Plant Health Monitor

ECE 445 Design Document

Dishen Majithia, Esteban Roberts, and Tommy Bahary
Team 51
TA: Haoqing Zhu

1. Introduction

1.1 Objective:

The main issue with growing plants at home is monitoring their health. Plants regularly dry out or even die if not given the proper attention. Many people do not know the proper way to care for a plant and this project attempts to remedy this problem by giving the grower precise measurements about the plant's health and automating repetitive tasks.

We are planning to make a multi sensor device that can accurately measure various soil health indicators in real time, so the user can get live feedback and can adjust the conditions accordingly. The device will also allow watering of the plant automatically when needed. This would provide the user with a way to conveniently check on and maintain their plants even when they are not around.

A similar product exists on the market but doesn't measure pH and only works with bluetooth. Our project differentiates itself by transmitting data remotely with wi-fi so that the data is displayed on a website. This data would provide the user with a way to conveniently check on their plants even when they are not around, as opposed to bluetooth which only works when the device is nearby. Our automatic watering system also allows the user to leave the plants when they are traveling or away from home for an extended period of time.

1.2 Background:

Growing plants at home is a hobby that many people enjoy but maintaining the health of the plants is not always easy. The main problem with plant health usually comes in the form of the soil. This includes bad pH levels and under/over watering the soil. Temperature and insufficient lighting can also be a factor. Our project would remedy this by monitoring each of these conditions. This way people do not have to do their own research into the more difficult aspect of plant growing, soil health. An additional concern for houseplants is travel. Being away from home for extended periods can leave plants dry and withered from the lack of watering and basic care. This project also attempts to remedy that by having an auto watering system and allowing the user to monitor the plant from anywhere.

1.3 High-level Requirements:

- Must get information from the various sensor modules and relay information to the user through Wifi and status lights at least every 5 minutes.
- This project must be able to automatically adjust the moisture level of the soil to achieve a target level within ±10% in 1 hour.
- Must be able to freely switch the power source from an outlet to a battery that can individually provide at least 3-5 days of functionality.

2. Design

The project begins with the power supply. This takes in power from either the Lithium Polymer battery or the outlet. The outlet outputs 120V and is transformed to 5V by the power adapter, which can be any kind usb adapter. The battery charging IC then outputs the battery charging voltage, near 3.7V. The battery protection IC protects the battery from damage occurring from under/overvoltage or under/overcurrent. From there 3.7V is connected to the linear regulator, the lithium polymer battery and the water pump driver. The driver turns the water pump on and off when given the signal from the MCU. The linear regulator in turn, converts the varying battery voltage into the constant 3.3V to power the MCU and all of the sensors. The MCU is connected to the sensors and both receives and processes the information that the sensors output. After receiving the information from the sensors, the MCU sends the signal to the driver to water the plant if necessary, and then sends all the relevant information to the web server by first communicating with the local host and then sending the HTTP request.

