Smart Health System for Plants

ECE 445 Design Document - Fall 2022

Team 13

Rohan Prasad, Tilak Patel, Yash Parikh

Professor: Arne Fliflet

TA: Hojoon Ryu

Contents

1 Introduction

1.1 Problem

There are many families in this world that - for a variety of reasons - are away from home and have plants sitting at home waiting for them to come back and provide water and sunlight. Further, many families love to have plants but don't have time to watch over them due to jobs and busy schedules. In these times, many plants can die out causing the owners to either purchase a new plant or throw out the old one completely. This is not only a problem of neglect, but also sustainability on a broader scope.

1.2 Solution

To solve this problem, we would create a Smart Health System for Plants with a phone app for the owner. We would create a plant potter which is built in with our Smart Health System and provides connectivity to the owner through an app. In our system, we would use different sensors to measure values like humidity, soil moisture and sunlight provided to determine exactly how much water/sunlight the plant will need. We then pump in water from our water reservoir straight to the roots and provide light when needed. Further, through the app, the owner would also be able to provide manual water and artificial light when they want to and see critical values from the sensor module. Overall, this Smart Health System for Plants will provide plants with the most ideal conditions they need to grow and survive and owners will never have to worry about dead plants due to their busy schedules and family vacations. Further, this system will consist of a sensor module, microcontroller, watering system, artificial sunlight system and a phone app.

1.3 Visual Aid

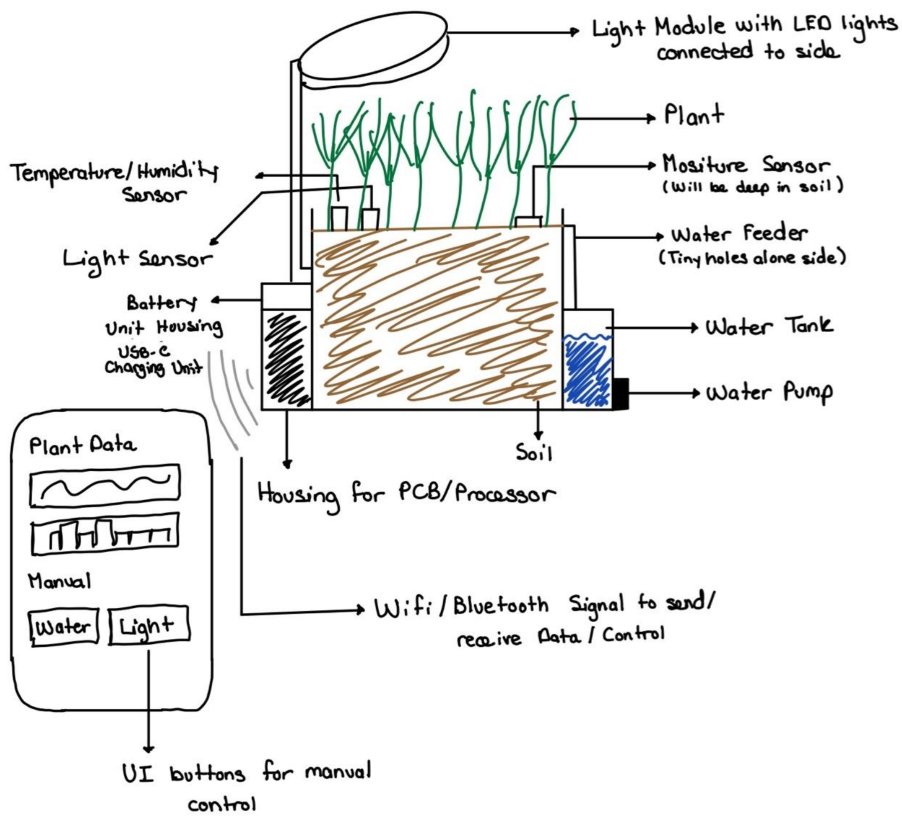


Figure 1: A rough visual aid for plant potter with our smart system

1.4 High Level Requirements

- The system should provide appropriate amount of water and light when required by the MCU/Microprocessor to do so using our created algorithm
- The system should be able to run on battery power and recharge the battery using the solar charging unit
- MCU/Microprocessor is able to communicate with a central system that the phone app can poll from every few minutes and aggregate plant information and metrics to display to the user

