ECE 445: SPRING 2026

CIRCLE OF LIFE: AUTOMATED DESKTOP AQUAPONICS SYSTEM

By

Anjali Aravindhan

Aishwarya Manoj

Estela Medrano-Gutiérrez

Design Document for ECE 445, Senior Design, Spring 2026

TA: Manvi Jha

May 2026

Project 73

Abstract

This proposal is a more in-depth understanding and visualization of our project Circle of Life: Automated Desktop Aquaponics System. This project is the senior design project for Anjali Aravindhan, Aishwarya Manoj, and Estela Medrano-Gutierrez and will be completed in the Spring 2026 semester of ECE 445: Senior Design at the University of Illinois Urbana-Champaign.

Contents

1. Introduction

1.1 Problem

Urban living and limited indoor space make it difficult for individuals to grow fresh produce sustainably. Aquaponic systems offer an efficient solution by combining fish cultivation and plant growth in a closed-loop ecosystem, but existing systems require frequent manual monitoring and maintenance. Current desktop-scale aquaponics kits often lack intelligent control features and are cost-and-space-prohibitive for individual users.

1.2 Solution

This project proposes the design and construction of a small desktop smart aquaponics system integrating automated environmental and fluid control. The system consists of a compact fish tank and plant grow bed forming a closed-loop water circulation path. An electronically controlled pump circulates water between the tank and grow bed, while a motorized dispensing mechanism provides automated fish feeding. A programmable grow-light module delivers controlled lighting cycles for plant growth. Embedded sensors monitor key system conditions such as water flow, ph level and water temperature. A microcontroller schedules feeding and lighting and processes sensor data. Depending on budget and difficulty, we may add capabilities. Our target customers are younger urban apartment residents and students with an interest in convenience, aesthetics, wellness, and/or sustainability but have limited time to care for plants. Our project would allow these consumers to be able to raise fresh herbs and leafy greens at home without requiring frequent maintenance.

1.3 Visual Aid

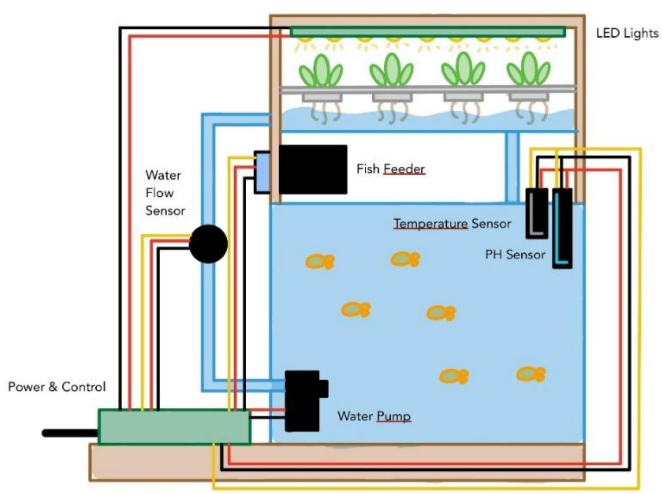


Figure 1: Visual representation of our aquaponics system

1.4 High Level Requirements

- Sensor readings should be measured once an hour for the water flow sensor and at least once every 10 seconds for the pH and temperature sensors, where abnormal readings will be displayed via LEDs (Outside of 6-8 pH, 78-80 degrees Fahrenheit, and rate of 3–5x the total tank volume per hour)
- All motorized subsystems such as the fish feeder and water pump will operate for at least 24 hours without issue, in which time the fish feeder should dispense food once and water pump will turn over the volume of the tank at least three times per hour.
- All visual indicators (LEDs) both used in the lighting schedule & displaying sensor abnormalities will be updated/checked at least every 10 seconds