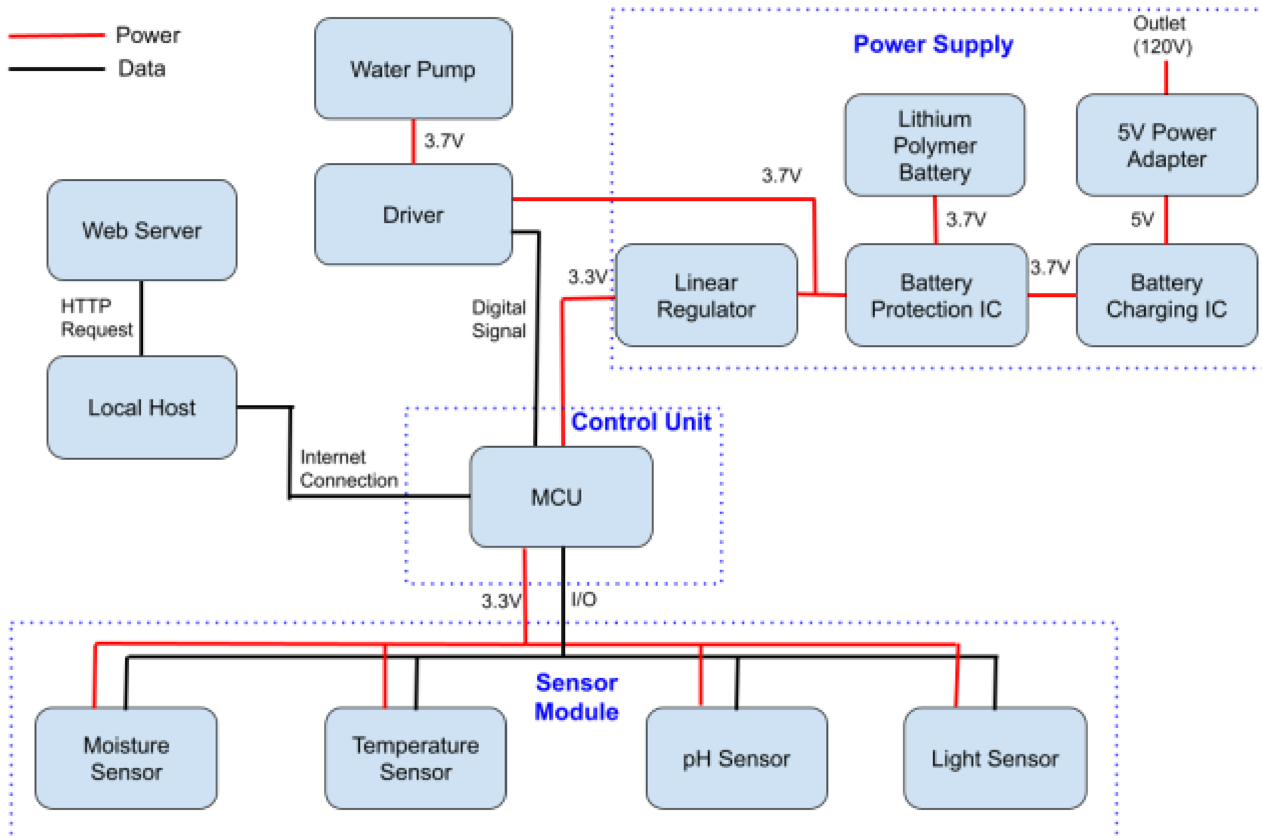

Fig.1. Block Diagram

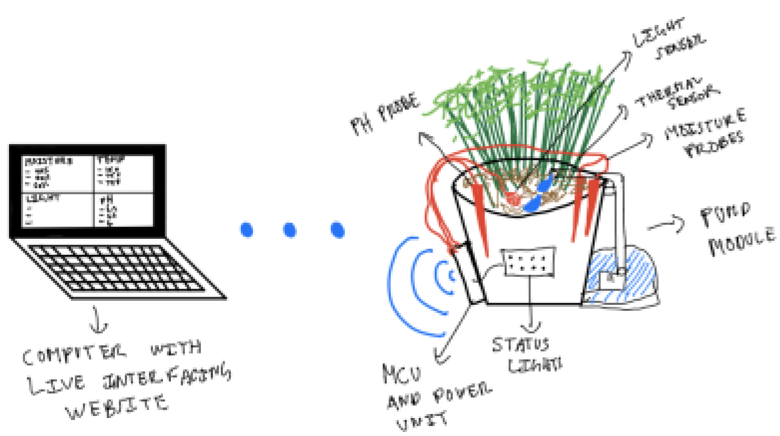

Fig.2. Physical design

2.1 Control Unit

The ESP32 S2[1][2][3] microcontroller will sample the sensors periodically and use calibrated algorithms to convert these measurements into meaningful data. This data will then be sent over the integrated wifi antenna to be published on a website where the data can be reviewed. The data will also be used to control several warning lights indicating plant health and battery level. This is critical for the first two high-level requirements because it processes and publishes raw data from the sensors as well as controls the pump.

An ESP32 S2 was chosen for several reasons. An ESP microcontroller was chosen because its options for integrated wifi and antennas are unparalleled in this price range. Additionally the simplicity of the integrated antenna and wifi in ESP32 modules allows us to avoid the intricacies of antenna design and configuration of external RF components. The ESP32 S2 is the cheapest of the ESP microcontrollers while still having comparable features and additional connectivity. Ultimately processing power was not a factor in this decision because many of the functions this MCU will perform are not computationally intensive and do not need to be repeated frequently.

Requirement	**Verification**
1. Must be able to transmit collected data over wiFi every 5 minutes in under 10 seconds.	1. a. Print to the console when data transmission over wifi begins, confirm change to website occurs within 10 seconds. b. Make any significant change to a sensor condition, change will be visible on the website in less than 5 minutes and 10 seconds.
2. Upon sensor reading, must be able to decide when values are not optimal and let the user know with the LEDs.	2. a. Connect a sensor ADC pin to a power supply, and sweep between 0-3.3V to simulate the entire range of sensor inputs. b. Using the console, take note of each data value upon sensor reading. c. When data is not within the programmed range for that sensor, the corresponding light will turn on. d. Repeat for each sensor.
3. Must be able to take readings from each of the sensors at least every 5 minutes.	3. a. Print sensor data to console to make sure that the values are changing over time. b. Take time between readings to prove that the readings are output every 300 seconds +/- 10.

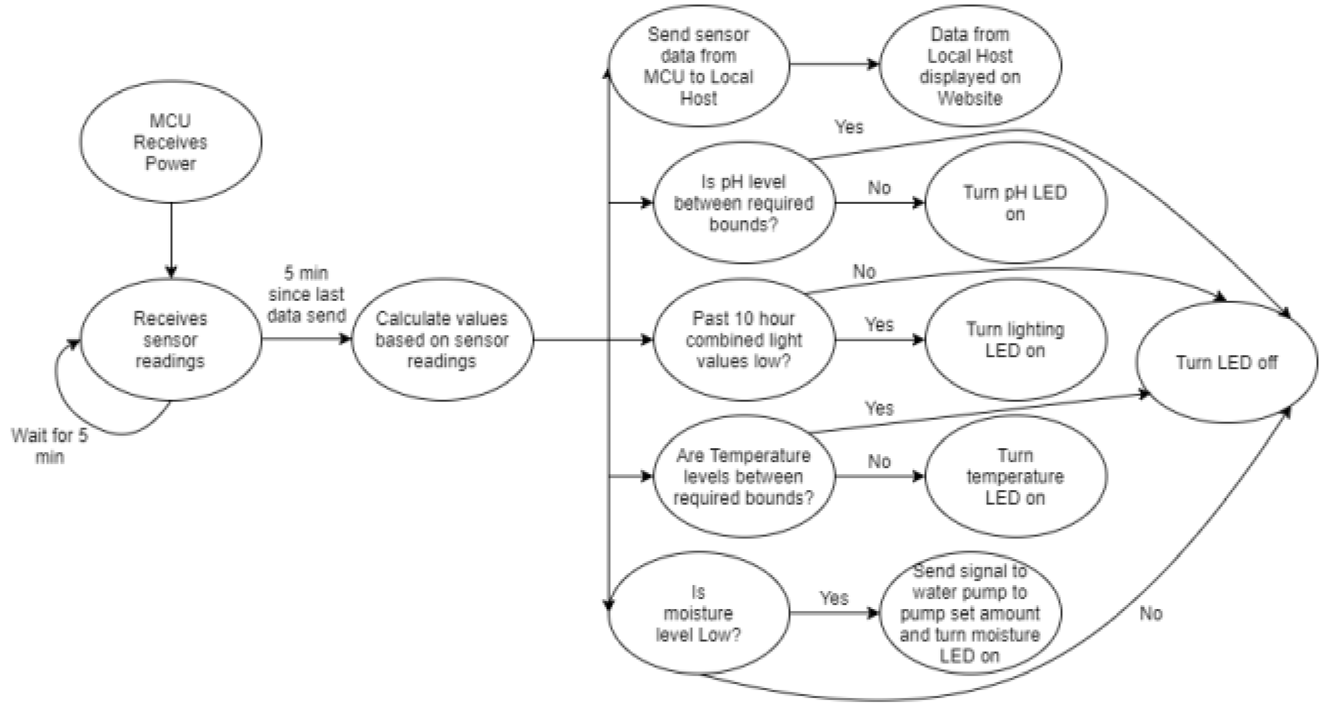

Fig.3. MCU flow chart

The flow chart in figure 3 shows how the microcontroller interfaces with the sensors and the website. We have set the sensor reading call time to be once every 5 minutes, since that was the time that allowed for longer usage of our battery without compromising on the availability of recent data. We have also shown the condition for the MCU to interface with the pump and turn it on, since that only needs to be done in the cases when the moisture level is too low. In these cases, the pump will be turned on for 10 seconds, and then the MCU will wait until the next sensor reading. This also allows us to make sure that the water has time to seep through the soil before a new reading is taken so that the soil doesn’t get overwatered.