2 Design

2.1 Block Design

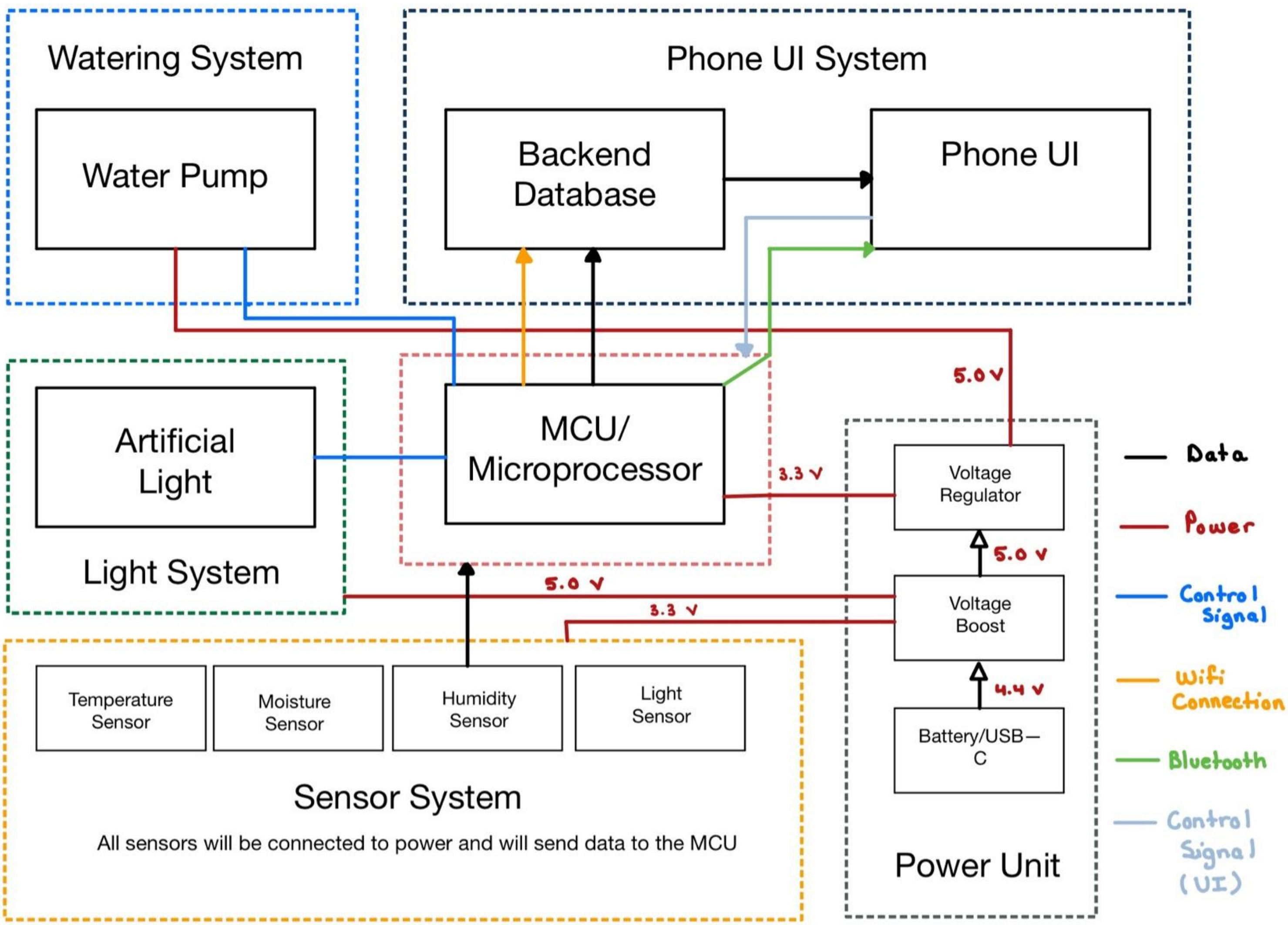


Figure 2: Block diagram of the layout of our system

2.2 Functional Overview & Subsystem Requirements

2.2.1 MCU/Microprocessor

The ESP32 processor/dev kit will be used in our design to be the center control system. The processor will collect raw data from the sensor module and convert the data to be used in our algorithm that can be used to signal the water pump motor and artificial light switch when needed. Furthermore, the processor will also communicate with our remote database via wifi connectivity to store the converted sensor data. This data from the remote database will then be polled by our phone app every few minutes and presented to the user in a meaningful way. Altogether, the converted data will provide the user with some insight on the health of their plant and the risks that could arise. The ESP32 will also provide us with bluetooth capabilities which will be used by our phone app to communicate with the processor for controls like manual watering and lighting. The processor will also be connected to our power unit which will use a voltage regulator circuit to reduce the input voltage to 3.3 volts for our processor. This subsystem will be the most critical part of our design as two out of the three requirements rely on it. In summary, the ESP32 processor is the most critical part of our design as it interacts with all our subsystems and processes the raw data received from the sensors to control various aspects of our system.