2 Design

2.1 Block Diagram

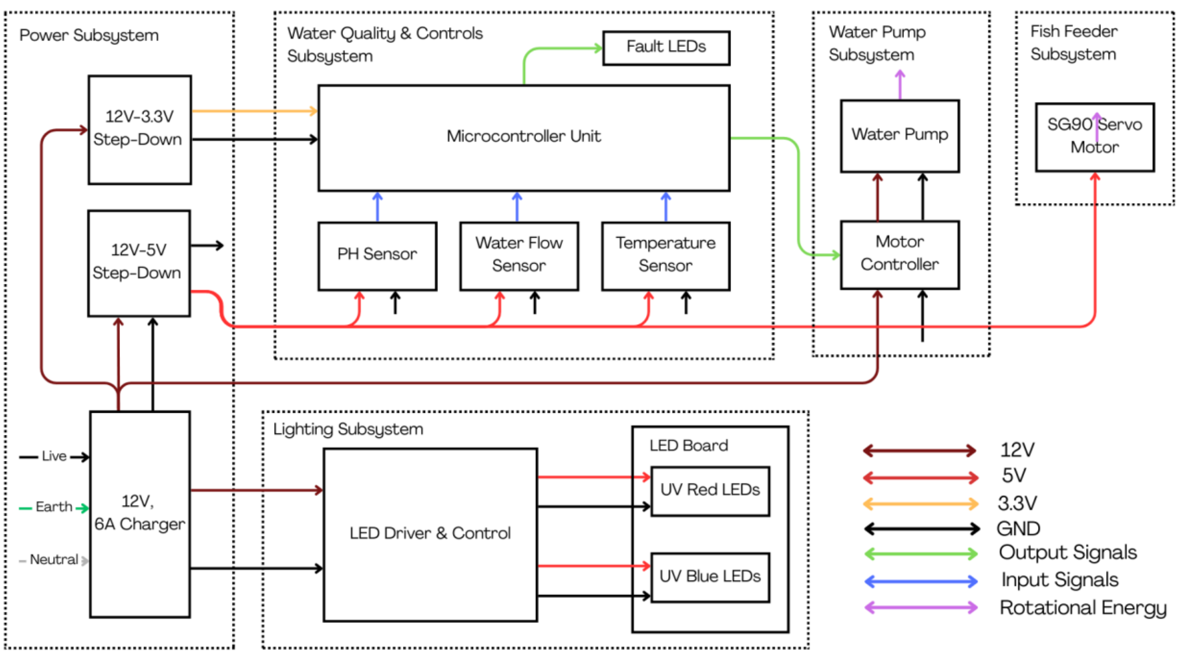


Figure 2: Block diagram with separate modules

Our project is an aquaponics system that is automated and will let the user know when the water flow, pH, and temperatures are abnormal and harmful to fish. On top of this, there will be an automated fish feeder to feed the fish once in 24 hours. The project is broken down into the power subsystem, water quality/controls, lighting, water pump, and fish feeder. The power subsystem starts with a 12V wall input and steps it down to 5V and 3.3V for the sensors and microcontroller respectively. For the sensors, they all require 5V and their outputs are connected to the microcontroller. From there, the microcontroller will interpret those values and light up the corresponding fault LEDs. The 5V is also supplied to the servo motor, which is the fish feeder mechanism. The fish feeder mechanism uses a PWM signal from the microcontroller to feed the fish at least once every 24 hours. The lighting subsystem uses an LED driver and the microcontroller to set up a schedule for the lights. It uses 12V straight from the wall. Lastly, the water pump subsystem uses a motor controller from the microcontroller and regulates the pump using a PWM signal.

2.2 Physical Design

2.2.1 Aquaponics Fish Tank & Grow Bed

Our desktop aquaponics system is designed around a compact 16.5 in L x 8.75 in D x 10.75 in H tank to meet the minimum 5-gallon requirement for a betta fish [1]. The grow bed uses a nutrient film technique (NFT) channel made from a 4" PVC pipe mounted on a wooden rack, so the bottom of the pipe sits about 1–2 inches above the tank rim and is tilted at a ~2% slope to maintain a flow of water coming back into the fish tank. The water is pumped from the tank into the left side of the PVC channel through an inlet hole. The water pump will be attached through built-in suction cups on the left most corner inside the tank, while the water flow sensor will be attached to the PVC itself. The wooden rack also serves as a mounting structure for the LED grow lights.

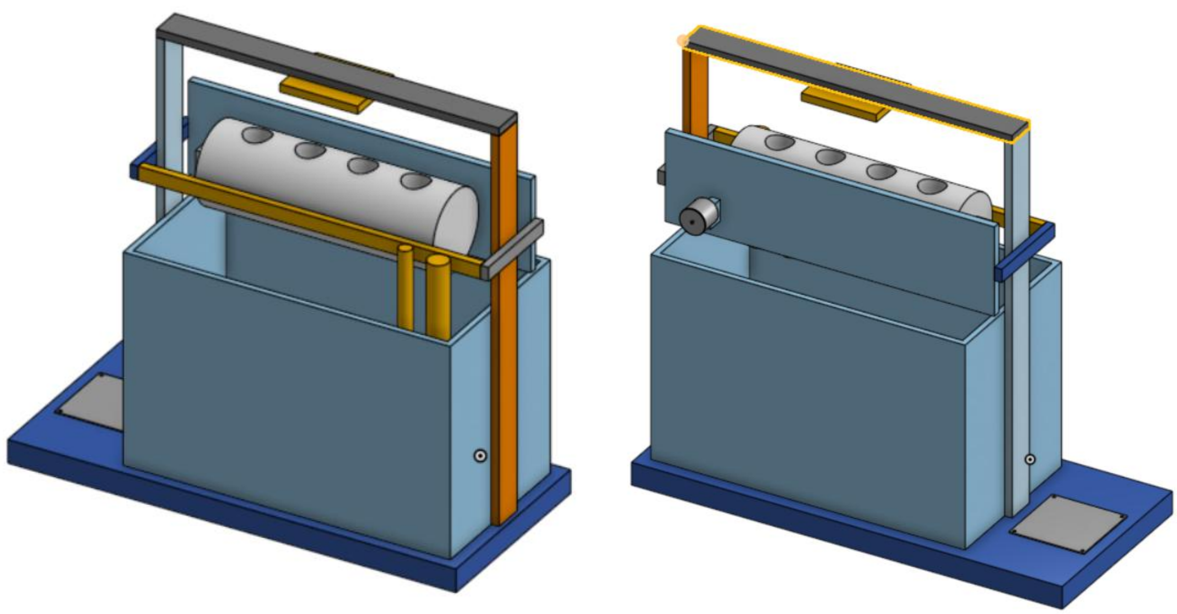


Figure 3: CAD Assembly model of aquaponics system prototype

We will be having around 4 cups of 3.4cm diameter each in the PVC pipe with each plant [16]. The fish feeder will also be attached to the wooden rank, at the back of the PVC holder. We will be using 4’’ pipe clamps to attach the PVC tube to the wooden structure. The temperature and pH sensor will be attached to a small plank on the wooden structure, attached with glue with an appropriate length so that enough of the sensor is submerged on the water to do its duty. There will be a wooden base at the bottom to hold the tank and the controls.

Figure 4: Net cups

We will be buying most of the parts and will get help from the ECE Machine Shop to build it. Below is a preliminary version of our mechanical design, with several boxes representing the sensor and PCB placement.

In Figure 5, you can find all the physical measurements of our current design as of now. Several things, such as the exact position of the fish feeder, are still to be fully decided. Although they currently have a set position in the diagram, we might move it based on convenience during building and testing of the system. Cabling and tubing are excluded from this physical diagram.