We also have similar checks for each of the sensors and also the power unit for battery level. These checks will also be done every 5 minutes when we get the readings from the sensor. These will basically just be comparisons to preset thresholds for each one. These thresholds will define acceptable values for each sensor, and if the actual readings aren’t in that range, a small LED is turned on. One exception is the lighting LED, which will take in the current light value plus the light values of the past 10 hours. This is done so that the LED does not turn on only because it is nighttime. These LEDs would basically serve to tell the user to check the website since it might not be convenient for them to constantly keep an eye on it.

When the MCU processes the sensor readings every 5 minutes, it will transfer the data to the website. This will happen regardless of if any of the LEDs have to be turned on or if the water pump has to pump water.

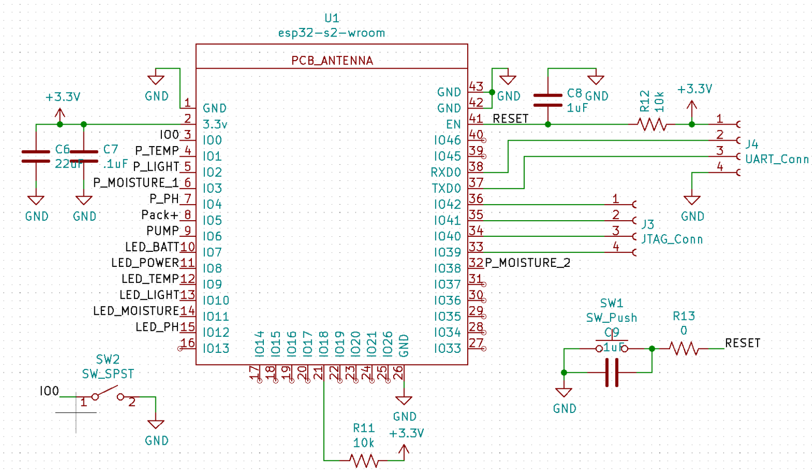

Fig.4. Schematic of Control Unit [4], [5]

2.2 Power Supply

The power supply will consist of several components. The first would be an off the shelf USB power brick to convert from 120V AC to 5V DC. A micro USB cable will plug into the PCB, and the power from this cable will be used in conjunction with a battery charging IC to charge the small lithium polymer battery. A linear regulator will also be used to regulate the voltage going to the rest of the system to 3.3V, and an ADC from the microcontroller will be used to measure battery voltage in order to report charge level. This reading will be used to approximate the battery charge percentage, which will be reported back to the user through the website and an onboard status light. The power supply is important for the third high-level requirement because it allows the system to function both plugged in and with a battery. It also provides power that is needed for the first two high-level requirements.

2.2.1 5V USB Adaptor

The first stage of the power system will consist of an off the shelf USB adaptor. This was chosen because these adaptors are standardized and common. This provides several benefits. The first of which is that the input is standardized and rated to 5V +/-5%. Additionally these are very common adaptors because the majority of electronics chargers utilize them, which allows them to be acquired inexpensively. This would also allow the user to charge the device more conveniently as they likely already have a micro USB charger available. Using an off the shelf converter also allows us to ensure the safety of the device. Most of these adaptors are electrically isolated from 120V AC and certified for safety so we can be sure that electronics further in the circuit aren't exposed to high voltage.

2.2.2 Battery charging IC

The TI BQ24050 [6] was chosen for the battery charge controller for several reasons. The first of which is because it's designed to be used in conjunction with a USB port. The data pins of the USB can be connected to this IC to determine the type of USB connection and adjust the charge current accordingly. For example, the current draw will be limited to 100mA if it is connected to a USB port, but can go to 1A if an adaptor is detected. It also allows the battery to be charged even while power is being drawn, a feature which is particularly important for allowing the device to work while plugged in and during initial stages of battery charging. Another added benefit is the CHG' pin, which will allow the connection of a LED to indicate charging. This will aid in debugging and increase user friendliness.

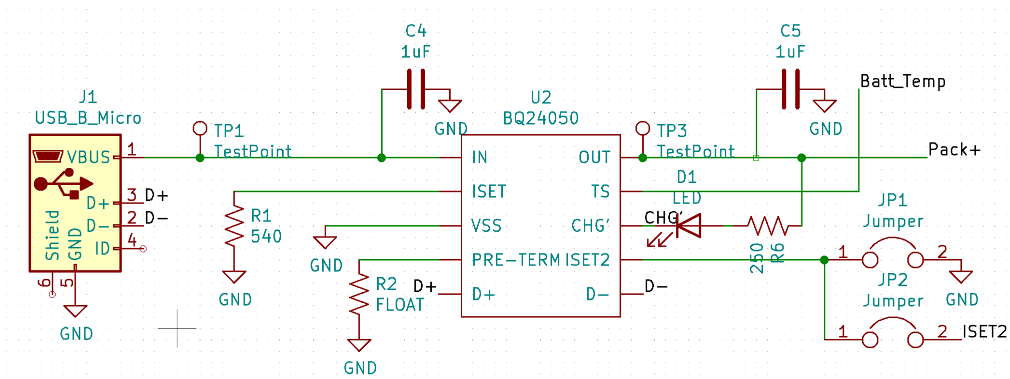

Fig. 5. Battery Charging IC Schematic

Requirement	Verification
1. Must be able to output at least a combined 750 mA for battery charging and circuit operation.	1. a. Connect circuit to battery and measure current to battery with a series ammeter. b. Connect a parallel load to the battery. Use an ammeter to measure current to this parallel load and esure these currents sum to be greater than 750mA. c. Battery must be sufficiently charged while these tests are performed, to ensure that the battery charger is outputting current for fast charging.
2. Battery must be able to source deficit current when load exceeds 750 mA up to a max of 1.5A, without voltage dipping below 3.3V	2. a. Connect a varying load to the output of the battery circuit while charging until an ammeter shows the load to be drawing 1.5A. b. Measure the voltage across the load to ensure that it is above 3.3V

Note: The battery will never be operated without its protection circuit connected and verified as specified in 2.2.3.