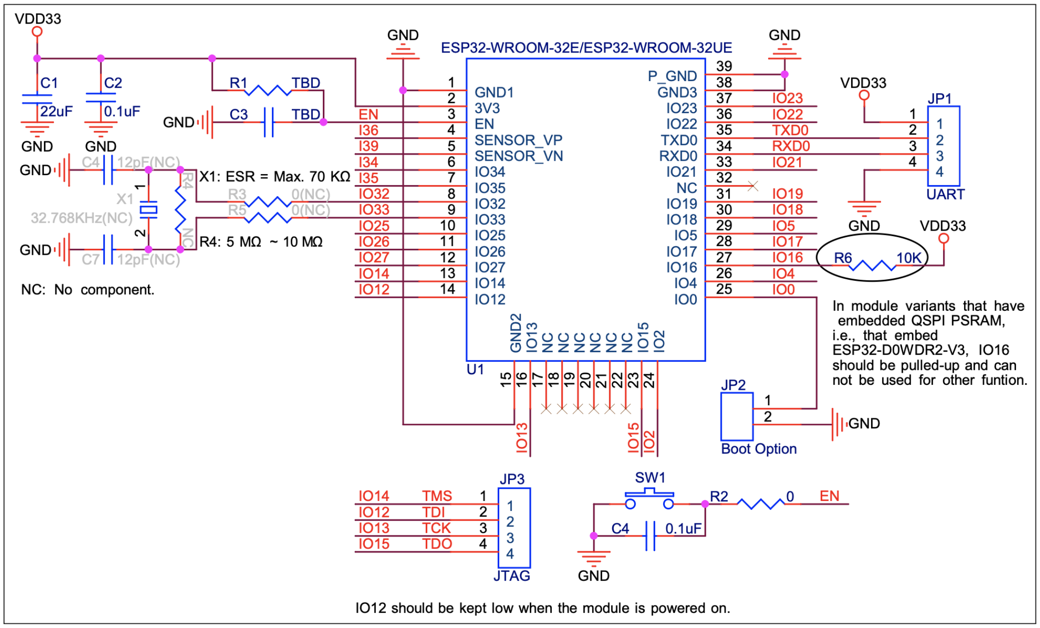


Figure 3: Setup for microprocessor and components needs to upload code/run processor[3]

Requirement	*Verification*
The processor must be able to receive manual signals from the phone app and override the algorithm to run the manual signal	To test that the processor can receive the manual signal from the phone app: 1. Power on the processor and start running the coded algorithm 2. Connect the phone app to the processor via bluetooth 3. Press the manual watering or lighting button in the app and verify we can see water pump start or LED light ring turn on
The processor must be able to transfer processed sensor data into our remote backend service via wifi connectivity every 2 minutes	To test that the processor can transfer the data into the backend service 1. Power on the processor and verify it is connected to the wifi 2. Upload the algorithm and start running the program 3. Verify in backend service that POST calls

Requirement	*Verification*
	are occurring every 2 minutes
The processor must be able to collect data from the sensors every minute and process it into the algorithm to trigger the watering system or artificial lighting system	To test that the processor is collecting data every minute from the sensors 1. Power on the processor and upload the algorithm 2. Start running the program 3. Manually simulate sensor values using a power supply or by using the sensor directly 4. Verify if the sensor values are in the coded range for the watering system then the system is turned on and off after proper calculated run time 5. Repeat same procedure for artificial lighting system

2.2.2 Sensor System

The sensor system will consist of a temperature sensor, humidity sensor, moisture sensor and light sensor. The sensors will provide data to the processor at an interval of 2 minutes which will be processed by our algorithm to trigger either the watering system or light system or both. The sensor system will also be connected to the power unit which will provide all the sensors their 3.3 volt operating voltage.

Temperature sensor:

The temperature sensor (DHT22 Sensor) will be used to collect data about the environmental temperature around the plant. The sensor will send the raw data to the processor through an I/O port which will convert the data and use it in our algorithm.

Humidity sensor:

The humidity sensor (DHT22 Sensor) will be used to collect data about the environmental humidity around the plant. The sensor will send the raw data to the processor through an I/O port which will convert the data and use it in our algorithm.

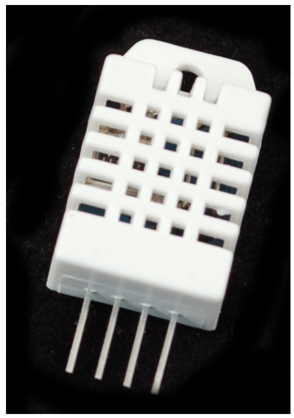


Figure 4: Image of DHT22 Sensor [2]

Light sensor:

The light sensor (TSL2561 Digital Light Sensor) will be used to gather data on how much light is provided to the plant. The light sensor provides us with three different light sensing modes including infrared mode, full-spectrum mode and human visible mode. We will use the full-spectrum mode which will return a value in the units of lux to our processor. We will have to convert the unit from lux to something meaningful in our algorithm. Lastly, the sensor will be connected to our processor using an I/O port which will collect the data.