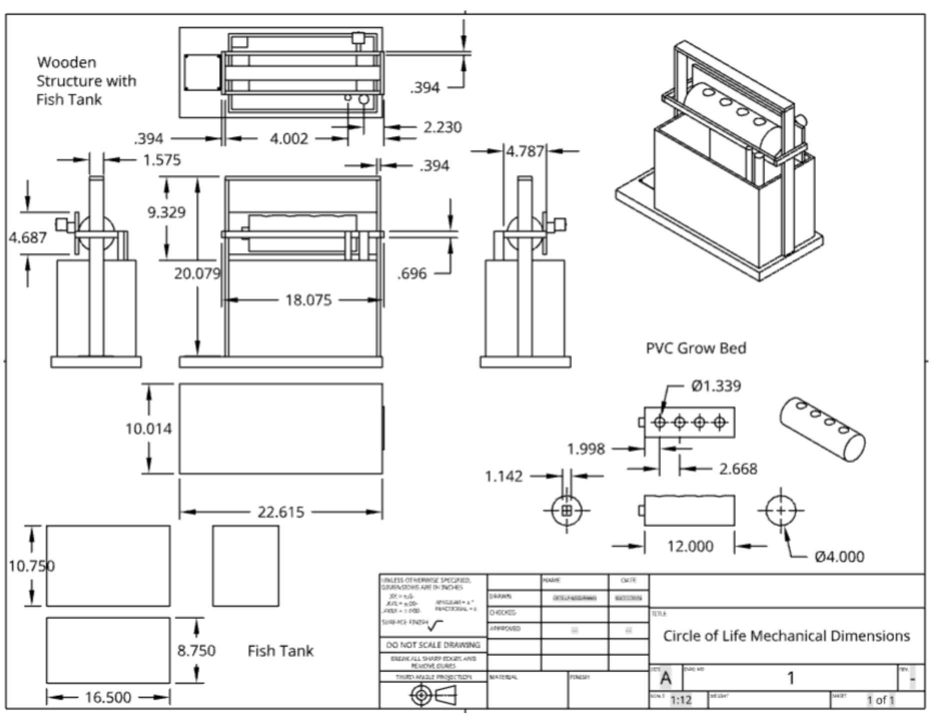
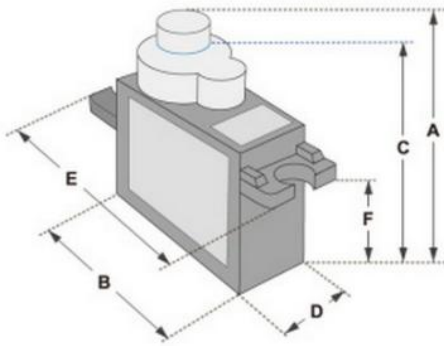


Figure 5: Mechanical dimensions technical drawing

2.2.2 Fish Feeder 3-D Printed Shaft

For the fish feeder draft, we are making a 3D printed enclosure that will be attached to the shaft of the following motor using these measurements for our SG90 servo motor.



Dimensions & Specifications
A (mm) : 32
B (mm) : 23
C (mm) : 28.5
D (mm) : 12
E (mm) : 32
F (mm) : 19.5
Speed (sec) : 0.1
Torque (kg-cm) : 2.5
Weight (g) : 14.7
Voltage : 4.8 - 6

Figure 6: SG90 Servo dimensions

Common pellet sizes range from 0.5mm to 1.5mm for small fish like betta, so we accounted for that size when making the slit of the 3D printed shaft. In figure 7 you can find the exact measurements of the shaft based on the SG90.

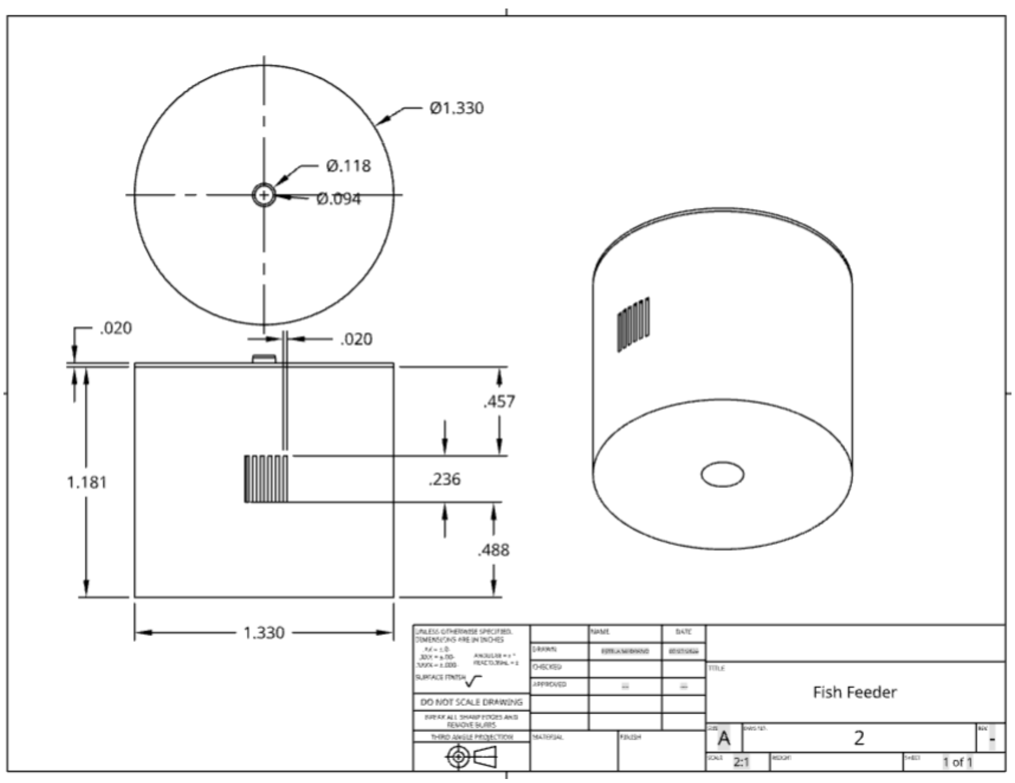


Figure 7: Fish feeder 3D printed shaft coupler technical coupler

2.2.3 Main and LED PCB Board

Our main power and control PCB and our LED PCB will have a max size of 100mm * 100mm. For our main board, we will be adding 3mm mounting holes on each corner, whereas for the LED board we will be adding same size mounting holes on the middle left and right side of the PCB to be attached to the wooden plank on top of our system as seen above. For both, we will be designing 3D-printed PLA enclosures for protection. For the PCB, we will be making a fully closed design with only holes for the connectors, mainly the DC Barrel Jack, the USB-C and the 2-wire to 3-wire through hole Terminal Blocks. For the LED, our enclosure will be open on the LED side, and will be mainly be used to isolate the wood from the PCB directly in case moisture escapes onto the wood from the tank.