2.2.3 Battery protection IC

The TI BQ29700 [6] battery protection IC was chosen for the purposes of protecting the battery from damage. The chosen IC can protect against overcurrent during charging and discharge, as well as overvoltage and undervoltage. One of the most important functions of this IC is the undervoltage protection. Under normal conditions overvoltage and overcurrent will only occur during a catastrophic failure of another component, which could be mitigated by the inclusion of a fuse. Undervoltage however could occur during normal operation, because the circuit will continue to function as normal and continue to consume charge even when the battery is operating at a dangerously low voltage. With a battery protection IC, the battery will be disconnected to prevent it from being discharged further than a safe level. The BQ29700 was specifically chosen for several reasons. The most important of which is its simplicity. Only a few components are used in conjunction with this circuit, therefore the circuit will be less prone to failure and design mistakes that will compromise the safety of the battery. Additionally the simplicity of the design will significantly lower costs and allow the circuit to be tested in isolation without the battery. It's important to note that these protections are already built into the chosen battery pack. Despite this, redundancy is important and well worth the cost, especially in this case when the cost is so low and there's potential for a disastrous failure.

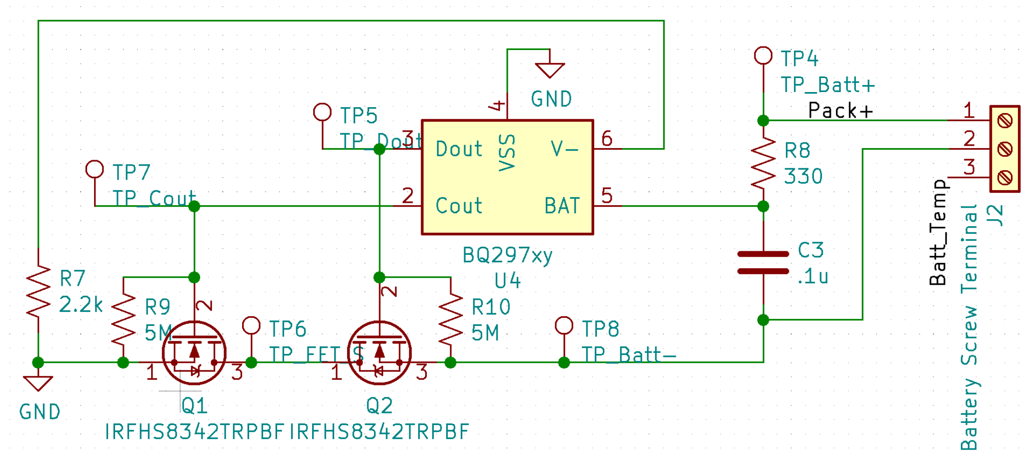

Fig. 6. Battery Protection IC Schematic

Requirement	Verification
1. The battery must be disconnected and charging terminated when charge voltage exceeds 4.3V.	1. a. Use a power supply to slowly raise the voltage across the BAT pin starting from 3.5V. b. When voltage exceeds 4.3V, MOSFET with gate connected to COUT will turn off. This will be tested by using a series ammeter to confirm that the no current flows through the MOSFET past this point. c. The ammeter will also be used to confirm that no current flows through the battery past this point.
2. The battery must be disconnected and discharging terminated when charge voltage dips below 2.8V.	2. a. Use a power supply to slowly lower the voltage across the BAT pin starting from 4V while grounding the V- pin. b. When the voltage dips below 2.8V the MOSFET with gate connected to DOUT will turn off. This will be tested

	by using a series ammeter to confirm no current flows through the MOSFET past this point. c. The ammeter will also be used to confirm that no current flows through the battery past this point.
3. The battery must be disconnected and charging terminated when charge current exceeds 2A.	3. a. Use a power supply to apply 3.7V across the input terminals. b. Apply a load where the battery would be in order to simulate battery charging, while using an ammeter to measure current. c. Increase the load until the current surpasses 2A, at which point the CHG MOSFET will be turned off. This will be tested by using a series ammeter to confirm no current flows through the MOSFET past this point. d. The ammeter will also be used to confirm that no current flows through the battery past this point.
4. The battery must be disconnected and discharging terminated when discharge current exceeds 2A.	4. a. Use a power supply to apply 3.7V across the terminals where the battery would be connected. b. Apply a load where the protection circuit would connect to the rest of the system in order to simulate battery discharging, while using an ammeter to measure current. c. Increase the load until the current surpasses 2A, at which point the DSG MOSFET will be turned off. This will be tested by using a series ammeter to confirm no current flows through the MOSFET past this point. d. The ammeter will also be used to confirm that no current flows through the battery past this point.

Note: All verification steps will be completed without the use of a battery. Instead resistive loads and power supplies will be used to simulate a battery in the test circuit. If a real battery was used, failure of these verification checks could result in catastrophic failure of the battery.

2.2.4 Lithium Polymer Battery

A lithium based battery was chosen in this application because of its high energy density. A larger battery type would be safer, easier to charge and maintain, however maintaining a small module size is paramount for the practicality of the device. A larger battery would make the device cumbersome and large, which would be less convenient than simply leaving the device plugged in. The required capacity of this battery is calculated in section 2.4. The max current draw is calculated below, in equation 1. This uses the pump power usage [12] and max MCU current consumption during 802.11n wifi transmit [1].

$$I_{pump} = 3W/2.8V = 1071\ mA$$
$$I_{total_{max}} = I_{pump} + I_{MCU} = 1071\ mA + 200\ mA = 1271\ mA$$

Equation 1

This is a calculation of the max current that could ever be drawn from the battery, using the MCU current in the highest power mode and the pump at full power. The chosen MIKROE-112 battery [7] can output a max of 2000 mA, which is more than enough. The charge speed is a less important factor, although the battery should charge significantly faster than it discharges in typical operating conditions. For this calculation in equation 2, we will once again utilize the results of section 2.4.

$$I_{Avg} = \frac{Capacity\ needed\ (mAH)}{Hours} = \frac{1300\ mAH}{5\ days*24\ H} = 10.83\ mA$$

Equation 2

This far exceeds any reasonable charging speed, as the battery charging IC is rated for 1A and the battery is rated for 2A.