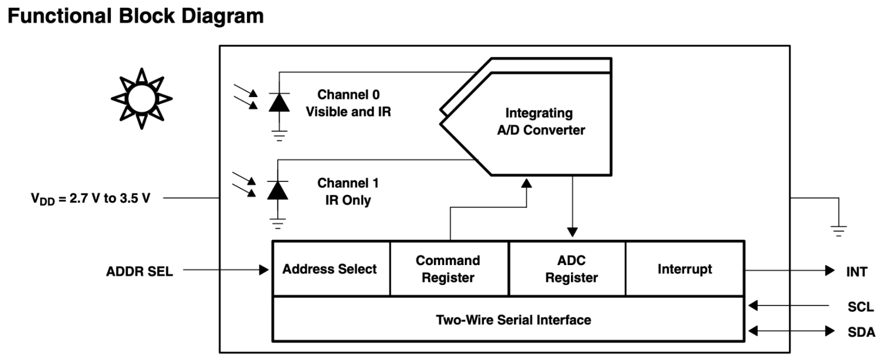


Figure 5: Diagram of how TSL2561 light sensor works [5]

Moisture sensor:

The moisture sensor (Stemma Soil Sensor) will be put about two inches into the soil and will gather data on moisture levels of the soil. This sensor is critical in our design because it will be used mainly to control the watering system. The sensor will be connected to our processor through an I/O port which will collect the data and use it in our algorithm.

Figure 6: Image of Moisture Sensor [1]

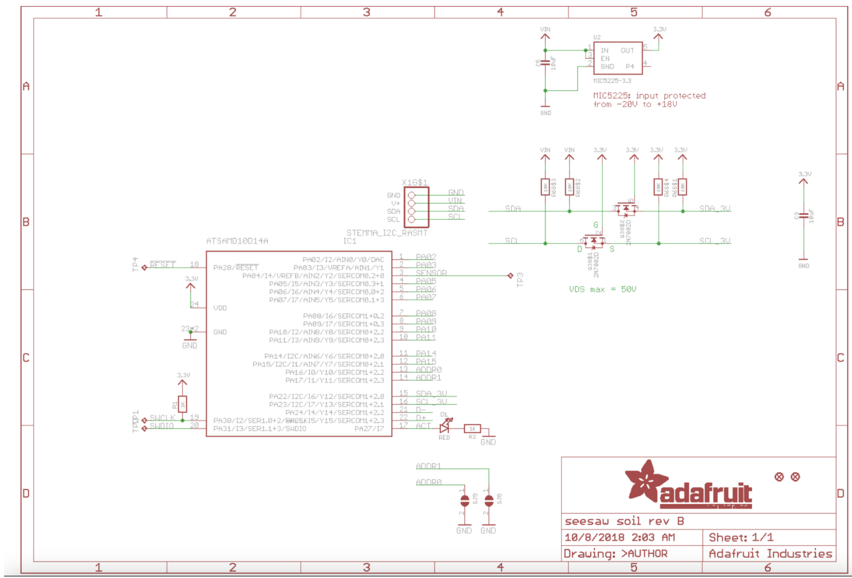


Figure 7: Schematic of Moisture Sensor from Adafruit [1]

In summary, our sensor system will be used by the processor to collect raw data and convert it to be used in our algorithm. The sensor data is critical because it will determine when our processor will turn off and on our watering system and artificial light system.

Requirement	*Verification*
The sensors should be able to provide new data to the processor every 2 minutes	To test that the sensors are providing new data every 2 minutes 1. Power on the processor and verify all sensors are connected 2. Manually change sensor values every 2 minutes 3. Verify the processor is able to receive the new sensor values using print statements

Requirement	*Verification*
The sensors are able to provide accurate data with a 5% tolerance	To test that the sensors are providing accurate data 1. Power on the processor and verify all sensors are connected 2. Create a controlled testing environment for each sensor (i.e controlled temperature/humidity room for temperature/humidity sensor, controlled soil with equal amounts of water for moisture sensor, different lighting environments for light sensor) 3. Verify sensor is able to provide accurate values for similar control environments with a ±5% tolerance

2.2.3 Watering System

The watering system will consist of two main components which include the water reservoir and the water pump. The water reservoir will be used in our system to store water which can feed the plant three to four times before needing a refill. The water pump will be used to take the water from our reservoir and provide it to the plant when needed. In our design, we will use a 5 volt submersible pump which will have two pipes connected. One of the pipes will be connected to the output which will lead water into the plant while the other will be connected to the input which will get water from our water reservoir. Furthermore, our water pump will be connected and controlled by the processor which will signal the pump to turn off and on after our algorithm processes the data collected from the sensors. The water pump will also be connected to our power unit which will use a voltage boost circuit to provide our water pump enough voltage needed to operate. In summary, our 5 volt submersible water pump and our water reservoir will work together to provide the plant water when instructed to do so by our processor.

