to a pair of BJTs to allow for automatic programing of the module. GPIOs 2, 8, and 9 are the strapping pins for this module. As such GPIO 9 is connected to a button tied to ground to allow for manual boot mode selection. GPIOs 2 and 8 are connected to VDD33 through 51kOhm resistors to ensure that they are logical 1 on boot since other values could result in potential upsets or unneeded boot mode selections. The 51kOhm resistors were selected to provide enough current to ensure a logical 1 while preventing an excess of leakage current.

The GPIOs 1, 3, and 6 will be used for I2C and the PFM communication with the soil moisture sensor. The I2C bus was configured with 10kOhm resistors to minimize the leakage current when not in use. The I2C bus is one of the largest current draws on this board when the system is idle so maximizing the value of these resistors is vital to ensure that the low power operations can be completed properly. Using 10kOhm resistors does lock us into using standard speed I2C but that is perfectly acceptable since we do not need the higher I2C speeds. The ESPRX and ESPTX are used for the UART to USB bridge and the ESPD+ and ESPD- are used for the JTAG serial USB. The circuit diagram for the USB to UART bridge can be seen below.

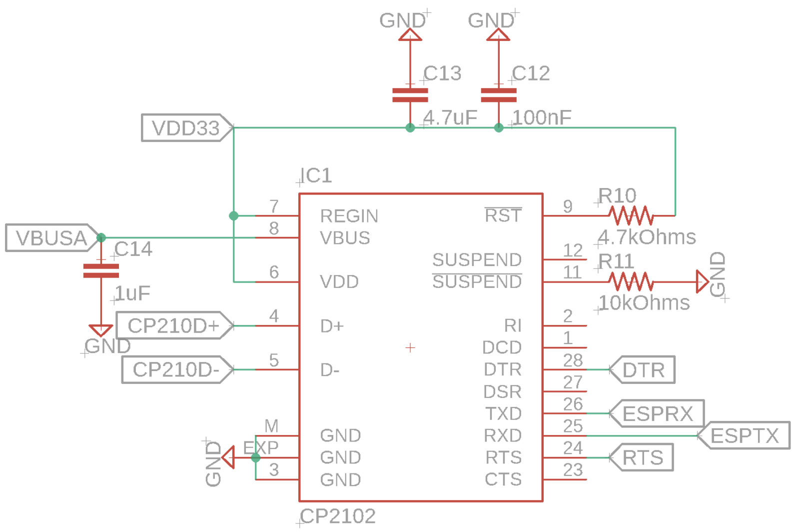


Figure 5.1.1-2-2. Sensor Package UART to USB Bridge Circuit Diagram

As mentioned in the camera package section the UART to USB bridge was designed to match the ESP32 dev kits as closely as possible [47]. The design for this circuit is identical to the circuit shown in section 6.1.1-1 and a more in-depth analysis of the design considerations for this bridge can be viewed there.

This concludes the MCU design for this project. As an additional note the camera and Raspberry Pi Zero 2 W are not included in the design section because the Pi camera is plug-and-play with the Pi Zero 2 and the only connection between the Pi Zero 2 and the other PCBs will be two wires connected to the camera package.

5.1.2 Sensors Subsystem

The sensors provide all the auxiliary support features to our final design. The IoT sensors will provide the user with a more comprehensive view of the state of their garden allowing them to make better use of the NDVI imaging that the camera generates. Generally speaking, the NDVI imaging provides a notification of an issue with a plant and the sensors provide a diagnosis of the particular issue.

For our system, the sensors are to be split up into two separate groups. The UV light sensor will be attached to the camera package due to the need for it to have good access to the sun to generate accurate readings. The temperature and humidity sensor and the ground moisture sensor are attached to a separate sensor package. This separation was done to allow the readings generated by these sensors to be specific to certain plants. Additionally, the ground moisture sensor needs to be on the ground close to the plant being monitored which would impede the functionality of the solar panel and UV light sensor. The control scheme and data flow for this layout can be seen in the block diagram below. Control blocks can be seen in blue, sensor blocks can be seen in green, and the web database can be seen in orange.

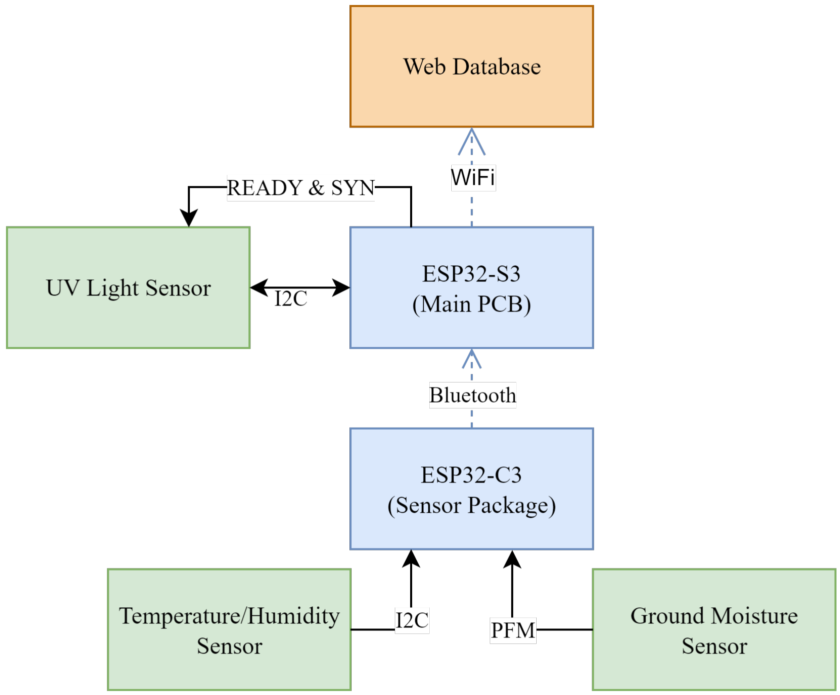


Figure 5.1.2-1. Sensor Package Block Diagram

As shown above the temperature and humidity sensor and the UV light sensor will be controlled by the MCU on its respective board and will transmit the data it obtains via I2C. The ground moisture sensor simply transmits a PFM signal over a signal digital GPIO to the on board MCU. Additionally, the UV light sensor requires two additional GPIOs from its MCU to control it. However, as previously mentioned the data from this sensor does use I2C for transmission. The data from the sensors on the sensor package will be sent to the camera package via Bluetooth before being sent to the web database with the UV light sensor data via Wi-Fi.

In the following sections we will not break up the sensors on the camera package PCB and the sensors on the sensor package PCB. This is because there is no meaningful impact on the design of the sensor circuits particularly. The differences between the PCBs mainly pertain to the power draws and MCU circuitry rather than the actual design of the sensor circuits. This is mostly because sensor circuits are fairly basic and hardware agnostic.

The first sensor we will discuss is the UV light sensor. This sensor will provide users with useful diagnostic information to help them understand what may be wrong with the plant they are monitoring. Its data is transmitted via I2C to the MCU before being sent to a web database. The circuit diagram for this sensor can be viewed below.

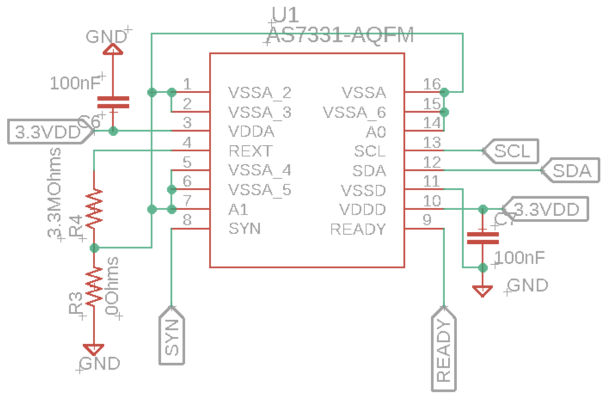


Figure 5.1.2-2. UV Light Sensor Circuit Diagram

The most important aspect to note with this design is the separation of the digital and analog grounds (VSSD & VSSA respectively) via the 0 Ohm resistor and via a physical separation on the finished PCB. This separation is meant to reduce the effect of noise and will carry over to the PCB layout as well. Similarly, the separate decoupling capacitors for the digital and analog inputs should help reduce the effect of noise on the system. The digital and analog power source will be the same since the buck regulator used has a low enough noise level to be below the tolerance of the IC. The SYN and READY tags on their respective pins will connect to digital GPIOs on the MCU. These pins are used to control the timing of the sensor readings as well as indicate when the sensor has completed its reading. They are also used to help configure the energy saving options of the sensor. The actual data gathered by the sensor is sent over the I2C bus using the SDA pin.

The next sensor we will discuss is the temperature and humidity sensor. This sensor is to be located close to the plants being monitored and provide even more peripheral diagnostic information on the potential problems a plant could be facing. This sensor should provide data that is particular to each plant being monitored. Similarly to the UV light sensor the data gathered by this sensor will also be transmitted over the I2C bus. The circuit diagram for this sensor can be viewed below.

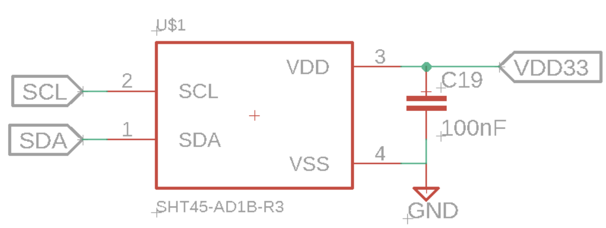


Figure 5.1.2-3. Temperature and Humidity Sensor Circuit Diagram

As you can see from the diagram the circuit for this sensor is incredibly simple. The IC does practically all the work and only needs one external decoupling capacitor on the power pin to function with stability. The commands to start reading the temperature/humidity, go into power off mode, go into idle, etc. are all sent through the SDA pin on the I2C bus. There is no other need for external inputs on this sensor.

The final sensor we will use in our design is the soil moisture sensor. This sensor fills a similar role as the temperature/humidity sensor in that it provides specific data about one plant to be used to diagnose issues. This sensor does not have a particular circuit diagram because the PIM520 is a fully realized break-out board with a capacitive moisture sensor integrated. Designing a capacitive moisture sensor from scratch was beyond the scope of what our group could realize within the time allotted to us. This sensor will simply be connected to the sensor package PCB via three wires. Two wires will supply 3.3V and ground and the third wire will be a digital GPIO for receiving the PFM data from the sensor.

This concludes the sensor subsystem design. Overall, these sensors will provide the user with a wealth of options to diagnose the issues with their plant in our all-in-one home gardening solution. For information about sensor power consumption please view table 3.1.12-1 and the following section on the power subsystem. For information on the software configuration of these sensors please view sections 5.2.2 and 5.2.6 and for information on the I2C configuration used for these sensors please view section 6.1.1 on the MCU design.

5.1.3 Power Subsystem

The power subsystem is a vitally important part of any design. While power tends to be the lowest level system with most of its design aspects decided by the requirements of higher-level systems, the power subsystem serves as the foundation upon which the rest of the design must lay.

For our design the power subsystem will consist of two different power supplies with two different power management systems. The first power supply will be the solar panel and backup battery on the camera package. This power supply will be managed by the main charge controller which will handle the MPPT for the solar panel and the charge of the backup battery. This will then flow into a 5V buck regulator whose output will then be fed into another buck converter which will regulate the 3.3V output required for the MCU and IoT sensors. The second power supply will be a 9V battery located on the sensor package. This power supply will simply be managed by one buck converter to regulate its output to 3.3V which will supply all the components of the sensor package.

5.1.3-1 Camera Package

The camera package’s power subsystem entails a significant level of complexity because it relies on a solar source. The solar panel will serve as the primary input during times of peak solar output, a supplementary source during times of lower solar input, and will be off during times of no solar input. The backup battery will act as the supplementary power source during times of lower solar input and the primary source during times of no solar input. This will be managed by a charge controller which has the MPPT algorithm needed for good solar efficiency already implemented. The output of this controller will then feed into a 5V buck regulator which will supply the 5V rail for the Pi Zero 2 Finally, the supply to the ESP32-S3 and UV light sensor will be regulated by a 3.3V buck regulator. The power flow for this system can be seen in the figure below.