Requirement	**Verification**
1. Battery can charge with a current of at least 100 mA.	1. a. Provide charging circuit with power, and ensure that the battery has been plugged in long enough to initiate fast charging. b. Use a series ammeter to measure the current into the battery.
2. Battery can discharge with a current of at least 1500 mA.	2. a. Leave the charging circuit unpowered so that all output current comes from the battery. b. Use a series ammeter to adjust the load until a current of 1500 mA is reached.
3. Battery can power the entire system for at least 3 days on a full charge.	3. a. Plug the entire system into USB adaptor until charging light turns off, indicating a full charge. b. Unplug system and leave operational

	for 3 days. c. Ensure the microcontroller is still powered by using a voltmeter to measure voltage across 3.3V pin.

Note: The battery will never be operated without its protection circuit connected and verified as specified in 2.2.3.

2.2.5 Linear Regulator

The LP38501-ADJ linear regulator [8] serves to convert the varying battery voltage into a constant 3.3V for use by the ESP32 S2 microcontroller. This was chosen over a switching regulator for several reasons. The most important of which is noise. A switching regulator can create considerable noise which may interfere with the wifi antenna on the board. A switching regulator would provide few benefits for this tradeoff, as the efficiency of the linear regulator will remain above 75% even in the worst cases. In such a scenario, the regulator converts from the max battery voltage of 4.2V to 3.3V, yielding 78.6% efficiency. The inclusion of a switching regulator would also increase cost and complexity for this mild increase in efficiency.

This regulator is only powering the MCU, and its max current requirement will reflect this. The peak possible current usage of the ESP32 S2 WROOM module is 310mA [3], so a current target of 500mA will easily power this with room for error.

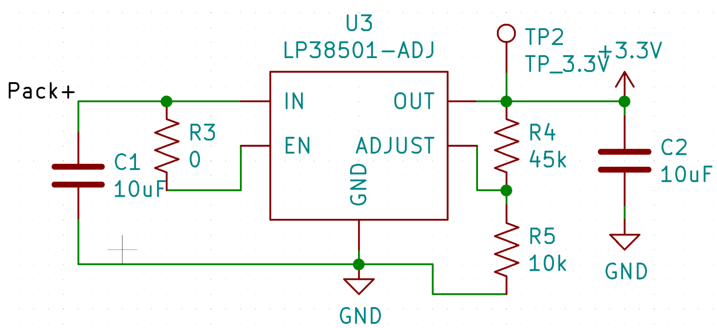

Fig. 7. Linear Regulator Circuit Schematic

Requirement	Verification
1. Linear regulator can convert an input voltage ranging from 3.3-4.2V to an output voltage of 3.3V +/- .1V at output currents up to 500mA.	1. a. Connect input of regulator circuit to a power supply. Vary supply voltage between 3.3 and 4.2V. b. Connect load to output of regulator and adjust until series ammeter current is 500mA. c. Measure the load voltage at this point to ensure it meets requirement 1.

2.3 Sensors

The project has four sensors in total: moisture, light, temperature, and pH. Out of these, the pH and moisture are pressed into the soil, while the light and temperature sensor can just be placed on top of the soil or around the pot. These sensors will all connect back to the MCU and send the respective measurements, which will then be displayed onto the website.

2.3.1 Moisture sensor

The moisture sensor is going to be made from scratch, and will consist of two metal probes that we press into the soil about one centimeter apart from each other. A measurement of the capacitive response of the probes through the soil will give us an idea of the amount of moisture in the soil.

Requirement	Verification
1. Measurements of similarly prepared samples must be within 10% of each other.	1. a. Place the sensor probes in water and then just in the air to get the 'wet' and 'dry' measurements. We will now get values in between these two numbers when we place it in soil. b. Test the sensor measurements using multiple soil samples, each made with specific amounts of water, and ensuring we get consistent results. All readings should be within 10% even if the probe is reinserted.
2. Must be able to send values to the website within 10 seconds after the data is measured, and sample the soil once every 5 minutes.	2. a. Connect the sensor to the ESP32 with our circuit and write code to pull data from the sensor. Start with basic measurements and then test qualitatively with different soil samples. b. Test the time delay by checking for the time difference between changing values that we know. Ex - Switch the probe from soil pot to a jar of water and see how long it takes for the value to reflect this change.

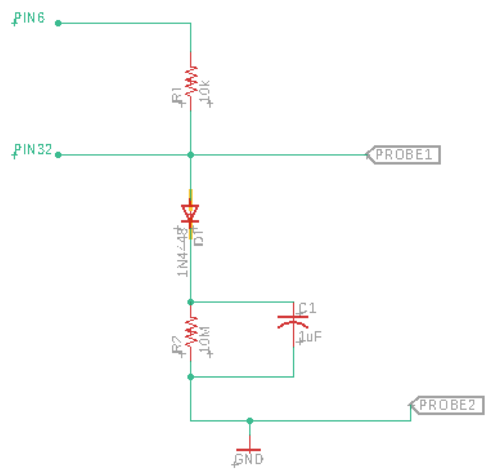

Fig.8. Moisture sensor schematic

2.3.2 Temperature Sensor

We are planning to use a DS18B20 [9] ambient temperature sensor with the circuit that we have designed to digitally measure the ambient air temperature around the plant. This is a good indication of the temperature of the soil and the moisture contained in it since for an indoor plant there are no sudden changes in the temperature of the air.

Requirement	Verification
1. Must be able to get accurate temperature values near the plant with 5%.	1. a. The tests for this sensor will be done using a room with a controlled temperature and changes in the temperature will be checked for consistency. This will be done by using a separate temperature gauge
2. Must be able to send values to the website within 30 seconds after the change, and check the values every 5 minutes.	2. a. Verify by running an experiment where we move the probe into a freezer and check the delay in measurement. The device is rated for -55 Celsius to 120 Celsius.[9]

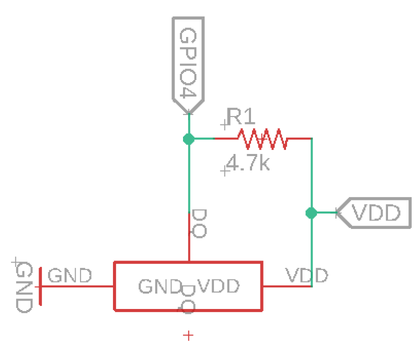

Fig. 9. Circuit Schematic for temperature sensor

2.3.3 pH Sensor

The pH sensor is going to be reverse engineered from an off the shelf version. We are planning to try and strip an existing one and redesign it's electrical components to work with our system rather than making our own pH sensor from scratch.