In figures 8 and 9 are the current physical size requirements of our main and LED board.

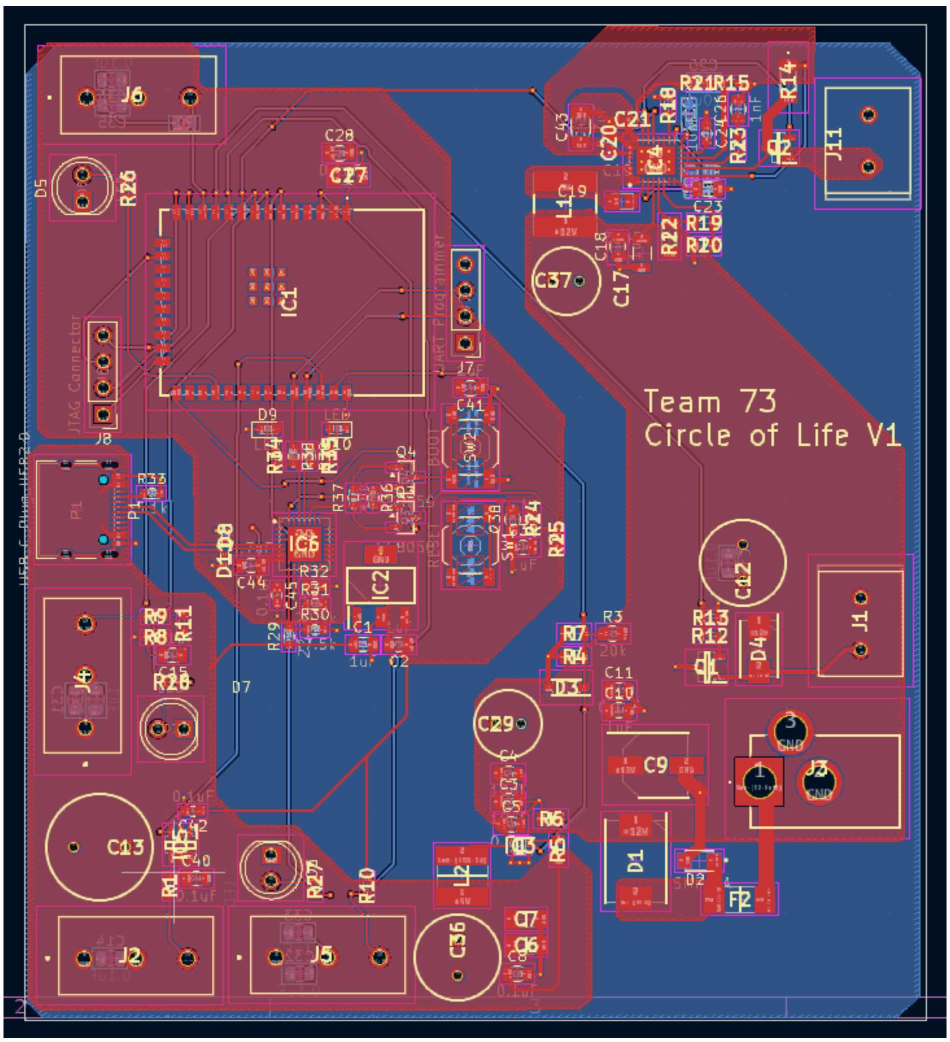


Figure 8: Main board layout

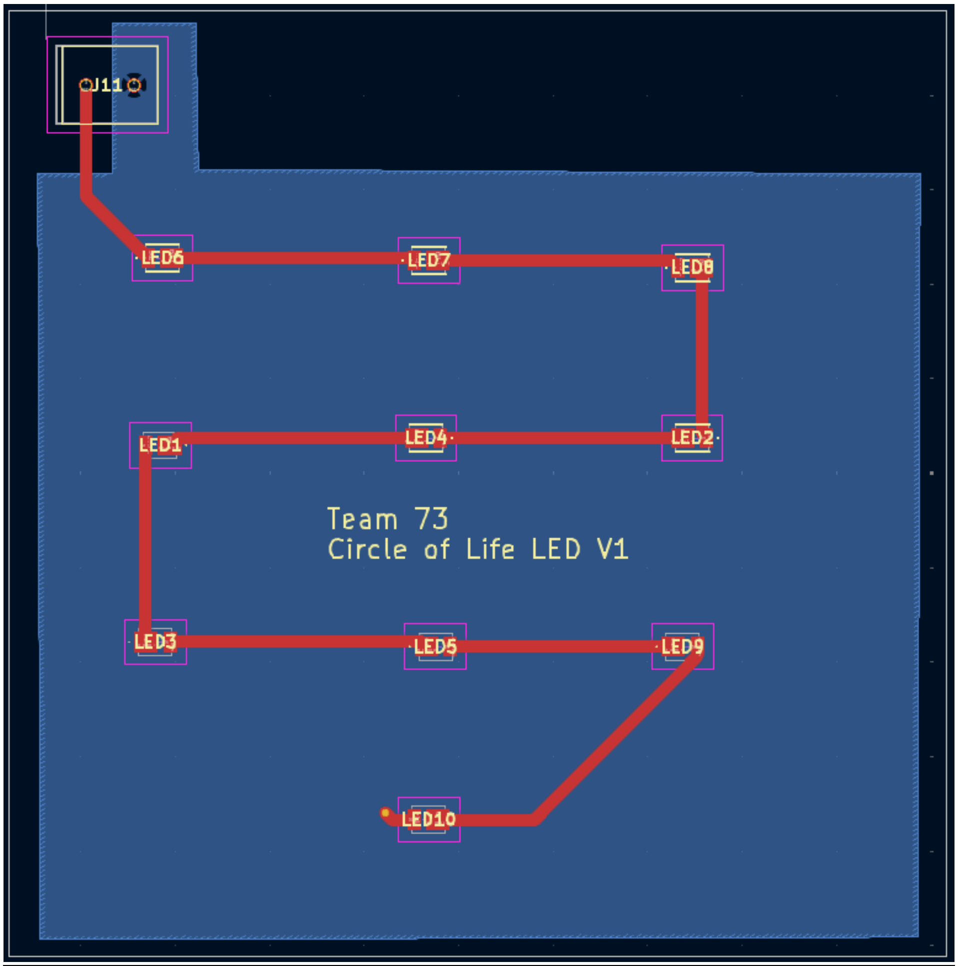


Figure 9: LED board layout

2.3 Subsystem Overview

2.3.1 Fish Feeder Subsystem

A simple automated fish feeder will be implemented using a servo motor operating between two angular positions, one away from the fish tank and another towards the fish tank for dispensing food. A custom 3D-printed food container will be mechanically coupled to the servo shaft using screws and will include a small outlet opening that allows food to dispense when the container is rotated downwards.

The servo motor, which operates from 4.8 to 6V, will be powered using 5V and will be controlled by taking in standard PWM signals coming out of a GPIO pin from our microcontroller [8] [11]. This microcontroller will also serve as the controller for the other subsystems [7]. Since the servo motor expects PWM signals at 5V but the microcontroller can only output signals of maximum 3.3V amplitude, we will be using a level shifter to step up this signal from 3.3V to 5V.

We have also added a big electrolytic capacitor to keep the 5V power in the servo stable in case of current spikes during startup or load variation. We have added another 0.1uF capacitor to filter high frequency noise coming into the servo, and a series resistor to reduce ringing and overshoot on the PWM line provoked due to long wire. This value might be changed with testing.

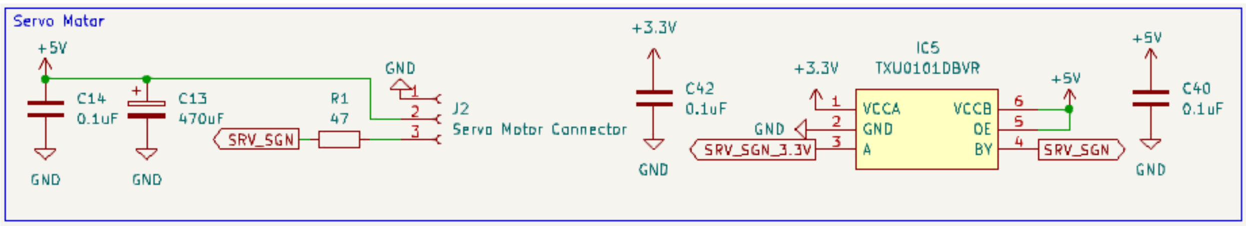


Figure 10: Schematic of servo motor

Requirements	Verification
• The fish feeder subsystem must receive a PWM signal from the microcontroller to turn on the servo motor	• Ensure the board is in a ready state to receive the signal: ESP-32 is getting 3.3V and the servo is receiving 5V. This shall be checked via multimeter. It should also be running for at least 60 seconds of error free before verification. • Then, send the signal from the microcontroller unit to the servo motor and confirm movement on the motor controller side. Check with an oscilloscope to make sure it is a valid PWM signal at the correct duty cycle and with the right voltage level coming from the level shifter.