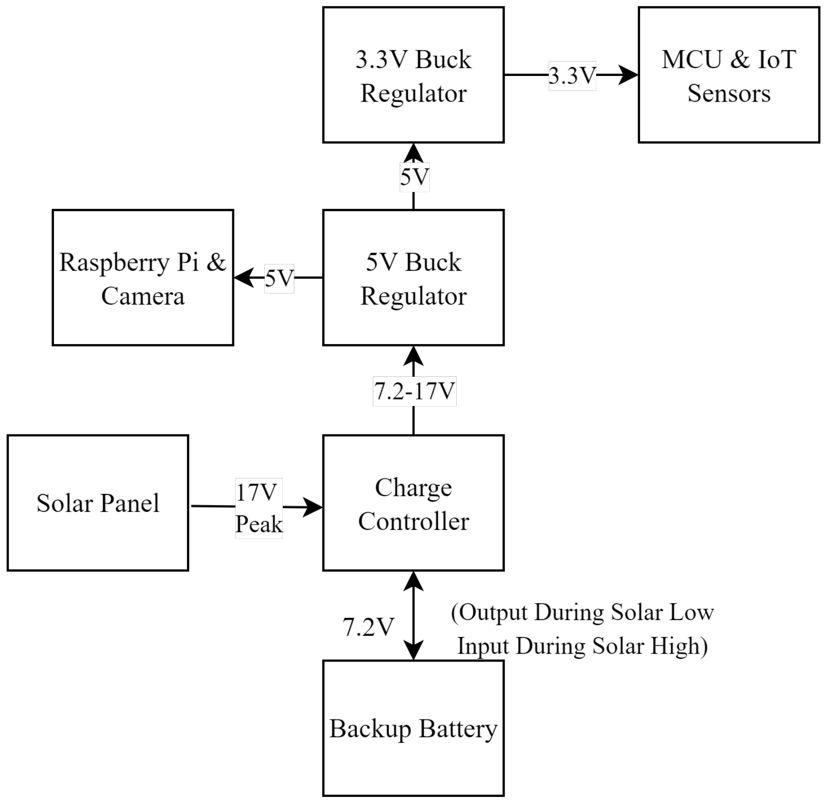


Figure 5.1.3-1-1. Camera Package Power Flow

The above power flow shows the required voltage level for each group of components on the camera package. The MCU & IoT sensors block refers to the ESP32-S3 WROOM U2 and the UV light sensor on the main PCB. The Pi Zero 2 W board has its own internal power management system to regulate the voltages being fed directly to the Pi cam. To make use of this system the 5V input power simply needs to flow into GPIO pins 2 or 4 with pin 6 connected to ground. For a more detailed analysis on the current consumption of each component please reference table 3.1.12-1 in chapter 3. The 5V power to the Pi Zero 2 W will connect through two wires (one at 5V and the other at ground). The battery will connect to the PCB through a pair of soldered wires connected to the battery input pin of the charge controller and to ground respectively. The solar panel will connect to the PCB through a two wires.

The 5V output to supply the Pi Zero 2 W and to be further regulated by the buck regulator will be supplied by the output of the charge controller. The controller will act as a buck regulator for this conversion. This controller is the “brains” of our power system for the camera package. The controller will allow the system to quickly switch between solar and backup power in a very short amount of time while also providing a MPPT interface with the solar panel. Finally, this controller will also monitor the backup battery and control its charge cycle, charge current, and charge voltage. The backup battery it is charging consists of two 18650 Li-ion battery cells connected in series and the solar panel driving it will be a 6V 5W solar panel. The circuit configuration for this controller can be viewed below.

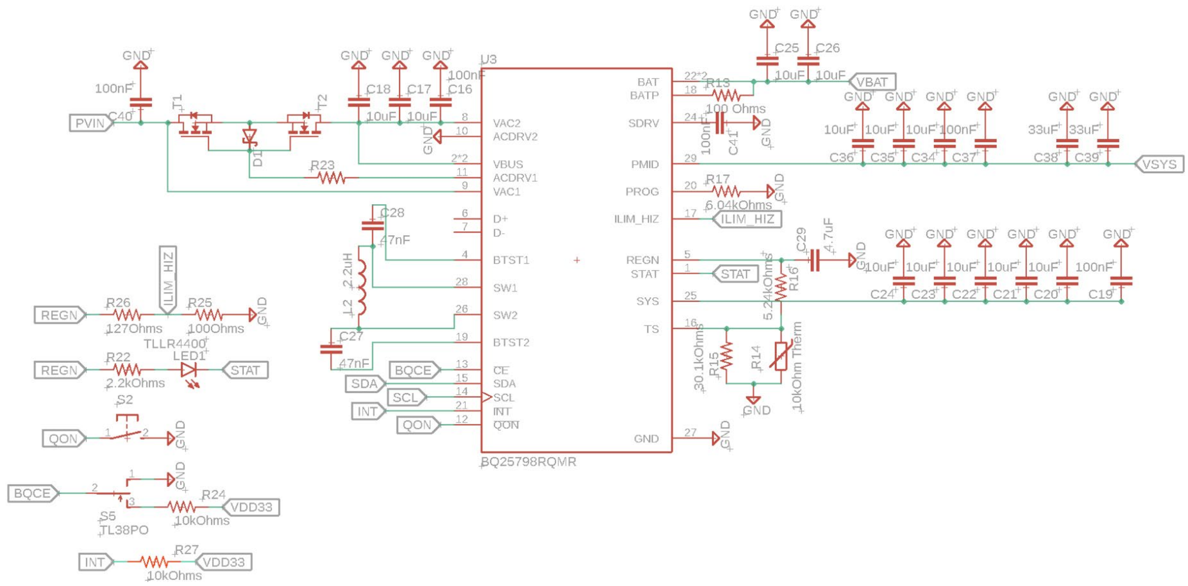


Figure 5.1.3-1-2. Sensor Package MCU Circuit Diagram

This is the most complex circuit being used and so we will split up the design into a few sections. We will begin with the input/output circuitry. The output capacitors on the SYS and PMID pins and the battery capacitors on BAT provide a large enough bank of power to temporarily prevent load power upsets in the time it takes to switch from main PV power to backup mode power as well as smooth out any EMI upsets. When this switch occurs the two n-channel MOSFETS connected to the VAC1 and VBUS pins get disabled, and power is unable to flow through the VBUS. Instead, the internal BATFET is enabled, and the battery voltage begins to discharge through PMID to the system load. When a larger voltage than battery voltage is applied on PVIN the charger will switch back to the active mode and will be supplied from the PV panel.

Next, we will discuss the protection circuitry. The REGN pin is the internal LDO which provides approximately 4.6-5V depending on the VBUS voltage [57]. This will drive the ILIM_HIZ voltage divider. The voltage sensed on this pin by the internal ADC will set the input current limit to prevent an overcurrent event. For our applications this voltage divider will result in around a 1.5-1.2A current limit which should be beyond what our PV panel is capable of outputting. The STAT pin provides an indication of the current state of the charger based on its value. This will be read via the attached LED with a low value (LED on) meaning charging is in progress, a high value (LED off) meaning charging is complete, and a 1Hz blink meaning there is a fault detected [57]. Finally, the TS pin connects to the internal ADC and will sense the temperature via the voltage seen on the pin. The attached thermistor is a negative temperature coefficient thermistor meaning at high temperatures the voltage on the TS pin will drop and the on-chip current will be reduced or in extreme circumstances disabled altogether.

Finally, we will discuss the control circuitry. The QON pin will only act as a system power reset for our operation since we decided to leave off the SHIP FET. When S2 is pressed, and the pin is driven to ground for the predetermined amount of time the chip will perform a power reset. The CE pin is the charge enable pin and will be left tied to ground via the connected switch for most of our operations since it is active low. However, when the switch is flipped, and the CE pin is pulled up to the 3.3V supply the charger will disable which can allow for troubleshooting. The INT pin is pulled high to the 3.3V power rail and will go mostly unused for our operation. The SDA and SCL lines will connect to the MCU and will allow for the control of the charger via I2C and will also be used to check the readings of the ADC to allow for monitoring of the charging process as well as potential diagnosis of faults. Finally, the PROG pin is tied to a 6.04kOhm resistor because the higher switching speed was needed to cut down on footprint and we will be utilizing a 2-cell battery pack.

The output of this charge controller will then be fed into a buck regulator with high efficiency capable of supporting the entire 7.2-17V output range of the charger. The output of the charger should fall significantly below 17V for most of the time. However, on startup before an initial MPPT calibration there is a small window where a full unregulated 17V may be sent to this regulator. The regulator chosen to complete this task is the LM64460-Q1 regulator. The realization for the regulator can be seen below.

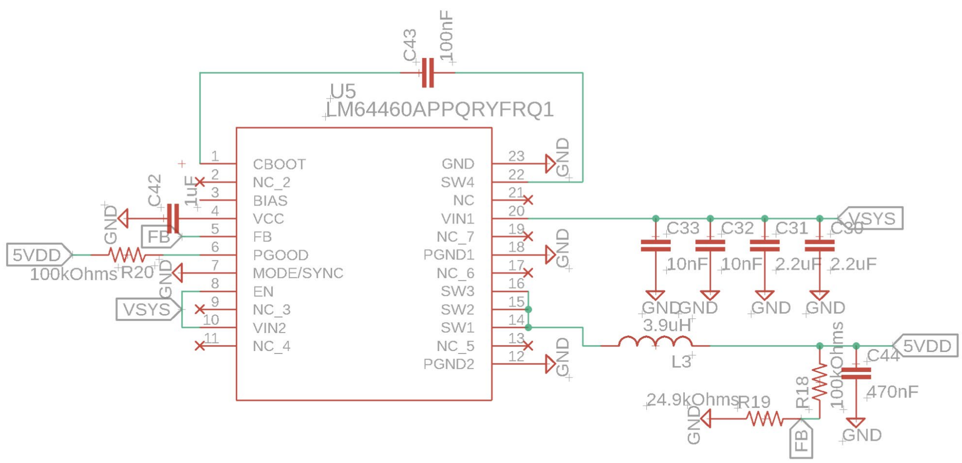


Figure 5.1.3-1-3. 5V Out Wide-Range Buck Regulator

The realization is similar to the other buck regulators we selected, and the design was created with the help of TI's WEBENCH tool. The footprint was able to be reduced to an impressive 90 square millimeters thanks to the high inductor resistance tolerance of this IC. The efficiency will be around 95.2% at peak operation but when the load current decreases due to the MCU and Pi Zero 2 W going into shutdown or idle the efficiency will dip to around 88-90% [64]. The BOM for this circuit will also be slightly more expensive at around $2.59. We do not require a UVLO circuit for this regulator or any other regulators in our design since any under-voltage event should be handled by the charge controller without the need for the buck regulators to lock-out. Similarly, the EN pin is tied to the 5V rail since we have no need to ever turn this regulator off.

The 3.3V power line for the IoT sensor and the MCU on the camera package will be provided by a 3.3V buck regulator. This regulator will get its input from the 5V output of the charge controller. The buck regulator we selected was the TPS62932. This is a TI part meaning we can utilize their powerful design tool WEBENCH to help with the design of the circuit. We used the data sheet to generate the initial circuit and validated it using WEBENCH [65]. The circuit generated can be viewed below.

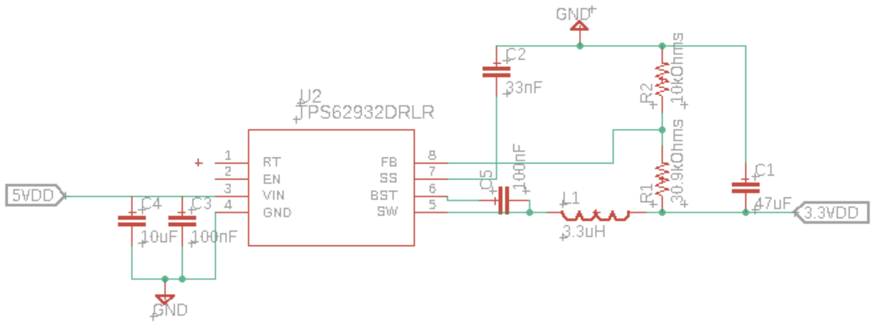


Figure 5.1.3-1-4. 3.3V Out Camera Package Buck Regulator Circuit Diagram

The above image is the circuit for the buck regulator on the camera package. The EN pin was left floating because we do not require a low voltage UVLO circuit for this implementation and as such the regulator can be left enabled indefinitely (which is done by leaving EN floating). The RT pin is left floating because the lower switching frequency of 500KHz is more efficient and for our design the higher efficiency takes priority over the lower footprint offered by a higher switching frequency. The 5VDD tag refers to the 5V output of the charge controller. The estimated footprint for this circuit is 491 millimeters squared and the estimated BOM cost is $0.88, and the estimated efficiency is around 94.6% [9]. The 5VDD label refers to the 5V input from the charge controller and the 3.3VDD label refers to the 3.3V supply for the MCU and UV light sensor. This buck regulator serves as the final stage in the power management subsystem for the camera package. Once the input power flows through this regulator all voltage levels are accounted for.

5.1.3-2 Sensor Package

The sensor package only requires a single output voltage level for all its components. The power supply for the sensor package will be a single 9V lithium battery with around 1.2 Ah of charge. There will be no other power source for this board. Additionally, there will only be a single regulator as there is only one required voltage level for this board. The power flow can be seen in the diagram below.