Requirement	Verification
1. Must be able to measure pH within 0.5 of the correct value.	1. a. We will compare the readings taken by the pH sensor to readings taken by a universal indicator on samples of varying pH. This will give us an idea of the accuracy of the probe.

2.3.4 Light Sensor

The light sensor will be made from scratch using a photoresistor connected with a resistor to make a voltage divider, and giving output to the ESP32 microcontroller. The resistance of the photoresistor varies with the amount of light on it, so we will be able to measure this resistance to get an idea of the luminescence around the plant.

Requirement	Verification
1. Measure the ambient light around the photoresistor and send data within 10 seconds of the change.	1. a. Run an experiment where we incrementally vary the amount of ambient light around the photodiode and check that the multimeter reading during those variances is affected by the correct increments. b. Check that the data shown on the website reflects a change consistent with the change in the multimeter reading and that it is updated within the required time frame.
2. Must have consistent readings within 10% of each other when exposed to the same light environments for 10 minute intervals.	2. a. We will be leaving the photoresistor under a constant lamp source in an otherwise dark room and record readings for 10 minutes. Then we will plot all the readings and ensure they are within the desired range.

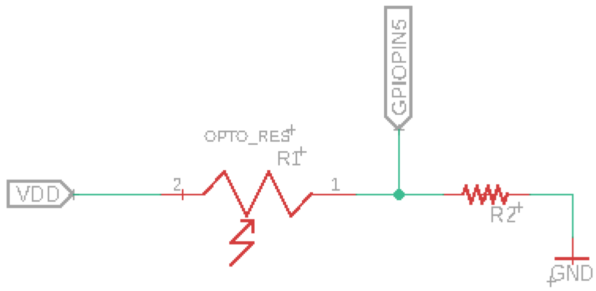

Fig. 10. Circuit Schematic for light sensor

2.4 Pump

The pump module will consist of a 3W submersible water pump and a MOSFET based logic circuit to control it. We will use the microcontroller to control when the pump is turned on and off. The pump will only be turned on for 10 seconds at a time, and will pump an incremental amount of water during that time. Then, the MCU will wait for the next testing cycle to decide whether the soil needs more water or not, and send the appropriate signal to the pump.

Requirement	**Verification**
1. Respond to on/off signals from the Microcontroller and be able to turn on for 10 second intervals.	1. a. We will set up a block of code to simply send a signal to the MOSFET for 10 seconds to turn on the pump, and check whether the pumping capacity outputs the correct amount of water. b. We will also do the same test with a block of code that sends this signal based on another GPIO pin input on the MCU, to check if it works with the logic of the entire system.

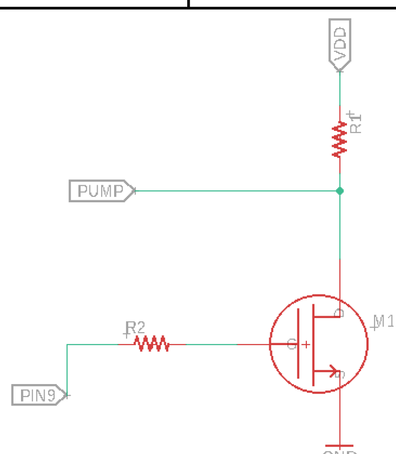

Fig. 11. Circuit Schematic for pump

2.5 Webpage

The web page will display the readings of the sensors and update whenever the MCU sends new values. We will use the wifi capabilities of the ESP32 to send the values from the sensors to the local host. The local host will then send the HTTP request to the web server[10]. Then the sensor values will be printed on the webpage.

Requirement	Verification
1. Properly display the correct values on the web page every 5 minutes.	1. a. Using the microcontroller, print out to console the values that it reads from the sensors and compare to those displayed on the website. b. Take the time between each value change to make sure that it is done in 300 seconds +/- 10.
2. The local host must successfully send an HTTP request to the web server when attempting to send the sensor values.	2. a. On the console, check if the HTTP response status code is one that indicates that the web server has successfully responded to the HTTP request[11].

2.6 Tolerance Analysis

One important tolerance to consider is the size of the battery pack and the estimated power usage of the device. One of the most important aspects of this project is the ability of the device to collect data constantly and report it to the user. It would be detrimental to the goals of this project if it could not operate without intervention for significant periods of time. Part of this includes operating for 3-5 days off of a fully charged battery, as is stated in the third high level requirement. Variations in the operation of the circuit and the requirements monitored could cause the average power consumption of the device to exceed a level that could be supplied by the chosen battery for the desired timeframe. For this reason the maximum average power consumption of each block will be calculated and compared to the battery's capacity.

The pump chosen is 3W, and can pump 200L per hour [12]. The daily target for pumping water is 1L, which is an estimated upper bound for the amount of water a plant would need. There is no datasheet available for this pump, so it will be assumed that the pump may pump at half speed while using 3W depending on the height difference between the basin and the plant. The battery capacity required to run this pump is estimated in equation 3.

$$6\,Wh/200\,L\ =.03\,Wh/day$$
$$.03\,Wh\,/\,3.3V\ =\ 9.09\,mAH/L\ =\ 9.09\,mAH/day$$
$$Capacity\ =\ 9.09\,mAH/day\ *\ 5\,days\ =\ 45.45\,mAH$$

Equation 3

The ESP32 S2 microcontroller will also be a significant sink of power, especially when communicating using wifi. For this reason, it will be programmed to only transmit data through wifi every 5 minutes, or when there is significant change in the data being measured. The datasheet specifies a current consumption of 200mA max for 802.11n wifi transmit. When the modem is sleeping power use is specified as 12 mA, and when in light sleep the MCU will use 450 uA [13]. It will be assumed that the entire signal consists of a transmit (higher power use) and lasts for 10 seconds every 5 minutes. These 5 minutes will also include 10 seconds of modem-sleep mode to collect measurements, and 280 seconds of light sleep. These values are used in another battery estimate which is shown in equation 4.