• The servo motor should rotate approximately 180 degrees (+/- 20 degrees) based on the PWM signal sent from the microcontroller	• Ensure the board is in a ready state to receive the signal: ESP-32 is getting 3.3V and the servo is receiving 5V which shall be checked via multimeter. It should also be running for at least 60 seconds of error free before verification. • Then, send the PWM signal from the microcontroller and ensure the motor has moved. Record the starting position before the signal is sent. Then, use a protractor to confirm that the rotation

	is approximately 180 degrees from the start position.
• The fish feeder should dispense food into the tank at least once every 24 hours.	• Ensure the board is in a ready state to receive the signal: ESP-32 is getting 3.3V and the servo is receiving 5V which shall be checked via multimeter. It should also be running for at least 60 seconds of error free before verification. • Set up the system for a smaller interval test (such as 5 minutes) and program the MCU to send the fish feeder rotation signal once to rotate down, maintain downward position for 1 second, and rotate back upwards for that interval. Check to see that the motor moves correctly in that interval.

2.3.2 Water Quality Subsystem

This subsystem monitors water quality and circulation using a water flow sensor, a water pH sensor, and a water temperature sensor. Because the pump lifts water to the grow bed, monitoring flow provides feedback to confirm that water is reaching the plants and to detect faults such as clogged tubing, a disconnected hose, or an empty tank. The controller will continuously measure the flow rate to verify [expected operation]. If the measured flow deviates from the operating range, the controller can either adjust the pump's PWM signal within safe limits and/or trigger the fault indicator LED and stop the water pump from running [10]. The water flow sensor is powered through 5V, and we are pulling up the sensor output to 3.3V to keep the logic-high level safe for the ESP32.

Alongside this, we will have a pH sensor to measure the pH of the water. As we aim to have beta fish in the tank, that requires a pH of roughly 6.8 to 7.5, and we will have plants that require that slightly acidic to neutral pH range as well. If the pH is outside of this range, we will have an LED indicator so that the user knows it is time to change the water [6]. The pH sensor outputs an analog signal of 5V, and the voltage levels outputs in millivolts directly translate to different pH levels. This signal must be converted to digital through one of the MCU ADCs to be able to read it appropriately and write the code for faults. Because this output signal is 5V, it must also be stepped down to 3.3V using a voltage divider.