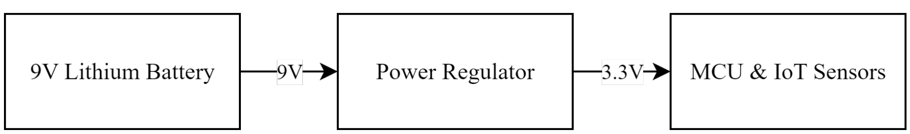


Figure 5.1.3-2-1. Sensor Package Power Flow

To regulate this output the TPS62932 buck converter was selected. The schematic for the buck converter circuit was generated with the help of Texas Instruments (TI) WEBENCH as well as the manufacturer datasheet for this part. The components were sized according to TI's manufacturer data sheet and verified with their WEBENCH tool [65]. The schematic for this circuit is shown below.

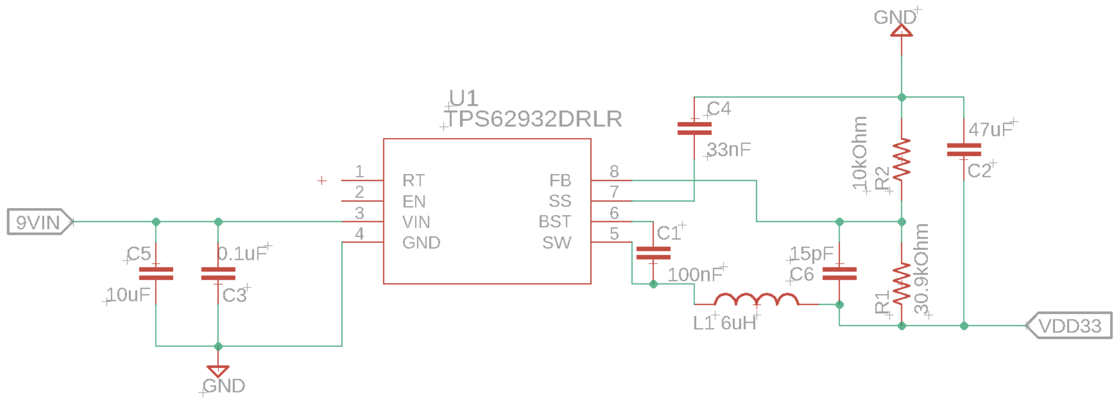


Figure 5.1.3-2-2. Sensor Package 3.3v Buck Regulator Circuit Diagram

Above is the circuit diagram for the 3.3V buck regulator on the sensor package. The EN pin was left floating because we do not require a low voltage UVLO circuit for this implementation and as such the regulator can be left enabled indefinitely (which is done by leaving EN floating). The RT pin is left floating because the lower switching frequency of 500KHz is more efficient and for our design the higher efficiency takes priority over the lower footprint offered by a higher switching frequency. The 9VIN tag refers to the 9V battery used as the power source for this board. For this design the estimated footprint is 493 square millimeters and the estimated BOM cost is around $0.90, and the estimated efficiency is around 95% [9]. This is the extent of the power footprint for the sensor package. The 9V battery will be external and connected via a snap connector soldered to the sensor package PCB. Since we only require the single voltage level for all the components this circuit will be the entirety of the power subsystem on the sensor package.

5.1.4 Final Schematics

Below are the final, full schematics for the camera package and the sensor package in their entirety. These schematics have been provided to give a big-picture view of the finalized design and to provide a concrete reference for chapter 7 when reviewing the PCB layouts in that section

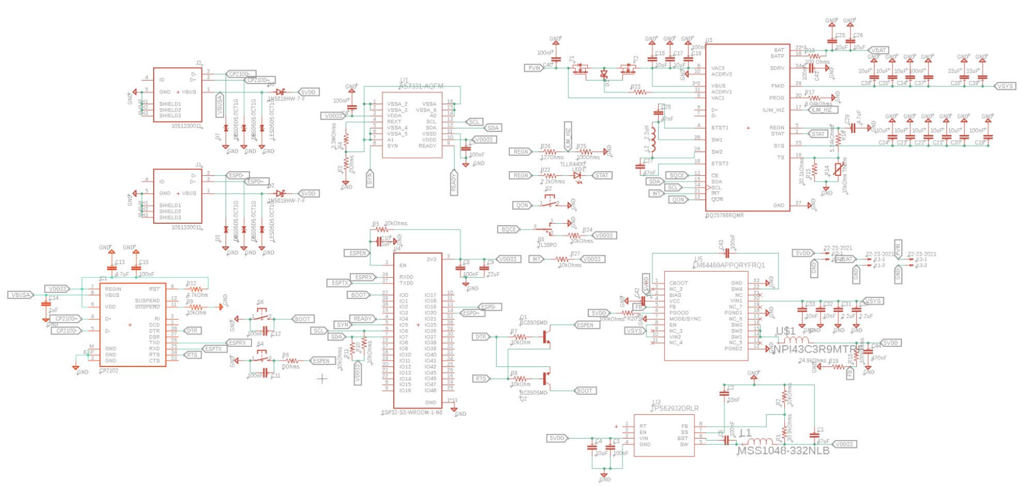


Figure 5.1.4-1. Camera Package Full Schematic

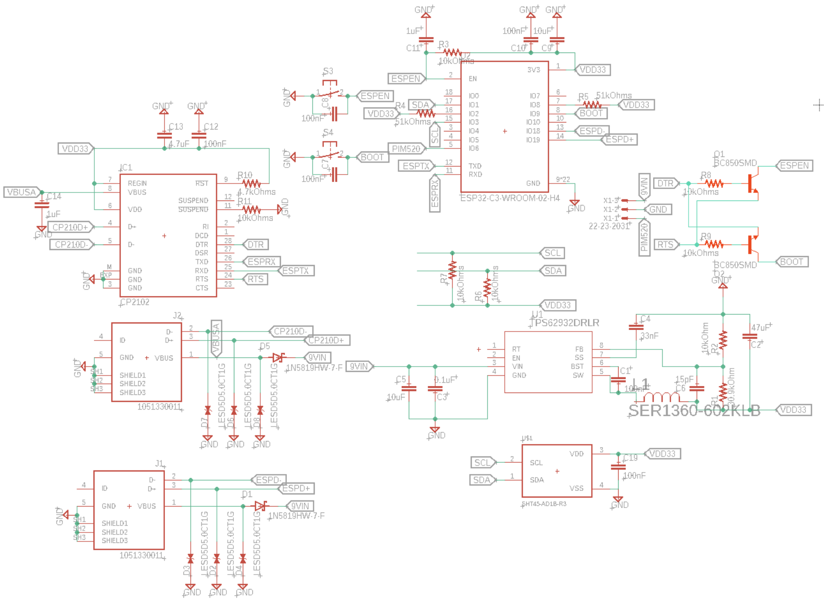


Figure 5.1.4-2. Sensor Package Full Schematic

As seen above the full schematics for both boards are largely the same as the subsystem schematics show in their respective sections. The only component left out of the discussion of the various subsystems was the schematics for the USB housing. This is simply because the design of these housings is generally the same regardless of the part chosen. Our design is again referring to the ESP32 dev-kits and is just the shield and ground pins tied to ground and each of the input pins tied to ground through a surge protector. Similarly, the 9V inputs are connected through a Schottky diode to prevent any surge. Unlike the layouts, however, these schematics are mostly set in stone. The only major potential changes are the addition of 0 Ohm resistors on the power outputs or LEDs on the power outputs to aid in debugging the power subsystems or the addition of a couple extra pin headers to communicate with the Pi Zero 2 W when it is powered down.

5.2 SOFTWARE DESIGN

5.2.1 Use Case Diagram

An overview of how our devices and software interface together and with the user is depicted in Figure 6.2.1-1. The user interacts with a web app on a mobile phone, registering an account via our web server. Then the user registers a camera device that acts as a central hub for several sensor devices to connect to via Bluetooth low energy (BLE). Once the camera device is registered and connected to home wi-fi the camera will go through setup by asking to be placed in front of the plant/s, the camera device will try to verify it can see the plant correctly and identify it. Then additional sensor devices can be connected to the camera device and assigned to visible plants.

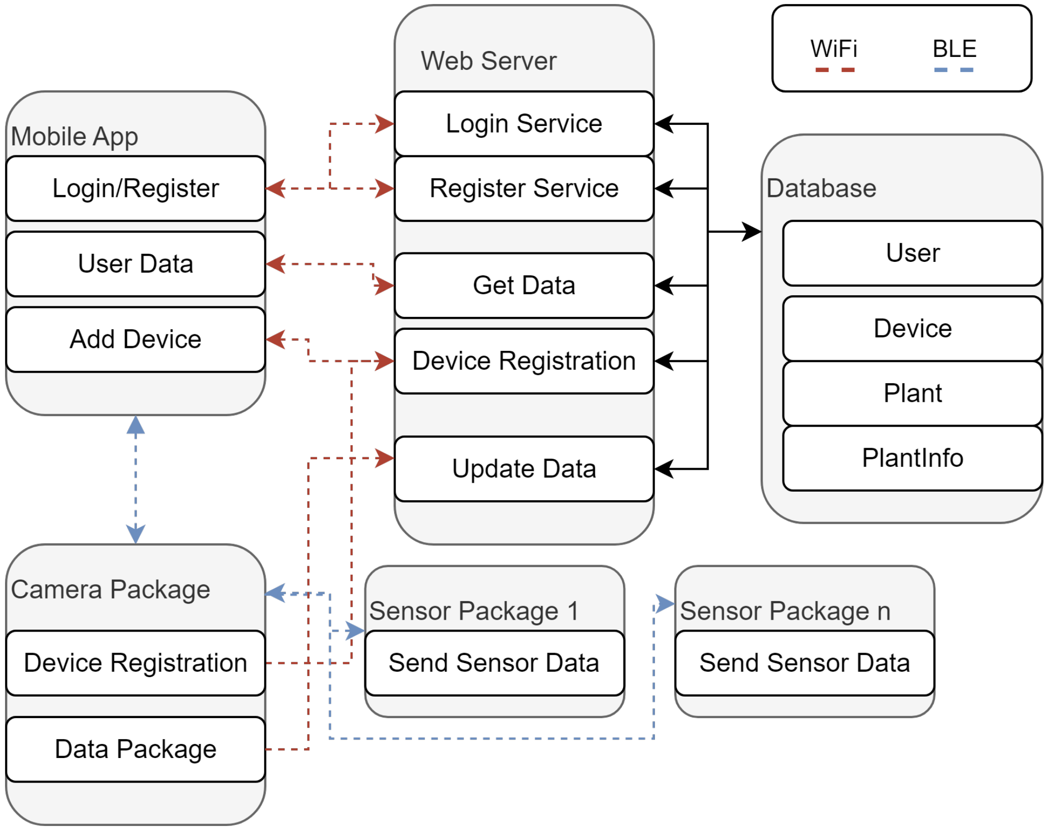


Figure 5.2.1-1. Use Case Diagram

5.2.2 Camera Package Software Flow

The camera package software flows from boot to scheduled monitoring of identified plants through the procedure described in Figure 6.2.2-1. On initial boot the device (camera package) checks if it has been registered and if it has not it waits at activation until activation has been completed then reboots the device into its normal operating mode. A schedule is defined by the web server based on user location and seasonal light and temp conditions for optimal image capture. On a scheduled event the camera checks to see if it is an image event and wakes up the camera system if true, if not proceeds to external sensor data. Once an image is taken the image is verified and processed with the thresholding system to produce the NDVI data and with our CNN to identify potential disease or insect problems. The device then packages the data and sends it to our web server via Wi-fi. Once the data has been sent successfully the device goes back to low power mode waiting for the next event.