$$I_{avg} = 200\frac{10}{300} + 12\frac{10}{300} + .45\frac{280}{300} = 7.486\,mA$$
$$Capacity = 7.486mA * 5\,days * 24\,hours = 898.4\,mAH$$

Equation 4

Finally, we can sum these two values to get an estimate of the power used. This is done in equation 5. The efficiency of the regulator does not need to be considered because the current output from the regulator will be equal to the current input.

$$Capacity_{min} = 898.4 + 45.45 = 943.85\,mAH$$

Equation 5

It's incredibly important to note the decision to report data every 5 minutes and the tradeoff that is being made. In this case, the MCU accounts for nearly 900 mAH of the required capacity which is the overwhelming majority. Increasing the frequency of reporting would increase this number by several orders of magnitude. It was determined that reporting at frequencies greater than 5 minutes would be excessive given the measurements being taken will not change quickly. That being said, it is still important to be providing the user with current data and the addition of a battery in excess of 1000 mAH will not significantly increase the price, size or complexity of the product.

The use of a 2000 mAH battery would guarantee that the device could function for 5 days, and is a commonly used battery size which will give us more choices for the battery. In this analysis, it was shown that the most power that could reasonably be used in the given amount of time, with tolerances included, is near 1000 mAH. This helps us ensure that we meet or even surpass our third high level requirement.

3. Cost and Schedule

3.1 Cost Analysis

- Labor
  - $47/hr x 2.5 x 100hrs = $11750 (Esteban-Computer Engineer)
  - $39/hr x 2.5 x 100hrs = $9750 (Dishen/Tommy-Electrical Engineer)
  - $11750 + $9750 * 2 = $31250
- Parts

Function	Manufacturer	Part Number	Quantity	Unit Price
MCU	Espressif systems	ESP32 S2 Wroom	1	1.99
Battery Charger IC	Texas Instruments	BQ24050DSQR	1	1.29
Battery Protection IC	Texas Instruments	BQ29700DSER	1	0.56
Linear Regulator	Texas Instruments	LP38501TSX-ADJ/NOPB	1	2.36
Lithium Polymer Battery	MikroE	MIKROE-1120	1	12.5
Thermistor for battery	Semitec	103AT-2	1	2.25

Submersible water pump	Driew	BW003-MB009	1	19.49
Tube for pump	Uxcell	B085N1GYSV	1	6.99
Photoresistor	Adafruit	485-161	3	0.95
Temp sensor	Maxim Integrated	700-DS18B20+	1	5.27
pH sensor	Vivosun	B01DNUIGUY	1	10.9
USB to UART converter	HiLetgo	CP2102	1	5.19
Micro USB Type AB Connector	Molex	47589-1001	1	0.92
Battery connector	JST	B2B-XH-A(LF)(SN)	1	0.15
Terminal block x2	CUI Devices	TB007-508-02BE	11	0.249
SPDT switch	Nidec Copal	MFS101D-10-Z	1	1.03
Reset Button	C&K	PTS645SK50JSMTR92 LFS	1	0.41
Green LED	Vishay	TLVP4200	2	0.38
Red LED	Vishay	TLHK42T1U2	5	0.46
MOSFET	ON Semiconductor	512-FDPF045N10A	1	3.44
MOSFET	Infineon	IRFHS8342TRPBF	2	0.51
Diode	ON Semiconductor	1N4448	2	0.1
549Ω Resistor 1206	Bourns	CR1206-FX-5490ELF	1	0.1
250Ω Resistor 1206	ARCOL / Ohmite	APC1206B250RN	7	0.18
5MΩ Resistor 1206	Susumu	RG3216P-5004-B-T1	2	0.57
330Ω Resistor 1206	Vishay / Dale	CRCW1206330RJNEAC	1	0.19
2.2kΩ Resistor 1206	Yageo	RC1206FR-102K2L	1	0.23
44.2kΩ Resistor 1206	Bourns	CR1206-FX-4422ELF	1	0.12
10kΩ Resistor 1206	Vishay / Dale	RCS120610K0FKEA	2	0.33
1uF Capacitor 10V 1206	Vishay / Vitramon	VJ1206Y105JXQTW1BC	5	0.19
.1uF Capacitor 10V 1206	Vishay / Vitramon	VJ1206Y104JXQCW1BC	7	0.22
10uF Capacitor 10V 1206	Samsung Electro-Mechanics	CL31B106KPHNNNE	2	0.3
22uF Capacitor 10V 1206	Samsung Electro-Mechanics	CL31B226KPHNFNE	1	0.49

- Total parts cost = $88.45

- Sum of Costs
    - Total project cost = Total parts cost + Total labor cost
    - = $88.45 + $31250 = <u>$31,338.45</u>

3.2 Schedule

Week	Dishen	Tommy	Esteban
3/3	Order Parts for sensor and pump circuits. Design schematics for each one.	Design Schematics for power module and control module, choose and order parts.	Begin writing local host server code for the website.
3/10	Incorporate all the sensor and pump schematics into PCB. Finalise version 1 of the PCB.	Incorporate all the power and control schematics into PCB. Finalise version 1 of the PCB.	Research into how to send data from ESP32 to the website using WiFi.
3/17	Test the individual sensor circuits on a breadboard, correct in the PCB layout. Also finalise the second PCB version with corrections.	Test power components in isolation from the system. Debug and make changes to the PCB for round 2.	Work on the backend and frontend of the website and begin MCU code for data transfer between ESP32 and the website.
3/24	Work on the sensor MCU integration to calibrate them and get correct values from readings.	Assemble and test power portion of PCB, make changes if necessary for PCB for round 2.	Work on code for sensor calibration and integration. Also continue working on the main MCU code.
3/31	Test the pump circuit with the moisture sensor and the MCU. Check for proper water feeding and wait times. Incorporate all the corrections in the final PCB.	Rebuild power system on new PCB if necessary. Test and debug changes. Test power module with MCU and pump.	Integrate status lights into the project while combining all previous work and making sure it is working.
4/7	Incorporate all the sensor and pump circuits onto the final PCB and test each module together.	Incorporate all the power and control circuits onto the final PCB and test each module together.	Wrap all previous work together and fix bugs.
4/14	Start working on the presentation and the final report.	Start working on the presentation and the final report.	Start working on the presentation and the final report.
4/21	Finish presentation and final reports.	Finish presentation and final reports.	Finish presentation and final reports.