Requirement	*Verification*
The water reservoir must be able to hold enough water to feed the plant three-four times before needing a refill	To test that the water reservoir can hold enough water to feed the plant three-four time: 4. Power on the processor and set the signal to the water pump to high 5. Repeat this three-four times and verify we do not run out of water before the third or fourth time
The water pump must be able to turn on and off when instructed by the processor for the calculated amount of time from our algorithm	To test that the water pump turns on and off for a specific amount of time: 1. Power on the processor then upload and run the algorithm 2. Code the algorithm to turn the water pump on for a specific amount of time 3. Manual get data on the amount of time the water pump turned on using a stopwatch 4. Verify the manually collect time and the coded time are within ±1%

2.2.4 Artificial Light System

The artificial light system will consist of a LED ring light which will be attached above the plant as shown in the visual aid (page 4). The purpose of this system is to provide light to the plant just like the sun would when there is not enough sunlight reaching the plant. The LED light will be connected to the power unit which will use our voltage boost circuit to provide the LED light the ideal 5 volt operating voltage. Furthermore, the LED light will be connected to the processor which will signal the light to turn on and off after our algorithm processes the data collected from the sensors. In summary, the LED ring light is also another critical component in our design as it is needed for one of our high level requirements and will help provide artificial sunlight to the plant when instructed to do so.

Requirement	*Verification*
The LED light ring must be able to turn on and off when instructed by the processor for the calculated amount of time from our algorithm	To test that the LED ring light turns on and off for a specific amount of time: 1. Power on the processor then upload and run the algorithm 2. Code the algorithm to turn the LED ring light on for a specific amount of time 3. Manually get data on the amount of time the LED ring light turned on using a stopwatch 4. Verify the manually collected time and the coded time are within ±1%

2.2.5 Power Unit System

The power unit system will consist of three main components which include the battery/USB-C charging unit, the voltage regulator and the voltage boost. The battery/USB-C charging unit will consist of 3.7 volt lithium ion batteries which will be connected to a charging circuit. The charging circuit can provide the user the ability to charge using a USB-C cable. The goal of the charging unit is to make our system as versatile as possible by providing the user the capability to run on batteries and charge the batteries when needed. Furthermore, the charging unit will output a constant 4.4 volts which will go through a voltage boost circuit to bring the voltage up to 5 volts. After the voltage boost, we will provide all the components in our system that require 5 volts to operate the voltage output from the voltage boost circuit. On the other hand, we will also take the constant 4.4 volts and connect it to a voltage regulator circuit which will bring the voltage down to 3.3 volts. After the voltage has been reduced, we will provide all the components in our system that require 3.3 volts to operate the output from the voltage regulator circuit. The power unit is the most important component in our design as it will be needed by all our other subsystems to operate. The power unit will also require some precise work as we do not want to provide high voltages to any of our components. In summary, the power unit will use 3.7 volt lithium ion batteries with a charging circuit, voltage boost circuit and voltage regulator circuit to provide all the components with their appropriate operating voltages.