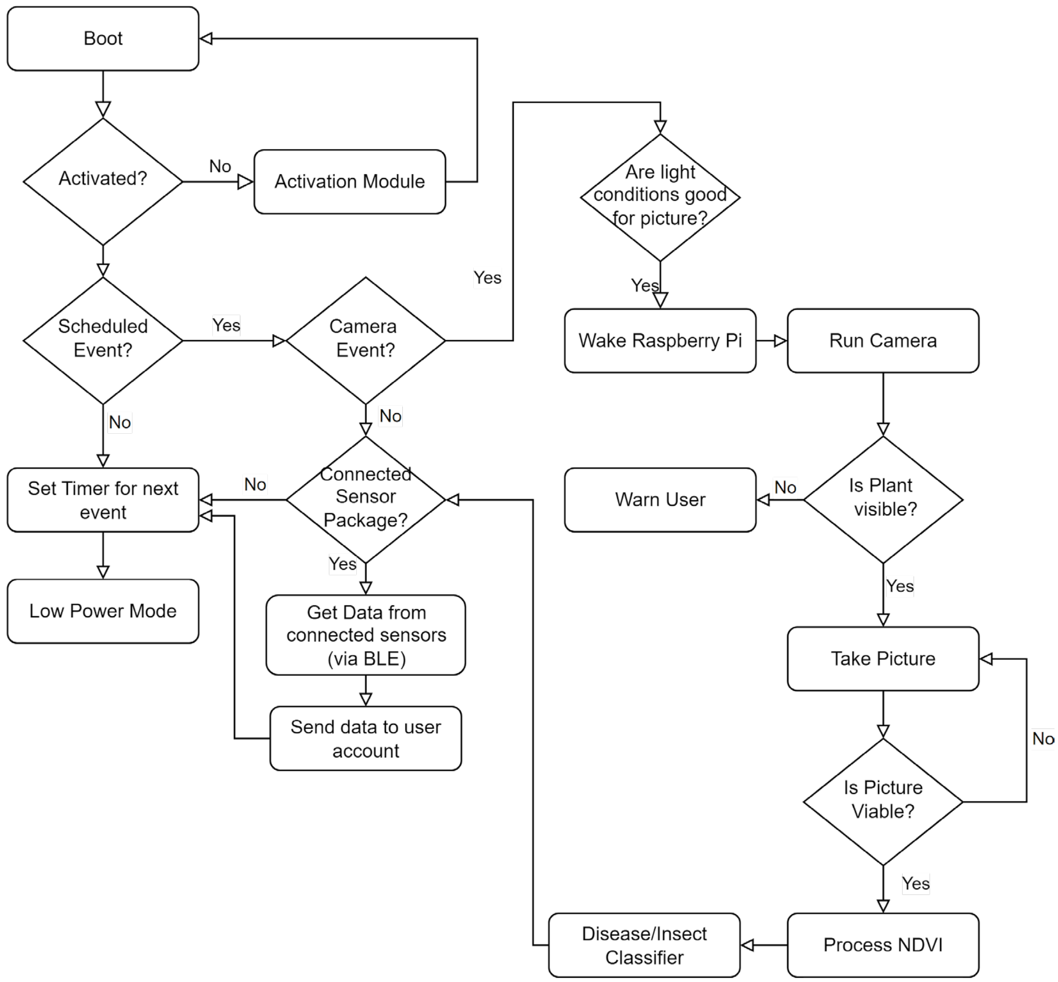


Figure 5.2.2-1. Camera Package Software Flow Diagram

5.2.3 State Diagram

The camera package has four states. Starting in the idle state the camera package is in a low power mode using the real time clock (RTC) module, the device keeps time until a scheduled event is triggered. One event triggers the device transitions to the next state based on the event type. A camera event initiates a state change to image mode. In image mode the UV sensor is read to judge the light conditions and if appropriate light is present the raspberry pi and attached camera module wakes up. Once the camera is ready an image is taken, it is verified, processed, and sent to the ESP32 device. Once the data is gathered the state moves to sensor mode and the raspberry pi is shutdown. In sensor mode the MCU connects to the associated sensor packages and sensor data is sent to the camera package. Once the data from each of the associated sensor packages is received and prepared with the image data to send to the web server. The MCU then changes state to send data mode. In send data mode the MCU wakes up the Wi-Fi module and connects to the Wi-Fi network it is associated with. Once connected the device attempts to send the data package to the web server. Once the data is sent successfully the camera package returns to idle state. If the event is a sensor event the camera state is skipped, and the process continues from sensor mode forward. Each state attempts to progress successfully but on failure re-attempts are made until time occurs. The Figure 6.2.3-1 illustrates these states and their progression.

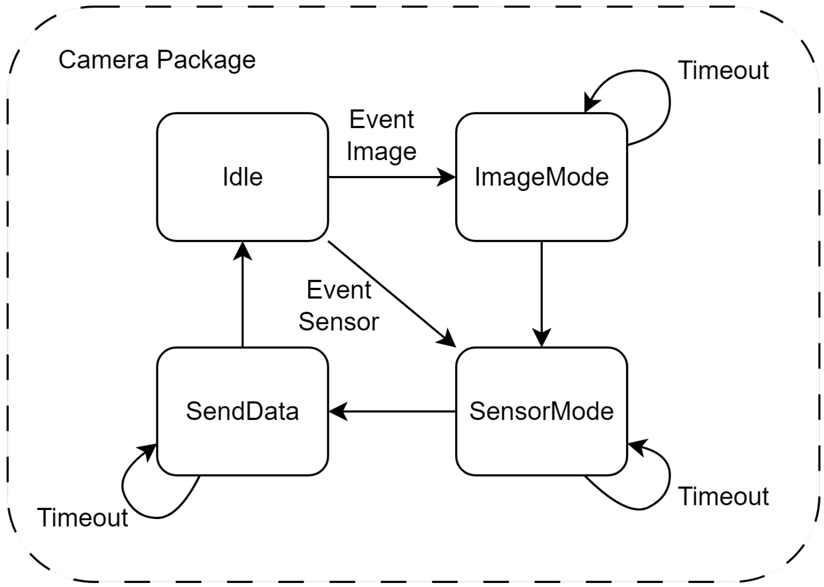


Figure 5.2.3-1. Camera Package State Diagram

5.2.4 Data Structure

Each device will store a config file that will store device configurations assigned in the activation process. Things such as Wi-Fi network SSID and credentials, connected Bluetooth device MAC addresses, device id, plants associated, and zone will be saved in this file.

During operation the sensor package during the day the sensor readings are added to a list mapped to date and time and sent via BLE to the camera package. During evening to early morning hours, the sensor data is packaged as a map of date/time with a list as value and sent as a batch on the first camera event. The data is deleted once the data is confirmed to be sent.

The camera package only temporarily stores data as it prepares image data, potentially identified plant disease or insect issues. Sensor data received is parsed and packaged into JSON format in preparation of sending to the web API. Once the image data is processed, the NDVI data is added to the JSON data along with any potentially identified diseases/insect issues. The data is then sent to the web APIi and on confirmation the data was received the data is discarded on the device.

The web server will store all the sensor and image data in a document data model using MongoDB. Each date/time reading will be associated with a plant entity that in turn is associated with a user. When a user logs in to the web or mobile app the server fetches most current data to display to the user for each plant. As a stretch goal, a knowledge base of recommendations based on plant type, the history of sensor readings and NDVI data, and any identified disease/insect issues are stored as a decision tree in the database.

5.2.5 User Interface

The user will interface with the monitoring system primarily through a web/mobile app. Creating an account, activating devices on that account, and placing the devices will all be done on a mobile device. The application will walk a user through adding a device, beginning with a camera package. Upon device startup (if not activated) the device enters activation mode, where the device waits to be activated via the mobile application. The user connects to the camera package via BLE where the user is prompted to name the device and assign it to a zone. Once assigned to a zone the application asks the user which Wi-fi network the device should connect to and its associated credentials. The information is sent to the camera package and stored in a config file, then attempts to connect to the Wi-Fi network.

Once the device is connected to the Wi-Fi network the device drops the BLE connection and begins to stream images to the application for placement confirmation. The application attempts to verify good placement by identifying a plant in the image and the type of plant. Feedback is given to the user to optimize the placement of the camera and if it cannot identify all plants correctly the user is prompted to input the plant or plants type. Continuing the setup process the application will ask the user if they would like to add a sensor package to the configured plants.

Setting up a sensor package is similar but attempts to identify inactivated sensor packages in BLE range. Once it finds a sensor package ready for activation the application asks the user if they would like to add this device. If the user acknowledges that this is a device, they would like to add the camera package then attempts to connect via BLE. If the connection is successful, the application informs the user of the successful connection and asks the user to assign a plant to monitor. The camera package stores information about the sensor and associates it by the sensor's Bluetooth MAC address. The user can continue this process for as many sensor packages as they need for the plants associated with the connected camera package.

After registering all the devices, the user is directed to a monitoring dashboard where they can see all the plants being monitored. By clicking/touching a plant picture the specific plants details page is shown to the user. This details page shows the latest readings of the plant including a full color image, the NDVI heatmap image, the UV light conditions, the temperature, the humidity, and the ground soil moisture level. A histogram of the sensor data is also shown to give the user a full picture of the conditions and status of their plant. The initial concept for our user interface can be seen below in figure 5.2.5-1.

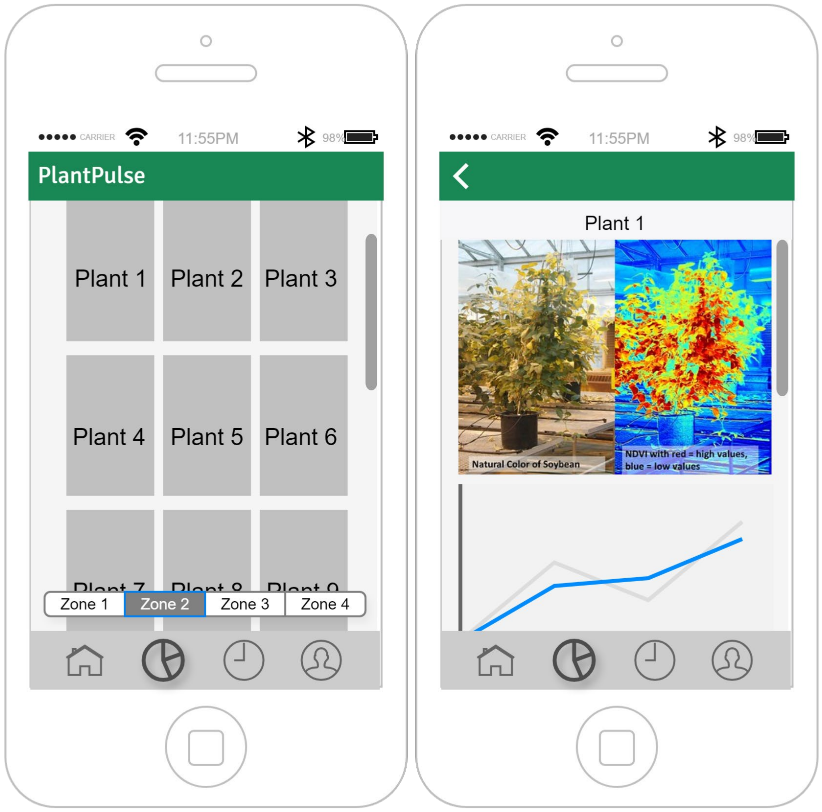


Figure 5.2.5-1. User Interface Mockup

<u>5.2.6 Sensor Package Software Flow:</u>

For the sensor package there were three sensors included the soil moisture sensor, the temperature sensor, and the humidity sensor. The ultraviolet light sensor is not necessarily included in the sensor package as it is included as part of the camera package since it is required to determine when the camera is going to conduct a reading, but it will be included in the sensor software flow as it is a part of the overall sensor system of this plant monitoring system.

The software flow for the entire sensor system begins with the system boot, from here there is a check to see if the system is activated. If the system is not activated it returns to boot. If the system is activated there will be a timer interrupt acting as a scheduler which will be running to indicate whether we will gather any sensor data. Since there are two separate microcontroller units working together for this system it is required to have two different interrupt setups.

The first one will run locally on the ESP32-C3 microcontroller unit which handles the sensor data and will trigger an interrupt reading the sensors relatively frequently. The idea here will be a longer interrupt for the soil moisture sensor as the sensor does not need to be continuously making readings with soil moisture being unlikely to change a lot throughout a given day. The soil moisture sensor is also a capacitive type of sensor, which is more resistant to corrosion but not completely so having the sensor powered as infrequently as possible is beneficial for the longevity of our system. The humidity sensor and the temperature sensor both are going to run on shorter interrupts as these specifications will fluctuate more throughout the day and require more frequent transmission to the user. Regardless of the type of sensor if the conditions are for the nighttime hours of the day the interrupt will be much longer with the conditions most likely not fluctuating at all and these readings will be cached and sent in a batch during the first day time sensor reading to conserve power. When the sensor data is collected it will be transmitted via Bluetooth to the ESP32-S3 unit and sent to the server via wi-fi.

The second interrupt setup will be on the ESP32-S3 microcontroller unit which will signal when we need to take an ultraviolet light sensor reading. This interrupt will only run a few times during the day and since it is directly related to the camera taking its reading it will be with the camera in the camera package. When this interrupt occurs the ESP32-S3 unit will send a communication via Bluetooth to the ESP32-C3 microcontroller unit handling the sensor package to also take an overall sensor reading for all the current conditions to be sent via wi-fi to the server for the user to see. This interrupt will not run at all during the night as the conditions for the camera to run will never be met when the sun is not up. This means the first day reading containing the batched overnight sensor data will also be a scheduled camera event.

After each successful transmission of data whether it is a current data transmission or a caching of the data collected, we will then put the sensors in a low power state to conserve as much power as possible and all the battery to charge via the integrated solar panel system. The flowchart for this system can be seen below in figure 5.2.6-1.