Finally, we will have a sensor measuring the temperature of the water to ensure that it is habitable for the fish. Again, for beta fish this requires a temperature of 76 to 85 degrees Fahrenheit. The temperature sensor will measure the temperature of the water in the tank and if it is too high or too low, an LED indicator will be triggered, allowing the user to change the water or the temperature of their room [14]. The output of our temperature sensor is a digital signal that outputs between 9-12 bits of selectable resolution. This signal will be directly to a GPIO and pulled high to 3.3V similarly to the water flow sensor, which will allow the signal to not flow whenever the sensor is not pulling the line low. We have also added general decoupling to filter high and mid frequency noise and keep the supply to the sensors clean and stable.

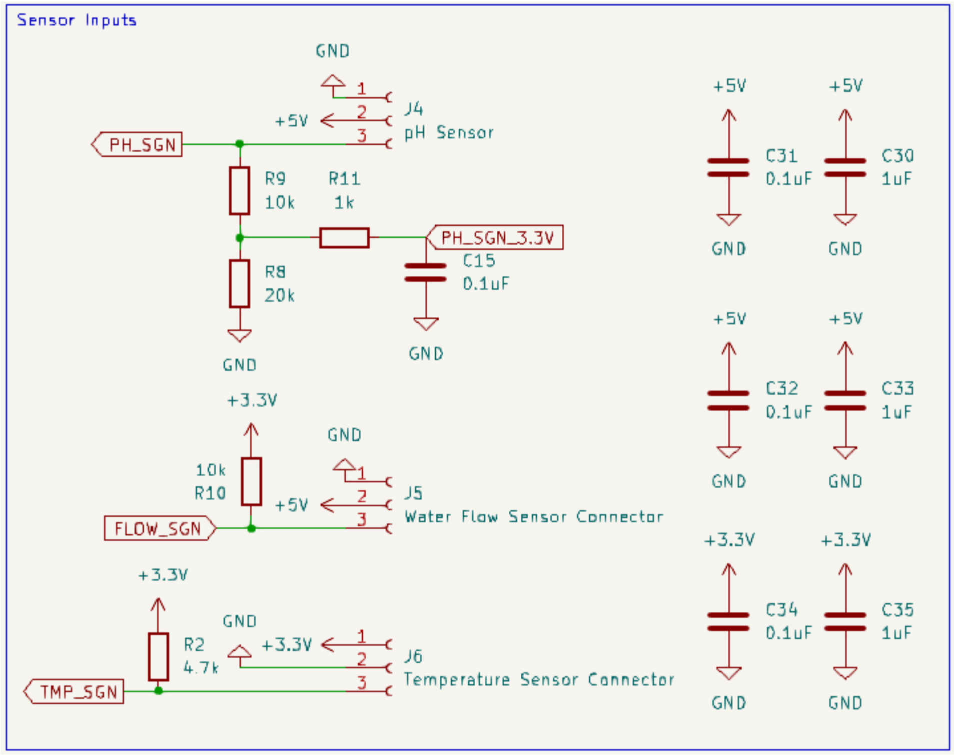


Figure 11: Schematic of sensor inputs

For programming, we are currently utilizing ESP32-WROOM-32E-N8R2. We added JTAG for debugging purposes, and a combination of BOOT and RESET buttons plus UART to program the microcontroller as per the datasheet information. All signals for the sensors, fault LEDs and enable signals can be found in their corresponding pins, mostly GPIOs and ADCs, except for the programming connectors which have their own respective pins.

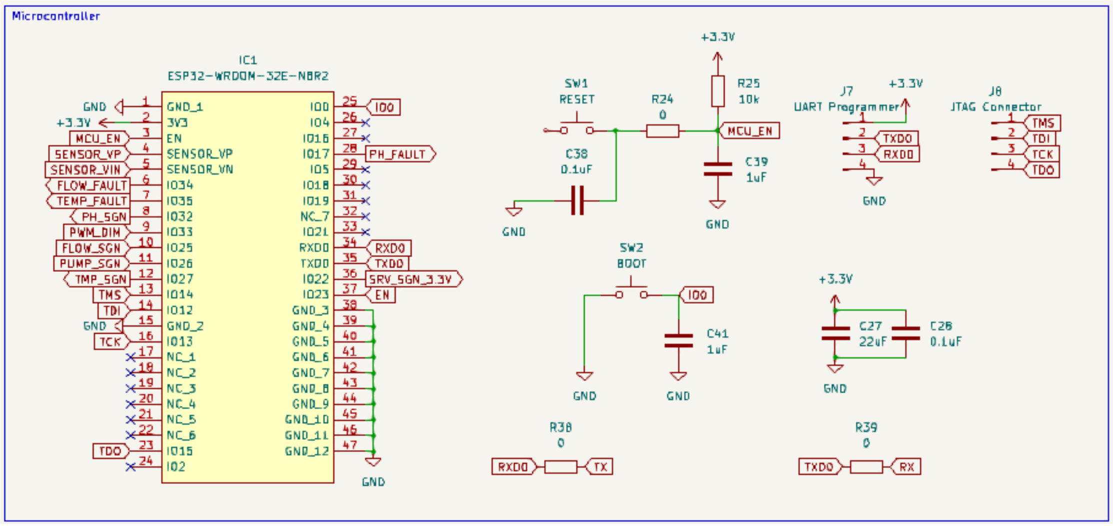


Figure 12: Schematic of MCU

For ease of programming, we also added a USB-to-UART bridge to be able to program MCU through USB-C instead of UART pins directly. For full disclosure, although the datasheet was used to make this part of the schematic, this design is for the most part directly taken from the schematic of the ESP32 DevKitC V4. Since this part is critical for the functionality of our project since without it programming the MCU is difficult, we decided that utilizing a design that is known to work would be the best choice in this case. In our references you can find the schematic utilized. We ensured that the functionality of this dev kit made sense too by thoroughly reading the datasheet. We added ESD protection diodes that protect the MCU from static discharges.