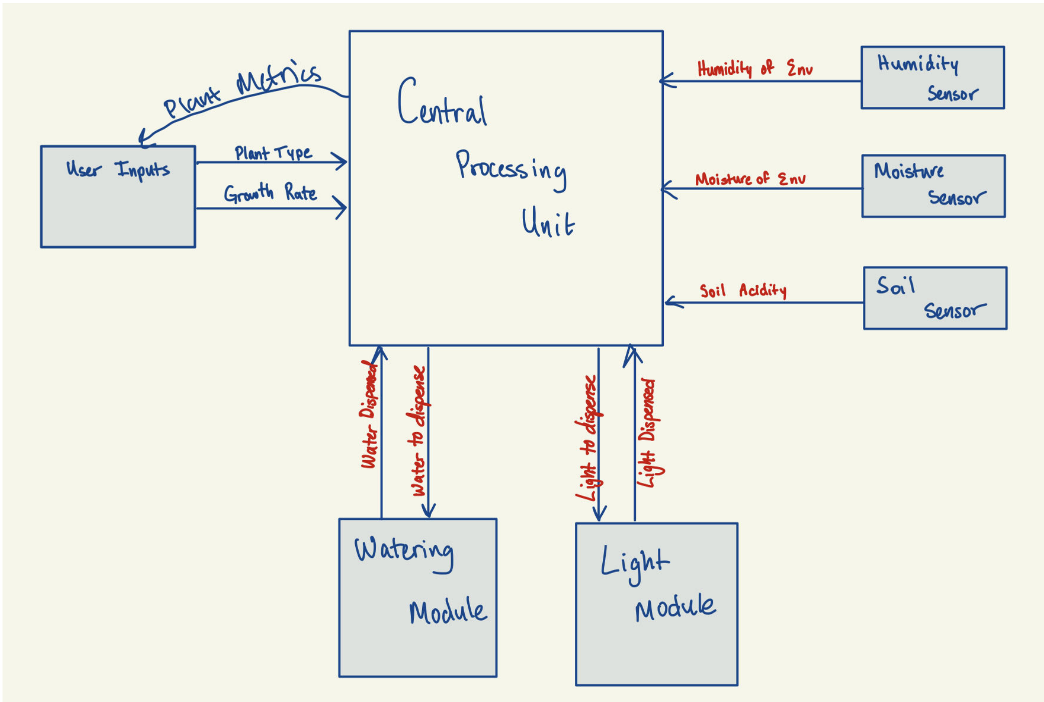
Requirement	*Verification*
The power unit should be able to provide 5 volts to components that require 5 volts operating voltage	To test that the power unit is providing 5 volts to the component that require 5 volts accurately: 1. Power on the Power Unit and ensure that the 3.7 volt lithium ion batteries are functioning 2. Verify that the voltage coming out of the voltage boost circuit is 5 volts with a ±0.1 volt tolerance using a voltmeter
The power unit should be able to provide 3.3 volts to components that require 3.3 volts operating voltage	To test that the power unit is providing 3.3 volts to the component that require 3.3 volts accurately: 1. Power on the Power Unit and ensure that the 3.7 volt lithium ion batteries are functioning 2. Verify that the voltage coming out of the voltage boost circuit is 3.3 volts with a ±0.1 volt tolerance using a voltmeter

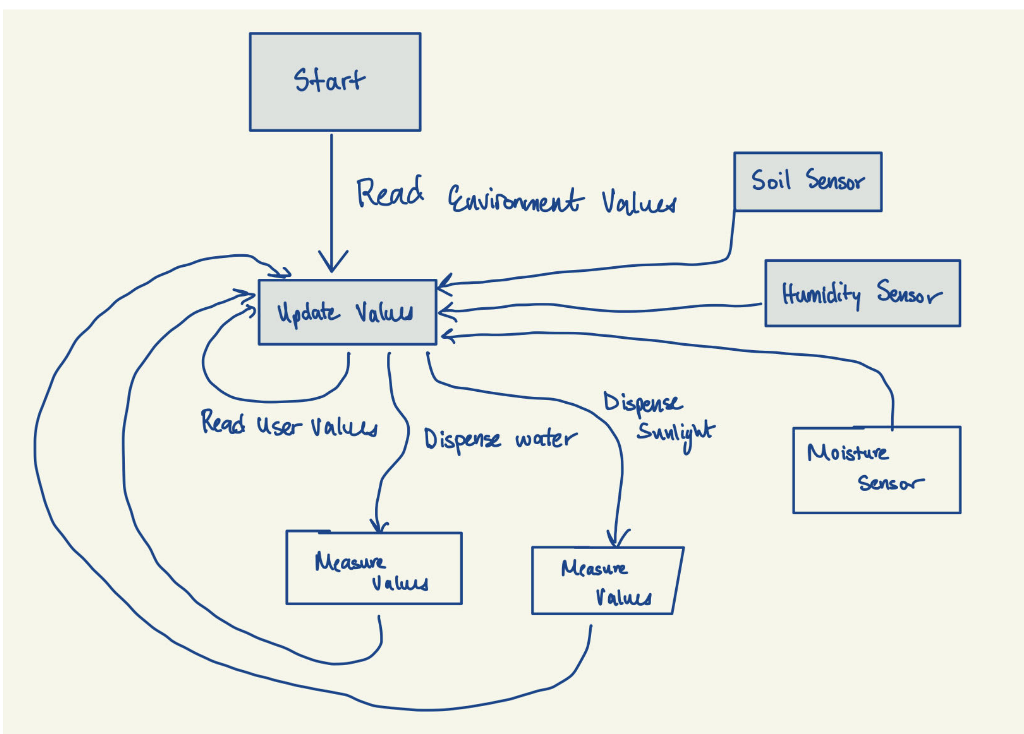
2.2.6 Phone App

The phone app will be used to provide the user a graphic interface to see critical sensor information and trigger the watering system and lighting system manually. The app will be created using a native mobile framework like React Native or Flutter. Additionally, the app will connect to a backend service like AWS or Google Firebase where it will poll for the critical sensor information updated by our processor using its wifi capabilities. The goal of the app is to be able to poll for sensor data every time the user opens the app and convert the data into a meaningful graphical representation. The app will also be able to connect to our processor through bluetooth connectivity which will provide the user an option to trigger the watering system and lighting system manually using their phone. In summary, the phone app will be created to provide the user with meaningful information about their plant and provide some manual capabilities to our smart system.