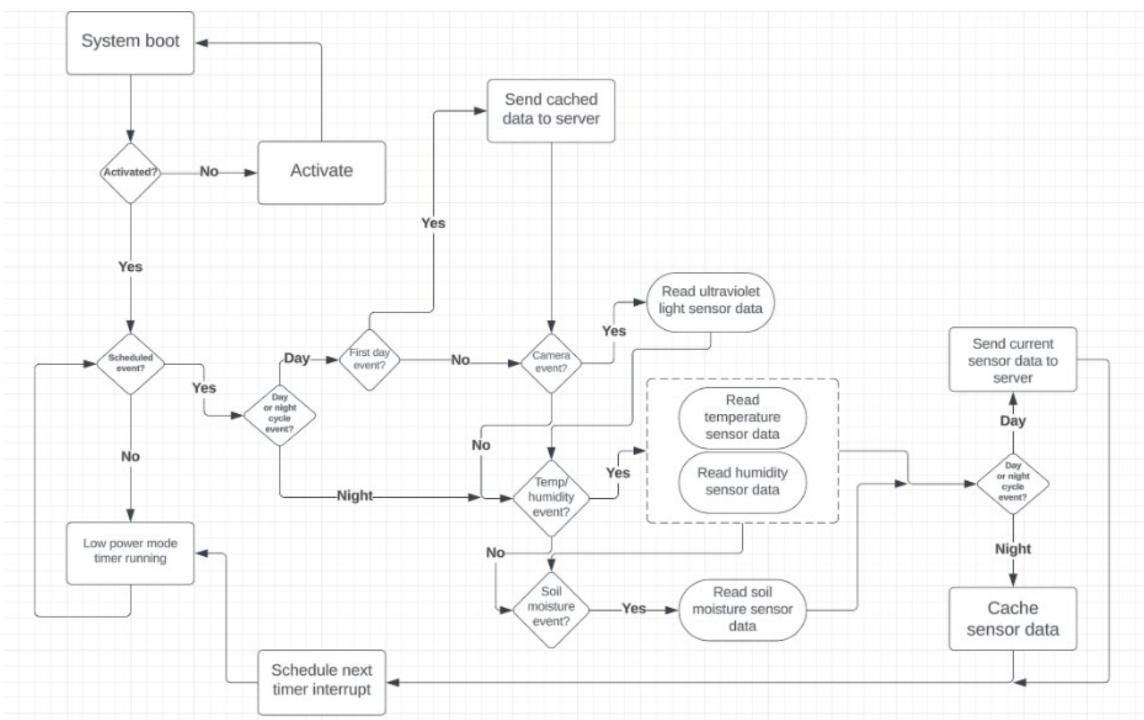


Figure 5.2.6-1. Sensor Package Software Flow

<u>5.2.7 Sensor Package State Diagram:</u>

For the sensor package, including the ultraviolet light sensor, there will be five total states that the microcontroller unit can be in at any given time in reference to the sensors. Each sensor timer interrupt will vary according to our final determination, but the procedure followed in each state is essentially the same.

This procedure will consist of an idle state where the ESP32-S3 and ESP32-C3 will be in the suitable low power mode to reduce power consumption, with this state continuing indefinitely until the real time clock raises an interrupt. At this point, the ESP32-S3 or ESP32-C3 will be in the sensor data request state, where it will send a data request signal to the desired sensor, and then enter the sensor data received state, where the sensor has transmitted the reading successfully to the microcontroller unit.

At this point, depending on the number of readings and the specific sensor data, the microcontroller unit will continue to go between these two states until the full data reading is complete. Once this has concluded the microcontroller will move into one of two next states, either the transmitting state or the caching state, which will depend on whether we are in a daytime event or a nighttime event. In the transmission state the full sensor data received by the microcontroller will be transmitted in real time either via Bluetooth or wi-fi depending on the data destination and will only occur during a daytime event. Following this state the microcontroller will return to the first state, being the idle low power mode state and the procedure continues endlessly. The caching state will only occur in a nighttime event, and the microcontroller will cache the received sensor data instead of transmitting it. The cached sensor data will be transmitted as a batch in a later day event. At the end of this state the microcontroller unit will also enter the infinite low power mode idle state and continue the cycle.

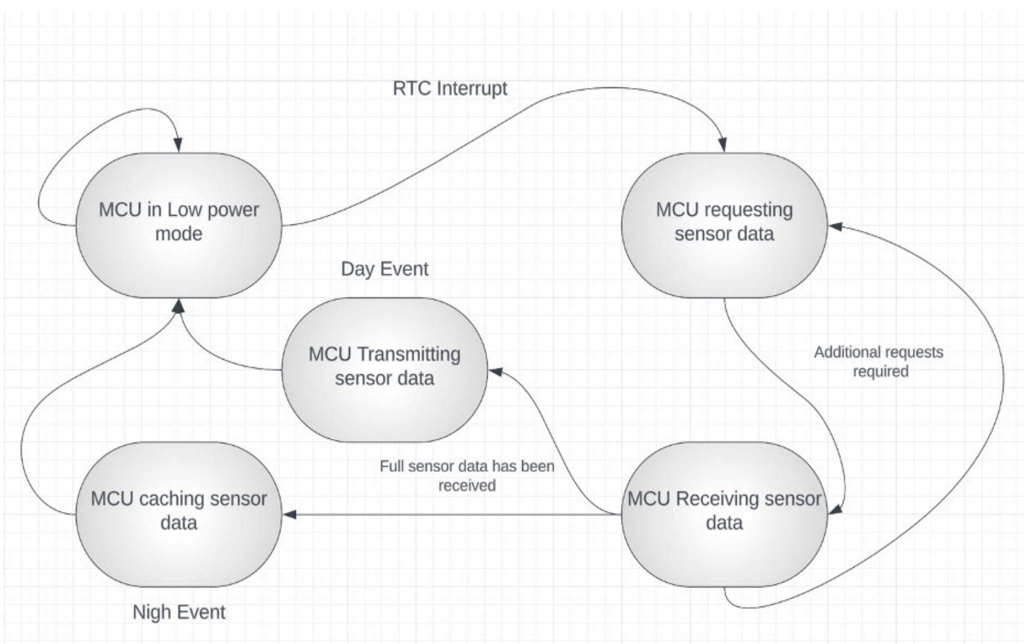


Figure 5.2.7-1. Sensor Package State Diagram

6 - System Testing and Evaluation

6.1 Sensor Testing

6.1.1 ESP32-S3 Initial Testing:

The initial testing of the ESP32-S3 consisted of first finding a development board for testing. The development board we used for testing was the ESP32-S3-DevkitC-1-N8R8. This board features the ESP32-S3 microcontroller unit, Pin headers for easy testing of other devices to the ESP32-S3, an integrated LED, and a UART to USB bridge including a USB port and USB to UART port. To utilize the development kit, it was necessary to install the Espressif IDF development framework to facilitate testing of the system. This IDF is available on all three of the major operating systems currently available, being Linux, MacOS, and Windows (It is worth noting that Windows does offer an integrated development environment as well called the Espressif-IDE).

The testing of the ESP32-S3 and the sensors discussed later in this section all were conducted on MacOS. Installing the Espressif-IDF was as simple as following the instructions on the Espressif website, which included installing the necessary packages which were CMake, and Ninja build as well as the most up to date version of Python being Python 3. From here we installed the Espressif IDF itself from the GitHub repository and set up all the tools and environment variables. With this direct installation, it was possible to code using the Espressif-IDF on any IDE or text editor, which we found to be a much better alternative to using the Visual Studio Code Espressif-IDF extension for programming (VSCode was still used as the preferred IDE as it is extremely flexible, we just did not utilize the Espressif-IDF Extension).

With the Espressif-IDF framework now installed on our system it was possible to begin programming for the ESP32-S3. Programming in this style for an embedded system utilizing a higher-level API instead of register level programming which was familiar to us from our embedded systems class took some getting used to. The Espressif-IDF was still interfaced with using the C programming language, which was familiar, and simplifies all of the major functions of embedded programming on the platform.

The initial testing of the ESP32-S3 was done by getting familiar with using the Espressif-IDF by first getting a simple GPIO output to blink an external LED connected to the microcontroller unit via a breadboard and then implementing the timer with an interrupt to blink the LED as well. The Espressif-IDF is well supported and includes extensive documentation outlining the general uses of the API as well as outlining the functions in detail. This API reference can be found on the Espressif website as well and is a vital resource in both learning the API as well as developing effective code for this system as it also includes various hardware details as well which simplifies research as it is often not necessary to sift through a standard datasheet. The API references also often contain links to an external GitHub repository created by Espressif which has all the driver code to reference as well as examples for common implementations such as getting a GPIO to work or implementing the timer interrupt. This detailed documentation is critical in the development for this platform and will simplify the development by a good amount when it comes to programming to interface with the sensors on the sensor package as well as transmitting via Bluetooth and wi-fi.

6.1.2 Sensiron SHT4x Testing:

For the initial testing of the Sensiron SHT4x series sensors we first had to determine which of the sensors we would be purchasing for testing and which breakout board to go with. For this there was only really one choice aside from building our own, coming from Adafruit, which offered breakout boards for the entire SHT4x lineup, including the SHT40, SHT41, and SHT45. These breakout boards included the actual SHT4x family sensor, as well as the supporting circuitry including pullup resistors and capacitors and direct connections to the SDA, SCL, Vin and Ground pins either with a STEMMAQT connector or by soldering a five-pin header to the breakout board, making them a good choice for easy development testing. When searching for the breakout board, the SHT45 was not available at the time so we went with the SHT41. The documentation for the SHT4x family is all the same for every sensor, with the only differences being the actual sensor accuracy itself. This meant that using the SHT41 for initial testing would not change the outcome as the code is directly interfaceable with the SHT45 when we begin building our system.

To begin interfacing the SHT41 with the ESP32-S3, it was necessary to familiarize ourselves with utilizing I2C on the ESP32-S3 through the Espressif-IDF. The Espressif-IDF allows I2C communication in either Master or Slave mode, meaning the ESP32 can act as either master or slave device. For our purposes we will be utilizing the master mode of the ESP32-S3 as the connected sensors will be slave devices.

The ESP32 handles I2C bus allocation by defining a bus handler which takes in a configuration from the associated struct, once the configuration struct is filled out according to the master configuration, i.e. SCL and SDA pins, CLK source, resistors, etc. then the Espressif-IDF allows for the function call i2c_new_master_bus(). This function call allocates and initializes the master bus and will allow us to connect a slave to the bus later.

The slave device is managed almost identically to the master, with its configuration struct where the user provides the address number, SCL frequency, and address bit length. Once this is configured, we call i2c_master_bas_add_device() to add that device to the master bus as a slave and allow for data read and write functions. As previously mentioned, along with the documentation found on the Espressif website, they include a link to examples of simple I2C interfacing from their GitHub repository which facilitated getting this test environment started. The documentation also specifies that for the ESP32 being in master mode there are three function calls possible, I2C read, I2C write, and a combined I2C read write operation.

With this information we were able to configure a simple test environment for the SHT41 using the Visual Studio Code IDE and configured the ESP32-S3 to master mode with the desired SDA and SCL pins. From here before configuring our SHT41 as the slave we needed to determine a few bits of information, the first of which being the I2C address for the sensor, as well as the proper method to generate a proper sensor reading. Referring to the datasheet for the SHT4x family, we found that our sensor was configured to I2C address 0x44, and it was also specified that there needs to be at least 6.9 to 8.3 milliseconds of a delay between a read and write operation otherwise the sensor will send a NACK signal across the I2C bus. The datasheet also specifies which commands we can send across the I2C bus from the master to the SHT41 to read the from the sensor the temperature and humidity reading in three different repeatability levels, read the sensor serial number, and configure the heater option of the sensor as well. Form here we were able to utilize this information to configure the SHT41 as a slave on the master bus with the correct address, and then send out a write signal with the correct command, utilize a small delay of 7 milliseconds, then send out a read signal to get the raw sensor data which was stored in a 6 byte array as specified from the SHT4x datasheet as well. Given this raw data, we then had to follow the datasheet once again to get the correct conversion for temperature in Fahrenheit as well as the humidity level in RH percentage.

<u>6.1.3 ams OSRAM AS7331 Testing:</u>

For the initial testing of the ams OSRAM AS7331 ultraviolet light sensor, we went with a breakout board manufactured by Spark Fun for initial testing. This board features the AS7331 sensor itself along with the supporting circuitry and initializes the I2C address to 0x74 by soldering those jumpers. The overview of the I2C communication by utilizing the Espressif-IDF to interface between the ESP32-S3 and the AS7331is the same as described above, with the same master configuration as used for the SHT41. The difference here was that the slave configuration included the address of the AS7331 which was 0x74, and we had to do some initial I2C write commands to configure the light sensor registers.