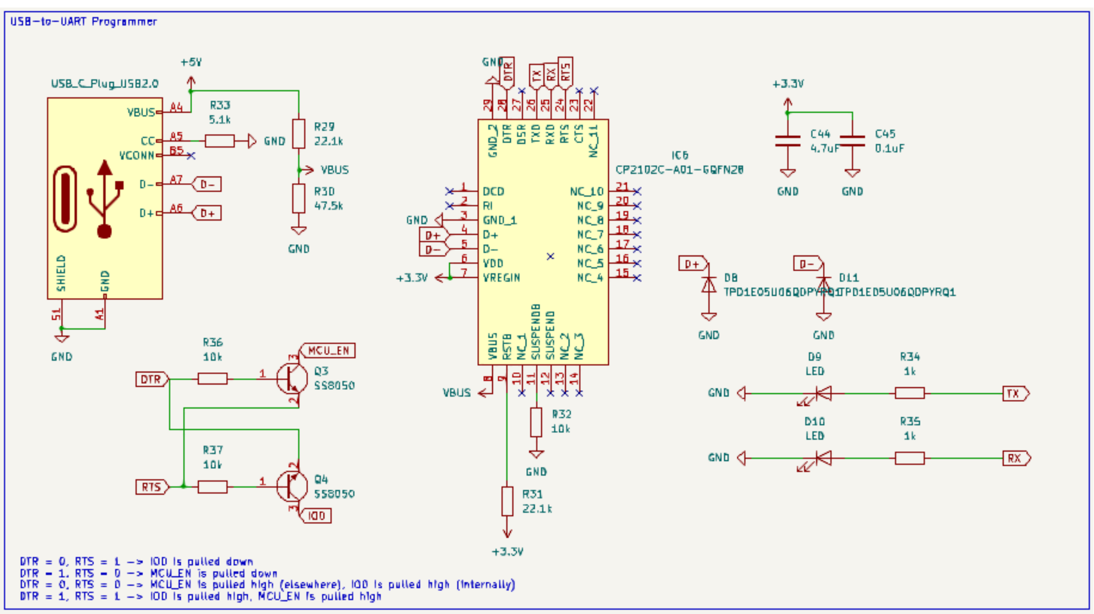


Figure 13: Schematic of USB-to-UART Programmer

The DevKitC V4 schematic utilizes an auto-reset and boot circuit for flashing the ESP32. The DTR and RTS of the USB-to-UART bridge drive two transistors that pull down the EN and BOOT pins. The main thing to note is that when EN goes from low to high and the BOOT pin helps low, the ESP32 enters programming mode. In the schematic you can see the truth table this circuit forms. We also added LEDs that will turn on during programming.

The main difference is that our connector is a USB-C instead of a Micro-USB, so we mainly accounted for that change ourselves. For the USB-C connector, the CC voltage is used to determine how much current is available from the upstream port. The standard requires that CC be pulled to ground with a 5.1kOhm resistor as shown above.