Requirement	*Verification*
The phone app should be able to show users sensor information and provide them with critical and insightful data about the health of their plants.	To test that the phone app is conveying accurate and up-to-date information to the user: 1. Launch the phone app and connect to the device using Bluetooth. 2. Attempt to trigger the watering and lighting systems through the app. 3. Using Wifi, we should be able to see a graphical representation regarding plant and soil health. 4. Ensure that the physical sensor values are corresponding to the values being displayed on the app, with an accurate timestamp of when the data was collected.

2.3 Algorithm Flow Chart





2.4 Tolerance Analysis

In order to determine potential pitfalls in our system, an important place to look is within the power system. We plan on using 3.7V lithium ion batteries, to provide voltages of both 3.3V and 5V either through a voltage boost or step down. However, how can we estimate the rate of decay of the batteries we use, and the rate of charge that must occur for the system to be self-sufficient in powering itself, but also having a battery component. We must consider all the elements being powered, and how their usage, voltage, and amperage play into the Power consumed and its effect on battery decay. Generally, Lithium ion batteries start at a voltage of 4.2V, and then drop down to 3.7V for the majority of their lifespan. When the battery's voltage is 3.4V, it's considered dead. To determine the timeline for decay, we can consider that our batteries have a lifetime of 5200 AH, or 5.2 amp hours. Consider the DH22 Temp & Moisture sensor. Its maximum operating amperage is 28 mA. The Grove soil moisture sensor has a maximum operating amperage at 35 mA. The Grove digital light sensor has a maximum operating amperage at 64 mA. Now, looking at the water pump, it operates at a maximum wattage of 3 W, at a voltage of 5V, giving us an operating amperage of 600 mA. Finally, the artificial light unit will use 60 mA per LED - assuming we power 10 LEDs, this will be about 600 mA for the entire unit. In sum, our components will require 1327 mA hours, meaning our batteries will decay after about **3.8 hours of use**. This is somewhat concerning - we may have to consider a larger battery that's able to provide the required voltage, but with higher mAH. For example, a 200 mAH battery that can also provide 5V and 3.3V would last about 146 hours before needing to be charged, which gives the charging unit some more leeway in powering the system. Another point of tolerance

2.5 Cost Analysis

The below figure illustrates that the total price before shipping and taxes is $107.73. Assuming shipping and taxes adds 10% to this cost, this adds another $10.77. The average salary for Computer Engineering graduates from UIUC is around $105,000 [6]. Using this information, our hourly rate comes out to about $54.68. In out project, we can expect a salary of $54.68/hour * 2.5 hours a day * 60 days = $8202 per group member. In total, salaries for members will be about $24606 in labor cost. With labor and parts, the total cost of our project would be **$24724.50.**

Description	Manufacturer	Quantity	Total Price	Link
DH22 Temperature and Humidity Sensor	Adafruit	1	$4.50	Link
Stemma Soil Sensor	Adafruit	1	$7.50	Link
Grove - Digital Light Sensor	Grove	1	$6.60	Link
DC Water Pump	Gikfun	1 (3 Pack)	$13.48	Link
Ring Lamp Light 24 Bits	Adafruit	1	$9.59	Link
ESP32 Processor	Espressif Systems	1	$3.69	Link
USB to Uart	Amazon	1	$8.29	Link
USB Charging IC	Adafruit	1	$6.95	Link
3.7 V Lithium Ion Battery	Adafruit	1	$29.95	Link
0.1uF Capacitors	Adafruit	1 (10 Pack)	$1.95	Link
Plastic Water Container	Pioneer Plastics	1	$8.82	Link

10K Resistor	Adafruit	1 (25 pack)	$0.75	Link
3.3 V Regulator	SparkFun	1	$2.10	Link
5 V Boost	Adafruit	1	$3.56	Link

2.6 Schedule

Dates	Tilak	Rohan	Yash
Week of 9/26	• Finish Design Document & start PCB Design	• Finish Design Document & start PCB Design	• Finish Design Document & start PCB Design
Week of 10/03	• Finish PCB design and seek instructor approval	• Consult with TA and start ordering initial parts	• Finish PCB design and seek instructor approval
Week of 10/10	• Start working on code	• Order any remaining parts • Start working on code	• Order any remaining parts
Week of 10/17	• Continue working on code • If PCB and parts come in, start assembling parts and soldering PCB	• Continue working on code	• Continue working on code • If PCB and parts come in, start assembling parts and soldering PCB
Week of 10/24	• Assemble any remaining parts	• Start initial code testing	• Start initial code testing
Week of 10/31	• Start basic testing of other subsystems • Order PCB if necessary	• Continue refining code	• Continue refining code and start basic testing of other