The I2C driver API provided by the Espressif-IDF as mentioned previously has three different functions for reading, writing, and a quick read write call. The write function works by sending the start signal, then the device address with the write bit set, then waits for the device acknowledgement, then transmits data, then sends a stop signal. The read function works the same way but instead sends the device address with the read bit set after the start condition, then receives the data instead of sending it. The write read function sends the start condition followed by the write call, then a repeated start and then the read call mentioned previously. These functions simplify communication via I2C on the ESP32 family of microcontroller units but does limit the programmer depending on the situation. With the SHT41 sensor there was no register accessing required which made these functions very useful for easy implementation, as we will discuss in a moment, the AS7331 does require register access and requires a repeated start condition which is not supported with this I2C function setup.

The datasheet for the AS7331 includes a general overview of how it expects the I2C communication to work between the microcontroller unit master and the AS7331 slave to transmit and receive data. For the write condition the procedure is standard, sending the start condition followed by the slave address with the write bit set. Following this the master should send the register address it is looking to access and then after the I2C acknowledgement the data is transmitted from the master to the slave. For the read sequence there is a slight difference after the write bit is set and the register is sent, the master sends a repeated start condition followed by the address appended with the read bit, then the data is sent from the slave to the master. The register we are referencing in both situations is also different, with the write procedure using the register we are writing to, i.e. one of the configuration registers, and the read procedure sending the register we are to begin reading from, which can be address 0h or address 2h depending on the setup of the configuration registers.

With this information we were able to begin interfacing with the AS7331 sensor utilizing the Espressif-IDF I2C write function and the I2C write read function to handle the repeated start condition required of the AS7331. From the AS7331 datasheet we were able to find that the recommended procedure for interface was to set a configuration across the I2C bus to the OSR register which keeps track of the state of the AS7331, then configure the registers as needed for the ultraviolet reading (we left these to default for testing), then set the OSR register to measurement mode allowing us to begin a measurement by setting a specific bit. The six output registers as defined in the datasheet were each 2 bytes long and transmitted the least significant byte first, which we stored in an array of 8-bit data values. For testing we utilized the first four bytes of the output register, which give the OSR state, the status bit, and the parallel temperature reading to the ultraviolet readings. The temperature readings were what we looked at to confirm the sensor was reading properly as the conversion was very simple. The ultraviolet sensing capabilities required more thought and testing to determine if they were correct readings which we will leave for senior design II.

6.1.4 Raspberry Pi No-IR camera NDVI data acquisition:

As part of our project, we're leveraging the Normalized Difference Vegetation Index (NDVI) to gauge plant health. To guarantee high-quality data, we put the Raspberry Pi No-IR camera through its paces, utilizing Python and OpenCV libraries for a barebones NDVI implementation.

We took a series of pictures of a test plant under varying lighting conditions and environments to gauge performance. The results are promising, as illustrated in Figure 6.1.4-1, which depicts NDVI color-mapped imagery highlighting areas with elevated active photosynthesis - think increased IR reflectivity. This is exactly what we're looking for: a visual representation of the plants health status.

Figure 6.1.4-1. NDVI Testing

Our tests revealed that some tweaking was necessary to accommodate differing light conditions. Specifically, we'll need to employ masking and exposure adjustment techniques to avoid overexposure. This is crucial because even slight variations in lighting can significantly impact NDVI accuracy. To ensure the camera captures the targeted 850nm wavelength for our NDVI calculations, we utilized an LED of the same wavelength. This allowed us to evaluate the camera's performance under controlled circumstances. By doing so, we're confident that our results will accurately reflect the plant's health status.

The test results show that the camera is sufficient for the task and will meet our requirements set in Chapter 2. We're excited about the prospects of using this technology to monitor plant health in real-world scenarios. The implications are significant: with NDVI, we can track changes in plant growth, detect early signs of disease or nutrient deficiencies, and even possibly optimize crop yields. While there's still much work to be done, our initial findings have us optimistic about the potential of this project. We're eager to continue refining our approach and exploring the many applications of NDVI-based plant health monitoring.

6.2 Plans for Senior Design II

The end goal for senior design II is of course to have our system fully built and functional from the design which we have compiled during senior design I and which has been described in this document. For senior design II we will have four specific overall goals which will be used to achieve our complete system. These four goals will be a fully functional and relatively accurate sensor package, a fully functional and relatively accurate camera package, a fully functional hardware section (including the printed circuit board, all the power system, and solar system), and a fully functional and accessible web application component.

For the sensor package system with the initial testing of three out of the four sensors completed, we have a solid foundation to begin working on for senior design II. The first goal of this section will be to get the soil moisture sensor reading successfully, and then acquiring an ESP32-C3 to port all the sensors initial tests over to ensure that they are working properly on the correct sensor for the specific package (being the soil moisture, humidity, and temp for the ESP32-C3 and the ultraviolet light for the ESP32-S3). With this functional, the next step will be to combine the sensor reading systems into one main firmware package which will most likely be broken into three drivers for each sensor system and combined into a fourth driver to run them all together according to the software flow described in section 5.2.6. With the software flow running successfully, the last functionality that will need to be implemented into the firmware will be the Bluetooth communication between the two ESP32 family sensors and the wi-fi communication between the ESP32-S3 and the web application. After this the complete testing of the sensor package will commence and allow for us to fine tune any necessary bugs or issues in the system.

For the camera package system, we will need to get the camera to successfully generate the readings which will be needed for the NDVI calculations. This will be done by first getting the camera to be able to read on command, then implement the readings from the ultraviolet light sensor which will be the only sensor not part of the sensor package, to periodically generate the camera readings. With the readings generated we will get the raspberry pi zero to process the image and generate the NDVI readings and send them via wi-fi to the web application. The main goal for this section in senior design II will be to get the camera package to follow the described software flow and interacting with the web application. Once this is complete like the sensor package ideally, we will have time for testing to iron out any issues with the system.

For the hardware section goals of senior design II, we will get the printed circuit board design for both the sensor and camera package finalized as described in section 7 of this document. With this design finalized we will be able to send the printed circuit board design off to be manufactured and allow us to begin testing the software components on the final circuit design instead of the breadboards. The second major part of the hardware section for senior design II will be to also finalize the design of the enclosure which will hold the entire system including the sensor and camera package along with the solar panel and battery. Once this is completed, we can begin to test the entire system in the complete enclosure and make tweaks as necessary. The main issue with this portion of senior design II will come down to good choices for the sensors which need ample exposure to make proper readings, while retaining enough protection from the elements to meet the water resistance standard we are aiming for with this system.

For the final goal section for senior design II, we will need to complete the software for the web application. The main design and configuration have been determined in this document already so what we will need to finish is setting up the database to store the sensor data and configure our host server and associate the database with it. From here we will need to form the necessary APIs for the web application for the user interactions and to run the periodic updates as needed. The bulk of this section will most likely consist of deciding on a good design to maximize the user experience and integrate necessary accessibility decisions to make sure it is accessible to everyone.

6.3 Performance Evaluation

With the goals for senior design II clearly defined, we will need to test the entire system or the sections of the system as well to ensure proper functionality within our specifications. For the evaluation of the performance of the system, we will conduct the testing in two different methods, breadboard, and printed circuit board.

Once the firmware for the sensors and camera system are completed, we will begin with initial testing on our breadboard setup. To determine the performance accurately for the sensors we will utilize known working sensors with similar accuracy to cross-reference our results and ensure the system is meeting our 90% accuracy specification. Ideally, we will perform these tests various times and in different locations to have a variety of sample data to ensure the system can perform in many locations instead of just in our testing environment.

For the camera system the testing will consist of taking various images in the suitable light and temperature conditions from the sensor system, then ensuring they are clear and able to generate the NDVI reading we are looking for. The testing of this will also be in various light conditions to ensure we have the proper range set for the ultraviolet light sensor. For the camera system there is no specific accuracy specification we are looking for, we just need it to reliably take the pictures and process them successfully.

Once the separated systems have been thoroughly tested and perform to our expectations, we will then connect the systems together onto the final printed circuit board and port the firmware. The same tests will then be conducted to ensure the migration to the final system did not affect any of the previous results, and if there are conduct the necessary fixes. As mentioned earlier the part of this system which will most likely require the most testing is going to be the humidity sensor which we will need to find a balance between accuracy and position, as this sensor needs to be exposed to generate a good reading. Ideally with the printed circuit board testing suitable to our accuracy specification, we will then test it in simulated harsher conditions to ensure that the system will meet the expected water resistance rating we are looking for and to ensure functionality on a realistic harsh situation a user may encounter.

For the software portion consisting of the web application, the only testing which we need to evaluate will be to ensure that there is a suitable response time and no major bugs which will inhibit the user from completing any of the critical features of the web application. During this phase we should also attempt to let the system run off the solar panel to determine if the battery life is expected or if we need to address this in any way. This should be handled during this section of testing as user requests and scheduled events to transmit via wi-fi will be the most taxing as far as power consumption is concerned for the system.

Overall, if we can achieve an accuracy level across all of the sensors of 90% and the desired battery life of the system off the solar panel, we will be able to label the project as a success.

7 - PCB PLANNING

In this section we will briefly go over the planned PCB layouts for our two PCBs and any design considerations that had to be made regarding them and their housing. The designs in this section are still prototypes and will be modified as we continue to further our design and testing. Specifically, the package size for the resistors and capacitors used is not fully set in stone yet as we will not have a definitive BOM until we have our PCB ready to manufacture as components go in and out of availability. Additionally, we will need further testing with the UV light sensor, ground moisture sensor, and temperature/humidity sensor to determine the housing that would give us the best results from these sensors. We will also do more work to optimize the footprint for both PCBs. The designs in this section will serve as templates/starting points as we continue to finalize the PCB layout into senior design 2. They will also serve as references for our planning on the design and research of housings to use for the final design.

7.1 Camera Package

Below is the prototype PCB design for our camera package. This PCB will have two micro-USB type-b ports for serial-JTAG and UART to allow for hardware debugging and programming. There are three sets of 2 pin headers for interface between the various power inputs and outputs and the PCB. These headers are located as close as possible to their respective regulators/controllers to minimize the length of the high-current traces. These traces will also definitely need to be expanded moving forwards. Specifically, the battery input trace is likely too small and will be heavily reviewed moving forwards to protect the PCB's thermal performance under load. Additionally, many of the capacitors and resistors may be able to be resized. However, we will need to balance the footprint of the finalized PCB with our own ability to debug the PCB as the smaller sized components (0402 and smaller) will likely be too small to hand de-solder and resolder. However, these are design considerations that we can make as we progress through our more thorough prototyping phase into the next semester in senior design 2. For now this layout suffices in giving us an, albeit too small, approximate size for the board.

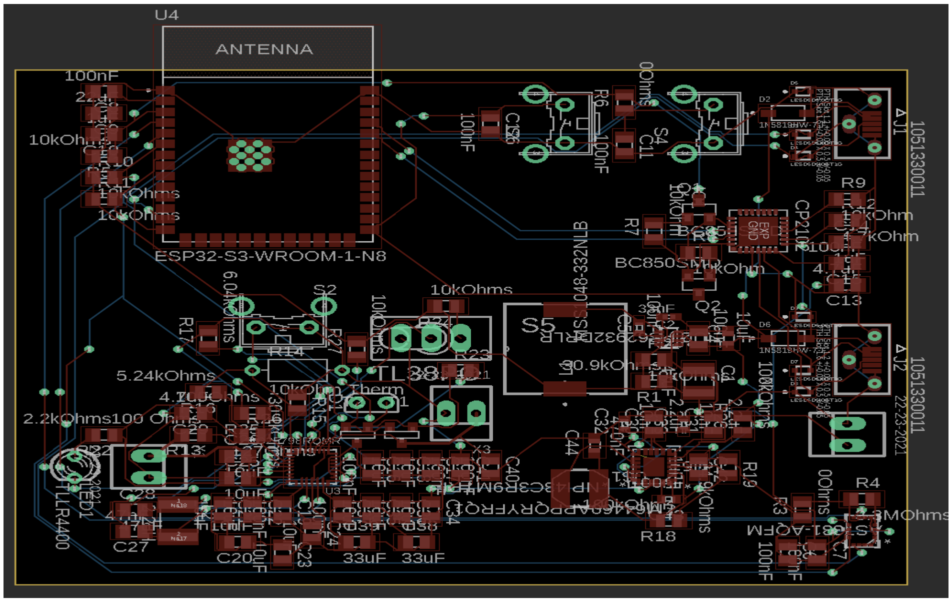


Figure 7.1-1. Camera Package PCB Design

Like we've been discussing this is just a prototype early layout. There has been little work into optimizing this design for thermal performance or sensor performance. As such the UV light sensor may have to be moved. One of the major design considerations with that sensor is the ability for it to generate accurate readings while attached to the camera package. Since it will be situated under the PV panel, we may need to create a third very small break-out PCB for just the UV light sensor and attach it to an extended arm sticking out from the main package. Another major concern for this PCB moving forwards will be thermal performance. These PCBs are meant to be used outdoors for very long periods of time and as such will often experience ambient temperatures far above the average. To mitigate this effect, we will need to rigorously ensure that our design is able to handle the currents required of load and supplied by the PV panel and battery without performance loss or damage to the system.