4. Ethics and Safety

Our main safety concern is the battery portion of the project. Improper use of rechargeable battery components can lead to them overheating or even exploding[14], so we plan to ensure that the electrical connections to the battery portion don't draw more power than the safe level and make sure that the cell voltage doesn't decay below the safe limit. We are using a voltage and current protection IC to further ensure that the power doesn't exceed the limit. Additionally the selected battery has a similar protection circuit built-in [6]. We will also be conducting charge and discharge tests with the battery circuit and will be following the given charging suggestions and battery requirements from the course safety guidelines [15]. As an added precaution, no connections to the battery will be made without both protection circuits connected. The designed protection circuit will also be extensively tested independently of the battery before it is used. These test procedures are defined in section 2.2.3.

The project also involves water being pretty close to the electrical components because of the automatic watering mechanism. This created the need for some way to ensure that these components, especially the batteries, don't come in contact with the water. For this, we plan to have some sort of 3D printed housing for the electrical components. This is also to ensure that they don't come in contact with the soil and just the sensor probes do. We are also going to try and have silicone waterproofing for these 3D printed cases, so that we can plug the holes for the wires going in and out of it.

There are some ethical considerations from the IEEE Code of Ethics that we are keeping in mind while working on this project. The main one is #1: "to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment;"[16].

We know that there are certain harmful metals like cadmium and arsenic that can enter the soil or deposit itself on the plant through various sources like cigarette smoke and water. This is a big issue for home plants since a majority of these tend to be herbs and small vegetables and fruits that people use in their food. Exposure to such harmful materials can cause health complications to the grower in the long run. These harmful substances are more prone to enter the soil when the pH is off balance[17]. Our product aims to safeguard their health and safety by using the pH sensing technology to ensure that the soil pH is maintained at a safe 6.5-7 range.

References

[1] Espressif Systems, "ESP32-S2 Family," [PDF]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s2_datasheet_en.pdf. [Accessed 27-Feb-2021].

[2] Espressif Systems, "ESP32-S2 Technical Reference Manual," [PDF]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s2_technical_reference_manual_en.pdf. [Accessed: 27-Feb-2021]

[3] Espressif Systems, "ESP32-S2-WROOM," [PDF]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s2-wroom_esp32-s2-wroom-i_datasheet_en.pdf. [Accessed 27-Feb-2021].

[4] Espressif Systems, "ESP32-S2 Hardware Design Guidelines," 25-Sep-2020. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s2_hardware_design_guidelines_en.pdf. [Accessed: 17-Feb-2021].

[5] Pratikmokashi, "pratikmokashi/ESP32-S2," *GitHub*. [Online]. Available: https://github.com/pratikmokashi/ESP32-S2. [Accessed: 27-Feb-2021].

[6] Texas Instruments, "BQ297xx Cost-Effective Voltage and Current Protection Integrated Circuit for Single-Cell Li-Ion and Li-Polymer Batteries," [PDF]. Available: https://www.ti.com/lit/ds/symlink/bq2973.pdf?ts=1614561305006&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FBQ2973. [Accessed 28-Feb-2021].

[7] MikroE, "Li-Polymer Battery 3.7V 2000mAh," *MikroElektronika*. [Online]. Available: https://www.mikroe.com/li-polymer-battery-37v-2000mah. [Accessed: 03-Mar-2021].

[8] Texas Instruments , "LP3850x-ADJ, LP3850xA-ADJ 3-A FlexCap Low Dropout Linear Regulator for 2.7-V to 5.5-V Inputs," [PDF]. Available: https://www.ti.com/lit/ds/symlink/lp38501-adj.pdf?ts=1614466610400&ref_url=https%253A%252F%252Fwww.ti.com%252Fpower-management%252Flinear-regulators-ldo%252Fproducts.html. [Accessed 27-Feb-2021].

[9] Mouser, "Programmable Resolution 1-Wire Digital Thermometer," [PDF]. Available: https://www.mouser.com/datasheet/2/256/DS18B20-370043.pdf. [Accessed 3-Mar-2021].

[10] "ESP32 Web Server - Arduino IDE," *Random Nerd Tutorials*, 02-Apr-2019. [Online]. Available: https://randomnerdtutorials.com/esp32-web-server-arduino-ide/. [Accessed: 27-Feb-2021].

[11] "Web technology for developers," *HTTP | MDN*. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Status. [Accessed: 03-Mar-2021].

[12] "Driew USB Water Pump, Water Pump, for Fountain Water Fountain Pump Submersible Water Pump 3W DC 3.5-9v (200L/H)," *Amazon,* [Online]. Available: https://www.amazon.com/Driew-3-5-9V-Submersible-Brushless-Waterproof/dp/B01CG2Y

E6K/ref=pd_ybh_a_8?_encoding=UTF8&psc=1&refRID=3G4V92NCQF15AVKP7S9B. [Accessed 27-Feb-2021].

[13] “ESP32: Connecting to a WiFi network,” *techtutorialsx*, 24-Mar-2019. [Online]. Available: https://techtutorialsx.com/2017/04/24/esp32-connecting-to-a-wifi-network/. [Accessed: 04-Mar-2021].

[14] C. Mikolajczak, M. Kahn, K. White, and R. T. Long, “Lithium-Ion Batteries Hazard and Use Assessment,” Jul-2011. [PDF]. Available: https://www.nfpa.org/-/media/Files/News-and-Research/Fire-statistics-and-reports/Hazardous-materials/rflithiumionbatterieshazard.ashx. [Accessed: 04-Mar-2021].

[15] University of Illinois, “Safe Practices for Lead Acid and Lithium Batteries,” [PDF]. Available: https://courses.engr.illinois.edu/ece445/documents/GeneralBatterySafety.pdf. [Accessed 28-Feb-2021].

[16] “IEEE Code of Ethics,” *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 17-Feb-2021].

[17] E. W. Support, “Should I Worry about Heavy Metals in My Garden Soil?,” *OSU Extension Service*, 25-Feb-2021. [Online]. Available: https://extension.oregonstate.edu/gardening/techniques/should-i-worry-about-heavy-metals-my-garden-soil. [Accessed: 16-Feb-2021].