Dates	Tilak	Rohan	Yash
			subsystems ● Order PCB if necessary
Week of 11/07	● Continue to test and refine all subsystems and ensure they work together ● Make any small changes necessary	● Continue to test and refine all subsystems and ensure they work together ● Make any small changes necessary	● Continue to test and refine all subsystems and ensure they work together ● Make any small changes necessary
Week of 11/14	● Prepare for and give Mock Demo ● Make minor adjustments wherever possible	● Prepare for and give Mock Demo ● Make minor adjustments wherever possible	● Prepare for and give Mock Demo ● Make minor adjustments wherever possible
Week of 11/21	● Fall Break	● Fall Break	● Fall Break
Week of 11/28	● Prepare for and give Final Demo	● Prepare for and give Final Demo	● Prepare for and give Final Demo
Week of 12/05	● Prepare for and give Final Presentations ● Work on and submit Final Paper	● Prepare for and give Final Presentations ● Work on and submit Final Paper	● Prepare for and give Final Presentations ● Work on and submit Final Paper

3 Ethics & Safety

When considering ethics and safety of the product, we must consider potential pitfalls and safety concerns in the products being placed in an individual's living space. We must consider the components that exist in the system - primarily the microcontroller, sensors, and watering/sunlight mechanisms. Although there isn't a huge safety concern at first glance, potentially malfunctioning or overheating of the microcontroller or sunlight systems may impose a risk of fire depending on where the device is located. In order to "avoid harm", we plan to ensure that elements are properly insulated, and don't impose a risk of mal-action in the case of misuse or malfunction.

On the note of ethics, our system simply attempts to aid families and plant owners maintain plants in a more convenient manner - this doesn't pose an ethical risk in regards to a broad objective. In accordance with the IEEE Code of Ethics Section 7.8.I.1, we will strive to implement an ethical design and follow all of the sustainable practices possible, while not endangering the public or the environment [4]. Additionally as stated in section 7.8.I.2, we plan to improve society's understanding and educate them of our project's capabilities, all while doing so in a safe and ethical manner [4]. As per section 7.8.I.5 , we plan to accept constructive criticism regarding our work, and will correct any and all errors regarding our project and its overall design and implementation [4]. This feedback and criticism will be provided to us by our professors, TAs, and even amongst ourselves. Furthermore, we plan to only perform tasks for which we are qualified for after the adequate technical training, as stated in section 7.8.I.6 [4]. To fulfill this, we are required to perform activities such as the 'Lab Safety Training', 'CAD Assignment', and 'Soldering Assignment' so that we are fully aware and prepared to use tools such as CAD to assist with building PCBs, and implement proper soldering practices to ensure no one is harmed. Finally, according to section 7.8.III.10, we will make sure that as a group, all three of us will check in on each other to make sure that we uphold the IEEE Code of Ethics at all times and encourage ethical and safe behavior and practices at all times [4].

4 References

[1] L. Ada, “Adafruit Stemma soil sensor - I2C capacitive moisture sensor,” *Adafruit Learning System*, 2018. [Online]. Available: https://learn.adafruit.com/adafruit-stemma-soil-sensor-i2c-capacitive-moisture-sensor/pinouts. [Accessed: 25-Sep-2022].

[2] L. Ada, “DHT11, dht22 and AM2302 sensors,” *Adafruit Learning System*, 2012. [Online]. Available: https://learn.adafruit.com/dht. [Accessed: 25-Sep-2022].

[3] Espressif Systems, “ESP32WROOM32E ESP32WROOM32UE - Espressif,” 2022. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf. [Accessed: 28-Sep-2022].

[4] IEEE Board of Directors. “IEEE Code of Ethics.” IEEE Code of Policies, Section 7 - Professional Activities (Part A - IEEE Policies). [Online]. June 2020. https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed 15-Sep-2022].

[5] L. Ada, “TSL2561 luminosity sensor,” *Adafruit Learning System*, 2012. [Online]. Available: https://learn.adafruit.com/tsl2561/downloads. [Accessed: 27-Sep-2022].

[6] Grainger Engineering Office of Marketing and Communications, “Computer Engineering,” *The Grainger College of Engineering | UIUC*. [Online]. Available: https://grainger.illinois.edu/academics/undergraduate/majors-and-minors/computer-engineering. [Accessed: 28-Sep-2022].