The main benefit we do get from this PCB layout is a good ball-park idea for the package size we will need for the enclosure. The current footprint of this PCB is 2.7x3.3 cm but for the reasons mentioned above this will need to be increased. However, this will give us a good starting point for understanding the approximate size we need for our enclosure. This will help with our planning into the next semester and will give us a head start on our realization of the final design.

7.2 Sensor Package

Below is the prototype PCB for our sensor package. As shown above there will be two micro-USB type-b ports to allow for programing of the MCU via UART and for debugging via JTAG. There will be an additional 3-pin header situated between these USB ports to allow connection between the 9V battery, the ground moisture sensor, and the MCU. The switches that were selected are placeholders as we do not yet know the exact dimensions of the housing we will use for this board and as such will re-size many components to fit our needs, the switches being prime examples of this. The MCU is situated on the left-hand side of the board with the PCB antenna as far away from any high frequency circuitry as possible. Notably the buck converter is as far away from the MCU as possible without increasing the footprint.

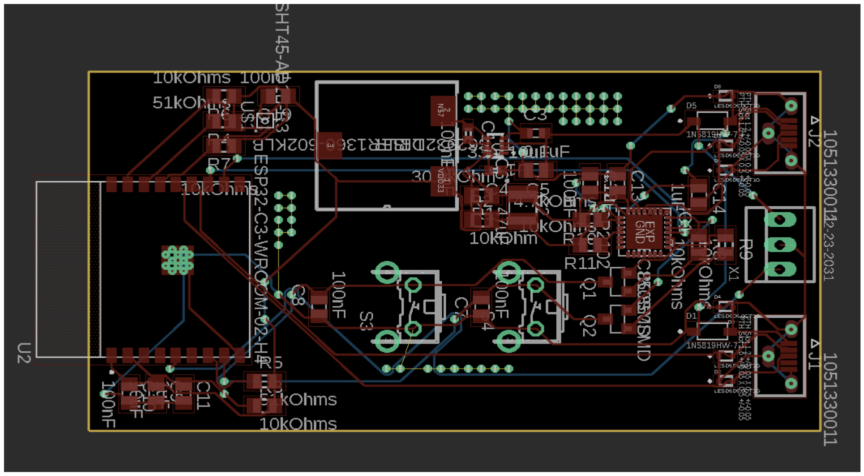


Figure 7.2-1. Sensor Package PCB Design

As mentioned above and at the beginning of this section many components will be resized as we continue into senior design two and have a more concrete idea of the enclosure size we will be limited to. The current footprint is slightly larger than 2x3 inches. This may be able to be reduced to around 2x2.5 inches with a reduction in the size of the capacitors associated with the UART to USB bridge and the MCU. Additionally, the inductor can be a smaller size, if necessary, but this would result in a non-trivial increase in cost so this is a last resort option if we are unable to reduce the size enough to fit into the housing we select.

Finally, as we discussed when we began this section further redesigns will be needed to ensure that the SHT45 sensor can take proper readings. The sensor may also need to be relocated to a breakout board or another section of the current PCB to have proper access to the vent designed to allow it to take its readings. This will be a large part of our future testing and design going into the next semester. One of the major challenges associated with this sensor will be finding a way to expose it to the appropriate amount of humidity without the PCB it is attached to being damaged. We have plans to attempt to deal with these challenges. Currently we are planning on creating a "vent" in the housing of the sensor package to allow a limited amount of external air into the housing. This will likely be guarded with a mesh that would prevent dust intrusion while allowing moisture in. However, this design will need to be tested and validated throughout the coming semester.

10 - ADMINISTRATIVE CONTENT

10.1 BILL OF MATERIALS

We set a target for our device to cost less than one hundred dollars per unit. The table below (Table 1) shows the total cost to build one device at our economy of scale. We were able to keep the cost down by selecting a cost efficient, yet powerful microcontroller unit with the Espressif ESP32 family of chips. We will build a total of five devices for testing and demonstration purposes during senior design II. The decision to create five devices comes from the five-board minimum from the PCB manufacturer and wanted a sufficient number of devices for testing to reduce the risk of hardware failure impacting our ability to demonstrate the device.

Table 10.1-1. Bill of Materials

Item	Model	Quantity	Price
MCU	ESP32-S3-WROOM-1-N16R8	1	$3.90
MCU	ESP32-C3-WROOM-02	1	$1.90
Raspberry pi	Zero 2 W	1	$15.00
Camera	Pi Cam No-IR	1	$25.00
Temp/Humidity Sensor	SHT45	1	$7.18
Moisture Sensor	PIM520	1	$7.50
UV Light sensor	AS7331	1	$10.86
Buck Regulator 1	TPS62932	2	$0.57
Buck Regulator 2	LMQ644A2-Q1	1	$5.12
Solar charge controller	BQ25798	1	$2.29
Solar Panel	Newpora NPA5S-12I	1	$21.49
Battery	Murata VTC6 18650	2	$6.49
PCB	PCBway	2	$5.00
3D Print material	ASA	350	$0.02
3D Print material	TPU	100	$0.02
		subtotal:	$129.15

	5 devices	total:	$645.74

The milestones for our project are laid out below in Table 10.1-2 and Table 10.1-3. These tables detail our deliverables and their due dates as we know them. Senior design I was planned according to the schedule we were issued at the beginning. This allowed us to give ourselves ample time to review our deliverables prior to the submission deadlines. Senior design II doesn't have as accurate of a timeline available at the time of writing this document, so the milestones for senior design II will be finalized when these become available. Additionally, certain aspects of Senior Design 2 are based on manufacturer wait times, which means we will not have exact deadlines for these specific items, instead having a soft deadline which we would like these to be completed by and work around those.

Table 10.1-2. Documentation Milestones

Project Documentation Milestones				
Milestone	Details	Start Date	Planned Date	Due Date
Divide and Conquer Initial 10 Page Document	Tentative plan for the project including tentative BOM and specifications.	1/27/2024	2/2/2024	2/2/2024
Recruitment of Review Board	Recruited three professors from UCF faculty to serve as reviewers. Professors recruited: Mark Maddox, Zakhia Abachar, Piotr Kulik.	1/22/2024	1/31/2024	2/2/2024
60-Page Document	First two-thirds of the document.	1/27/2024	3/25/2024	3/29/2024
Final Document	Final SD 1 document.	1/27/2024	4/20/2024	4/23/2024

Table 10-1-3. Project Creation Milestones

Project Creation Milestones				
Milestone	Details	Start Date	Planned Date	Due Date
Project Idea	Brainstorming and a meeting to go over various ideas for the project before deciding on one after deliberation.	1/11/2024	1/12/2024	2/2/2024

Component Selection	Selection of the essential components to begin system designs (e.g. MCU, camera, sensors, and charge controller).	1/15/2024	2/2/2024	2/7/2024
System Prototypes	Proof of concepts for the various hardware or software systems involved in the final project.	2/7/2024	4/16/2024	4/23/2024
Breadboard Prototyping	Finalization of design and testing on a breadboard.	4/23/2024	8/29/2024	9/2/2024
Software Prototyping	Implementation of software design and integration with the sensors done on the breadboard prototype before testing and finalizing.	8/19/2024	TBD SD2	TBD SD2
PCB Design/Manufacturing	Using the finalized design from prototyping to create the final PCB with Eagle and having the PCB manufactured.	9/2/2024	10/1/2024	TBD SD2
PCB Testing	Testing the PCB to ensure functionality and compatibility with software.	10/1/2024 (as soon as the PCB gets delivered)	TBD SD2	TBD SD2
Enclosure Fabrication	Creation of the housing for the PCB and sensors as well as the mount for the solar panel.	8/19/2024	TBD SD2	TBD SD2
Final Product Assembly and Testing	Final assembly of the product and testing in real-world settings to ensure functionality and ensure target accuracies are met.	TBD SD2	TBD SD2	TBD SD2

11 - CONCLUSION

This concludes our planning for the design of the PlantPulse. To reiterate, the purpose of our design is to enable the end user to have an easier time with their home gardening by giving them a powerful diagnostic tool about the health of their plants. This tool will use NDVI imaging and a variety of IoT sensors to provide valuable data such as temperature, humidity, soil moisture, and UV light. These will combine to create a system that can recognize the presence of plant health issues via NDVI and diagnose those issues with the IoT sensors.

Our design is to be protected from the elements by two enclosures with IPx ratings acceptable to average conditions. We will also take steps in our testing and prototyping process to ensure that the thermal performance of the system is acceptable. Our design will also run completely autonomously without intervention from the end user.

This design will be accomplished by the combination of a Pi Cam with its IR filter removed and three IoT sensors. The camera and sensors will be split up into three different packages. The camera package will handle the communication with the web database, the UV light sensor, and the Pi Zero 2 W. The Pi Zero 2 W will handle the operation of the Pi cam and the NDVI calculations and image processing. The Sensor package will consist of the temperature/humidity sensor and the soil moisture sensor. This will allow the user to have particular readings relevant to the plant they are monitoring via NDVI with the Pi Zero 2 W. After extensive research the sensors we decided on for an acceptable level of accuracy are the SHT45, the AS7331, and the PIM520. These sensors will wake up periodically to provide the user with periodic readings to ensure a high level of accuracy.

The packages will be managed by an ESP32-C3-WROOM-02 on the sensor package and an ESP32-S3-WROOM-1 on the camera package. These MCUs will manage the sensors through a variety of GPIOs and I2C. They both control the timing of the sensor readings as well as process the data gathered by the IoT sensors. The Pi Zero 2 W is the controller for the Pi cam and will process the NDVI images. These images are sent to the ESP32-S3 module along with the sensor data to be transmitted to the web database via Wi-Fi. The communications between the two MCU modules and the Pi Zero 2 W is over Bluetooth low power, and only the ESP32-S3 module communicates over Wi-Fi to reduce bandwidth usage and power consumption. These MCUs are programmed over a UART-USB bridge.

The power for the camera package and Pi Zero 2 W will be a photovoltaic panel and backup battery. The solar panel will be controlled via the BQ25798 which has an MPPT algorithm implemented. This controller will run mostly autonomously after being configured on POR. This controller will control the charge cycle for the battery. The controller will sense the output current and load requirements to activate the backup mode when load power requirements exceed the minimum power delivered by the photovoltaic panel. When the input voltage level drops below the battery voltage the battery will become the main power source and the input channel will be cut-off. The power from this controller will feed into two buck regulators which will regulate the voltage to 5 and 3.3 volts for the Pi Zero 2 W and the MCU/UV light sensor respectively.

The power for the sensor package will be a simpler combination of a 9V battery and a single buck regulator to regulate the voltage down to 3.3V for the sensors and MCU.

Once the data has been gathered in the camera package via Bluetooth low power it will be transmitted to a database via Wi-Fi. This database is the main link between the final system and the end user. This database will connect to the website or (as a stretch goal) app and will allow the end user to view the results gathered from the system. Through this server the user can view the past readings after their processing and adjustments. This site will also lay-out the NDVI imaging in a way that an amateur is able to understand.

Through this innovative combination of prior technologies, we will provide access to a never-before-seen perspective in consumer gardening. With the combination of NDVI imaging and a variety of diagnostic features the uncertainty, and the unknown can be removed from the process of gardening! Our system can provide you with the ability to take a quantitative approach to your gardening. With our system you can remove the frustration and confusion from your gardening and have fun simply enjoying your plants growth

APPENDICES

APPENDIX A - references

[1] “BQ25756E,” BQ25756E data sheet, product information and support | TI.com, https://www.ti.com/product/BQ25756E (accessed Jan. 29, 2024).

[2] “TPS62933,” TPS62933 data sheet, product information and support | TI.com, https://www.ti.com/product/TPS62933 (accessed Jan. 22, 2024).

[3] C. Cox, “How to use solar panels to power the Arduino,” Circuit Basics, https://www.circuitbasics.com/how-to-use-solar-panels-to-power-the-arduino/ (accessed Jan. 30, 2024).

[4] “Relative humidity and temperature sensor - sensirion,” Sensirion, https://sensirion.com/media/documents/33FD6951/63E1087C/Datasheet_SHT4x_1.pdf (accessed Jan. 25, 2024).

[5] A. Industries, “5V 0.6W Mini Solar Panel - ETFE,” adafruit industries blog RSS, https://www.adafruit.com/product/5856 (accessed Jan. 22, 2024).

[6] “Is now- on semiconductor,” Onsemi, https://www.onsemi.com/pub/Collateral/DN06054.PDF (accessed Jan. 27, 2024).

[7] Ingress Protection (IP) ratings, https://www.iec.ch/ip-ratings (accessed Jan. 30, 2024).

[8] “MCP16502,” Microchip, https://www.microchip.com/en-us/product/mcp16502 (accessed Feb. 11, 2024).

[9] “TPS62932,” TPS62932 data sheet, product information and support | TI.com, https://www.ti.com/product/TPS62932 (accessed Mar. 2, 2024).

[10] "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," in IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016) , vol., no., pp.1-4379, 26 Feb. 2021,https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9363693&isnumber=9363692

[11] "IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN - Specific Requirements - Part 15: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)," in IEEE Std 802.15.1-2002 , vol., no., pp.1-473, 14 June 2002, doi: 10.1109/IEEESTD.2002.93621. https://ieeexplore.ieee.org/document/1016473

[12] NASA, “Measuring Vegetation (NDVI & EVI),” Nasa.gov, Aug. 30, 2000. https://earthobservatory.nasa.gov/features/MeasuringVegetation/measuring_vegetation_2.php

[13] “Adafruit STEMMA Soil Sensor -I2C Capacitive Moisture Sensor.” Accessed: Apr. 22, 2024. [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/698/4026_Web.pdf

[14] “OV2640: Specs, Camera, Datasheet & Alternative (2022 Report),” Arducam. https://www.arducam.com/ov2640/

[15] “CMOS OV5640D AF Camera Module 1/4-Inch 5-Megapixel Module Datasheet,” 2017. Accessed: Apr. 22, 2024. [Online]. Available: https://www.arducam.com/downloads/modules/OV5640/5Megapixel_OV5640D_AF_CMOS_Camera_Module_DS.pdf

[16] “Raspberry Pi Camera Module 3 Standard NoIR Wide NoIR Wide,” 2023. Available: https://datasheets.raspberrypi.com/camera/camera-module-3-product-brief.pdf

[17] “J. I. Obianyo, “Humidity Sensors, Major Types and Applications,” *www.intechopen.com*, Jan. 04, 2023. https://www.intechopen.com/chapters/76696# (accessed Apr. 22, 2024).”

[18] “Saurabh, “Humidity Sensor: Basics, Usage, Parameters and Applications,” *Electronics For You*, Oct. 22, 2019. https://www.electronicsforu.com/tech-zone/electronics-components/humidity-sensor-basic-usage-parameter”

[19] “Anusha, “Humidity Sensor - Types and Working Principle,” *Electronics Hub*, Jun. 07, 2017. https://www.electronicshub.org/humidity-sensor-types-working-principle/”

[20]““Technology,” *sensirion.com*. https://sensirion.com/products/technology/”

[21]“D. K. Roveti, “Choosing a Humidity Sensor: A Review of Three Technologies,” *Fierce Electronics*, Jul. 01, 2001. https://www.fierceelectronics.com/components/choosing-a-humidity-sensor-a-review-three-technologies”

[22]““What is a UV sensor?,” *Davis Instruments*. https://www.davisinstruments.com/pages/what-is-a-uv-sensor#:~:text=A%20photodiode%2Dtype%20UV%20sensor”

[23]““Photo resistor | Resistor types | Resistor Guide,” *eepower.com*. https://eepower.com/resistor-guide/resistor-types/photo-resistor/#”

[24]“Wavelength Electronics, “PHOTODIODE BASICS – Wavelength Electronics,” *Teamwavelength.com*, Jan. 13, 2020. https://www.teamwavelength.com/photodiode-basics/”

[25]“*AS7331 Datasheet*. https://www.mouser.com/catalog/specsheets/amsOsram_AS7331_DS001047_1-00.pdf ”

[26]““LITE-ON DCC RELEASE Optical Sensor Product Data Sheet,” 2016. Available: https://optoelectronics.liteon.com/upload/download/DS86-2015-0004/LTR-390UV_Final_%20DS_V1%201.pdf”

[27]“S. Ahmed, “Optical Sensors Application Note SensorXplorer TM Installation Guide A P P L I C A T I O N N O T E.” Accessed: Apr. 22, 2024. [Online]. Available: https://www.vishay.com/docs/84922/sensorxplorerinstallation.pdf”

[28]“S. Hackett, “What is an RTD sensor?,” *Thermal Resource Management*, Nov. 25, 2022. https://thermal-resources.com/what-is-an-rtd-sensor/ (accessed Apr. 22, 2024).”

[29]““A Guide to Semiconductor Temperature Sensors,” *AZoSensors.com*, Jul. 17, 2023. https://www.azosensors.com/article.aspx?ArticleID=2852”

[30]“J. Gums, “Types of Temperature Sensors | DigiKey,” *www.digikey.com*, Jan. 26, 2018. https://www.digikey.com/en/blog/types-of-temperature-sensors”

[31]““What temperature probe is better for you? | Omega Engineering,” *www.omega.co.uk*. https://www.omega.co.uk/temperature/z/thermocouple-RTD.html#:~:text=RTDs%20have%20typically%20an%20accuracy”

[32]““How do soil moisture sensors work?,” *GroPoint Soil Moisture Sensing*. https://www.gropoint.com/new-blog/how-do-soil-moisture-sensors-work”

[33]““Sensor Technologies,” *SoilSensor.com*. https://soilsensor.com/sensors/sensor-technologies/”

[34]““Different Types of Soil Moisture Sensors - Latest Open Tech From Seeed,” *Latest Open Tech From Seeed*, Jul. 22, 2022. https://www.seeedstudio.com/blog/2022/07/22/%ef%bf%bcdifferent-types-of-soil-moisture-sensors%ef%bf%bc/”

[35]““Soil Moisture Sensor Classification,” *www.niubol.com*. https://www.niubol.com/Product-knowledge/Soil-Moisture-Sensor-Classification.html (accessed Apr. 22, 2024).”

[36]““Coaxial Impedance Dielectric Reflectometry Archives,” *SoilSensor.com*. https://soilsensor.com/articles/tag/coaxial-impedance-dielectric-reflectometry/ (accessed Apr. 22, 2024).”

[37]““Getting Started with Grow,” *learn.pimoroni.com*. https://learn.pimoroni.com/article/assembling-grow (accessed Apr. 22, 2024).”

[38]““Adafruit STEMMA Soil Sensor - I2C Capacitive Moisture Sensor,” *Adafruit Learning System*. https://learn.adafruit.com/adafruit-stemma-soil-sensor-i2c-capacitive-moisture-sensor ”

[39]““Grove - Capacitive Soil Moisture Sensor (Corrosion Resistant),” *www.seeedstudio.com*, Dec. 06, 2023. https://www.seeedstudio.com/Grove-Capacitive-Moisture-Sensor-Corrosion-Resistant.html (accessed Apr. 22, 2024).”

[40] “Le, S. Supporting, I. 11, G. Wi-Fi, and Le), “ESP32-S3 Series Datasheet 2.4 GHz Wi-Fi + Bluetooth Including.” Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf”

[41] ""ESP32C3 Series Datasheet UltraLowPower SoC with RISCV SingleCore CPU Supporting IEEE 802.11b/g/n (2.4 GHz WiFi) and Bluetooth ® 5 (LE) Including." Available: https://www.espressif.com/sites/default/files/documentation/esp32-c3_datasheet_en.pdf"

[42] "R. P. Ltd, "Raspberry Pi Zero 2 W," *Raspberry Pi*. https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/ "

[43]""SAMA7G5 Series Up to 1-GHz Arm ® Cortex ® -A7, MIPI Camera, Dual Ethernet, Audio and Security." Accessed: Apr. 22, 2024. [Online]. Available: https://www.mouser.com/datasheet/2/268/SAMA7G5_Series_Data_Sheet_DS60001765A-2949408.pdf"

[44] "1635-2022 - IEEE/ASHRAE Guide for the ventilation and thermal management of batteries for stationary applications | IEEE standard | IEEE xplore," IEEE Xplore, https://ieeexplore.ieee.org/document/9966500/;jsessionid=oCj7-MH40Gy8heN-bBs8ojxUr2sUuD-jPjimxsoQIhsXPQdbW-79!1905994060 (accessed Apr. 5, 2024).

[45] "1149.1-2013 - IEEE standard for test access port and boundary-scan architecture | IEEE standard | IEEE xplore," IEEE Xplore, https://ieeexplore.ieee.org/document/6515989 (accessed Apr. 23, 2024).

[46] "1562-2021 - IEEE recommended practice for sizing stand-alone photovoltaic (PV) systems | IEEE standard | IEEE xplore," IEEE Xplore, https://ieeexplore.ieee.org/document/9528316 (accessed Mar. 29, 2024).

[47] "ESP32-C3 - technical documents," ESPressif, https://docs.espressif.com/projects/esp-hardware-design-guidelines/en/latest/esp32c3/esp-hardware-design-guidelines-en-master-esp32c3.pdf (accessed Apr. 4, 2024).

[48] J.-L. Aufranc, "A deep dive into Raspberry Pi Zero 2 W's power consumption - CNX software," CNX Software - Embedded Systems News, https://www.cnx-software.com/2021/12/09/raspberry-pi-zero-2-w-power-consumption/ (accessed Apr. 8, 2024).

[49] "TPS566231," TPS566231 data sheet, product information and support | TI.com, https://www.ti.com/product/TPS566231 (accessed Apr. 10, 2024).

[50] "LMQ644A2-Q1," LMQ644A2-Q1 data sheet, product information and support | TI.com, https://www.ti.com/product/LMQ644A2-Q1 (accessed Apr. 10, 2024).

[51] "5 watt 6 volt solar panel - ETFE," Voltaic Systems, https://voltaicsystems.com/5-watt-panel-etfe/ (accessed Apr. 8, 2024).

[52] alexeames, Raspberry Pi, https://forums.raspberrypi.com/viewtopic.php?f=63&t=147687&start=25#p985213 (accessed Mar. 14, 2024).

[53] "5W 12V Monocrystalline Solar Panel," Newpowa, https://www.newpowa.com/5w-12v-monocrystalline-solar-panel/ (accessed Apr. 14, 2024).

[54] "BL0750F5030481S1PCMC GlobTek, Inc. | Battery Products | DigiKey," DigiKey, https://www.digikey.com/en/products/detail/globtek-inc/BL0750F5030481S1PCMC/10187617 (accessed Apr. 1, 2024).

[55] "Sony: Murata VTC6 18650 3000mAh 15A Battery," IMR Batteries, https://www.imrbatteries.com/sony-murata-vtc6-18650-3000mah-15a-battery/ (accessed Apr. 9, 2024).

[56] "PRT-13851 Sparkfun Electronics | Battery Products | DigiKey," Digi Key, https://www.digikey.com/en/products/detail/sparkfun-electronics/PRT-13851/6605199 (accessed Mar. 9, 2024).

[57] "BQ25798," BQ25798 data sheet, product information and support | TI.com, https://www.ti.com/product/BQ25798 (accessed Mar. 12, 2024).

[58] "BQ24210," BQ24210 data sheet, product information and support | TI.com, https://www.ti.com/product/BQ24210 (accessed Mar. 10, 2024).

[59] A. Industries, "5V 1.2W Solar Panel - ETFE," adafruit industries blog RSS, https://www.adafruit.com/product/5368 (accessed Mar. 18, 2024).

[60] "I2C-bus specification and User Manual," I2C-bus specification and user manual, https://www.nxp.com/docs/en/user-guide/UM10204.pdf (accessed Apr. 10, 2024).

[61] "946-2020 - IEEE recommended practice for the design of DC Power Systems for stationary applications | IEEE standard | IEEE xplore," IEEE Xplore, https://ieeexplore.ieee.org/document/9206101 (accessed Apr. 23, 2024).

[62] Ingress Protection (IP) ratings, https://www.iec.ch/ip-ratings (accessed Apr. 5, 2024).

[63] "1625-2004 - IEEE standard for Rechargable Batteries for portable computing | IEEE standard | IEEE xplore," IEEE Xplore, https://ieeexplore.ieee.org/document/1309986 (accessed Apr. 23, 2024).

[64] "LM64460-Q1," LM64460-Q1 data sheet, product information and support | TI.com, https://www.ti.com/product/LM64460-Q1 (accessed Apr. 17, 2024).

[65] WEBENCH Power Designer, https://webench.ti.com/power-designer/ (accessed Mar. 21, 2024).

[66] "Battery Charger Fundamentals: Article: Mps," Battery Charger Fundamentals | Article | MPS, https://www.monolithicpower.com/battery-charger-fundamentals (accessed Mar. 13, 2024).

[67] A. Gates, "What is Maximum Power Point Tracking (MPPT)," NAZ Solar Electric, https://www.solar-electric.com/learning-center/mppt-solar-charge-controllers.html/ (accessed Mar. 1, 2024).

APPENDIX B - copyright permissions

NDVI image used in user interface mockup is courtesy of K-State Research and Extension provided under Creative Commons 2.0 and can be found at the link provided https://www.flickr.com/photos/ksrecomm/8467659757

APPENDIX C - other