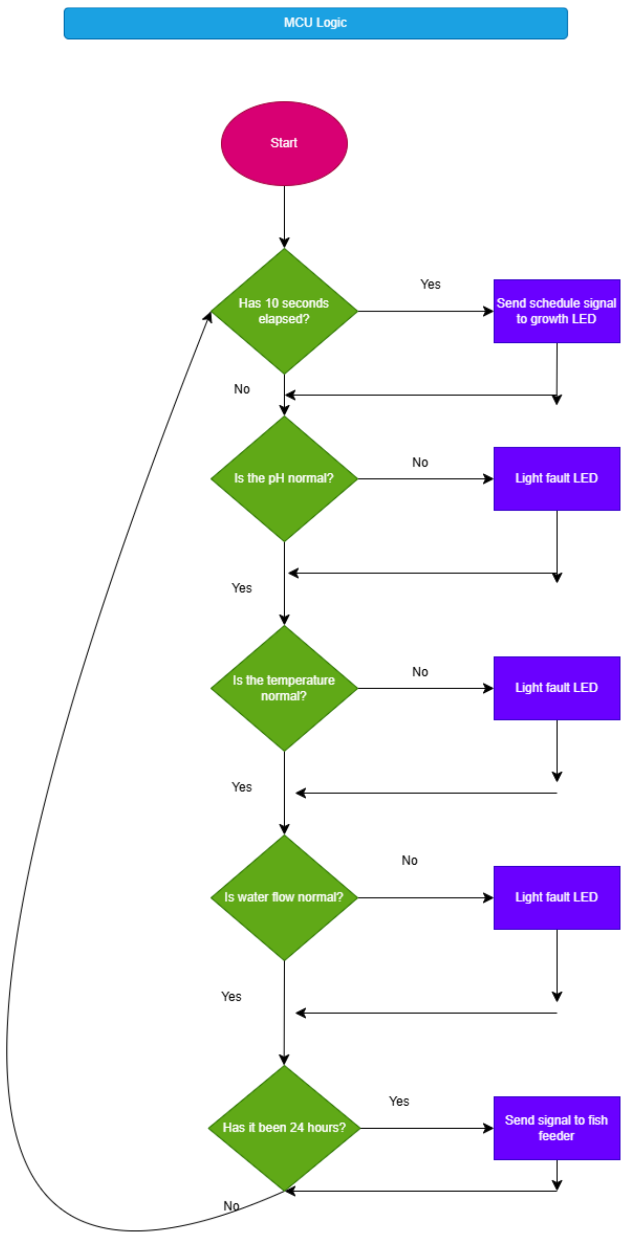


Figure 14: MCU Logic

Requirements	Verification
• The pH sensor must accurately measure the pH of the water in the fish tank within ± 0.5 pH, and it outputs a voltage reading corresponding to the pH it reads. The MCU must be able to read this signal properly and send out a fault signal to the LED if pH is wrong.	• Ensure the board is in a ready state to receive the signal: ESP-32 is getting 3.3V and the pH is receiving 5V which shall be checked via multimeter. It should also be running for at least 60 seconds of error free before verification. • Use two different water samples with different, known pH values and use the pH sensor readings to compare against the known values to make sure it is within +/- 0.5 of the actual value.
• Test MCU is working	• Attempt to program it using both UART pins and BOOT/Reset buttons. Wait for serial terminal message to say that code has been uploading properly. • Next, test using USB-UART connector, and check for message saying code has been uploaded through the serial terminal
• The fault LEDs will indicate when an abnormal pH is sensed by the pH sensor	• Ensure the board is in a ready state to receive the signal: ESP-32 is getting 3.3V and the LEDs are receiving 5V which shall be checked via multimeter. It should also be running for at least 60 seconds of error free before verification. • An abnormal pH is anything outside the range of 6-8 pH since that is the needed pH for fish. Using a water sample with a known pH outside of the normal range for a fish (6-8 pH), confirm the LED fault light should receive a signal from the microcontroller to be lit.
• The temperature sensor must accurately measure the temperature of the water within 5 degrees Celsius, and it will output the temperature reading into the temperature register, which the microcontroller will read	• Ensure the board is in a ready state to receive the signal: ESP-32 is getting 3.3V and the temperature sensor is receiving 5V which shall be checked via multimeter. It should also be running

from. If temperature is wrong, MCU must send out a fault signal to the LED indicator.	for at least 60 seconds of error free before verification. • Based on using a thermometer to measure the actual temperature of the water in the tank, we will then use the temperature sensor to measure the temperature and confirm it is in the bounds of accuracy with the actual temperature (+/- 5 degrees). • We will test that the fault is working correctly by heating up the water, setting the code to flash LED if water is too hot, and verify that LED is flashing when temperature is on water.
• The water flow sensor must measure the flow rate within an accuracy of 3%, and MCU must be able to read this signal. If water flow is not withing correct range, MCU must be able to communicate with water pump to change flow rate or send out a fault signal to the LED if there is a fault condition.	• Ensure the board is in a ready state to receive the signal: ESP-32 is getting 3.3V and the water flow is receiving 5V which shall be checked via multimeter. It should also be running for at least 60 seconds of error free before verification. • The water flow sensor should be measuring accurate water flow, this will be determined by seeing how long it takes to replace the entire volume of the tank, since for fish it must be at a rate of 3—5x the total tank volume per hour [10]. Use an external container to see how long it takes to move the full volume of the tank into it and calculate the flow rate. Compare this to the value from the water flow sensor to confirm the accuracy. • We will test that the fault is working correctly by slightly blocking the tube and verifying that the LED turns on if no water is allowed to come through the tube.

2.2.3 Lighting Subsystem

The lighting subsystem serves as artificial light sources for plants in our desktop aquaponics system. The purpose of this subsystem is to make sure plants will get the correct amount of light and intensity per day to simulate growth due to sunlight from the Sun. The lighting subsystem will use powerful horticulture LED colored lights with alternating deep blue and far-red colors to simulate sunlight and promote photosynthesis. It will be mounted above the plants and the aquarium portion of the aquaponics system and shine down upon the plants [3]. Our LEDs chosen to have the following characteristics:

- Deep Red: 2.25 VF, 200-300mA
- Royal Blue: 3.08 VF, 200-300mA

We will have a total of 5 red LEDs and 5 blue LEDs in series with each other, drawing in between 200-300mA in total. Adding forward voltages, we get the following.

- For the blue LEDs, 5 x 3.08 = 15.4V.
- For the red LEDS, 5 x 2.25 = 11.25V.
- In total, 15.40V + 11.25V = 26.65V.

Since this number of LEDs needs a higher voltage than the one supplied, we have decided to utilize a constant-current LED driver that fits the criteria above to take in 12V and generate higher VOUT needed to push a controlled current through the LED string. The LED driver chosen will also support PWM dimming used for our scheduling, and over voltage and over current conditions for safety. This LED driver generates higher VOUT by using an internal boost converter. Our LED driver chosen, the LT3922, is a step-up converter that allows for LED strings up to 34V, and input voltages ranging from 2.8 to 36V.

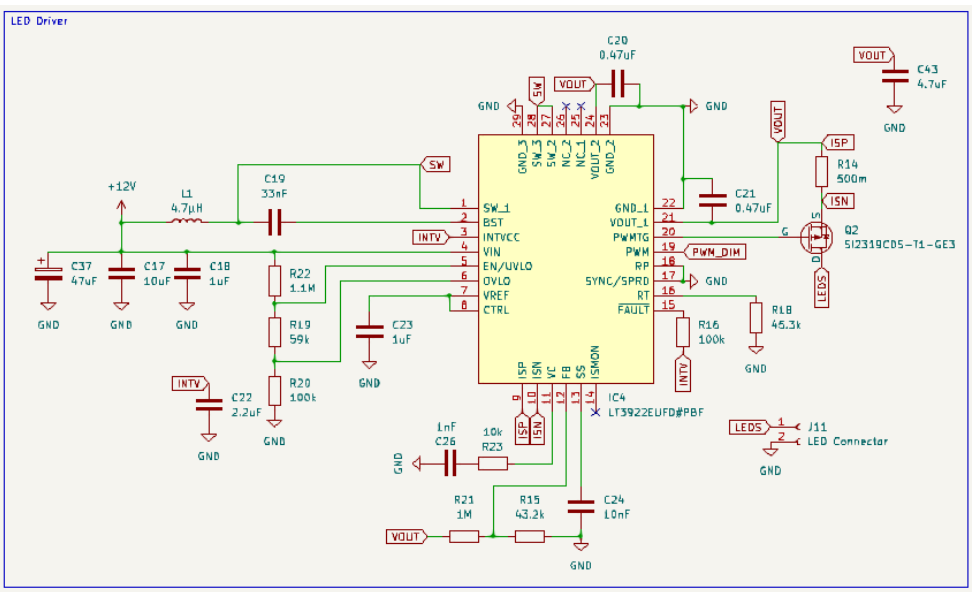


Figure 15: Schematic of LED Driver

To create the schematic, we have mostly followed the application notes on the data sheet when appropriate and changed values when needed. Attached is a picture of the application example followed and our calculations for our custom values. These equations can be found directly in the datasheet.

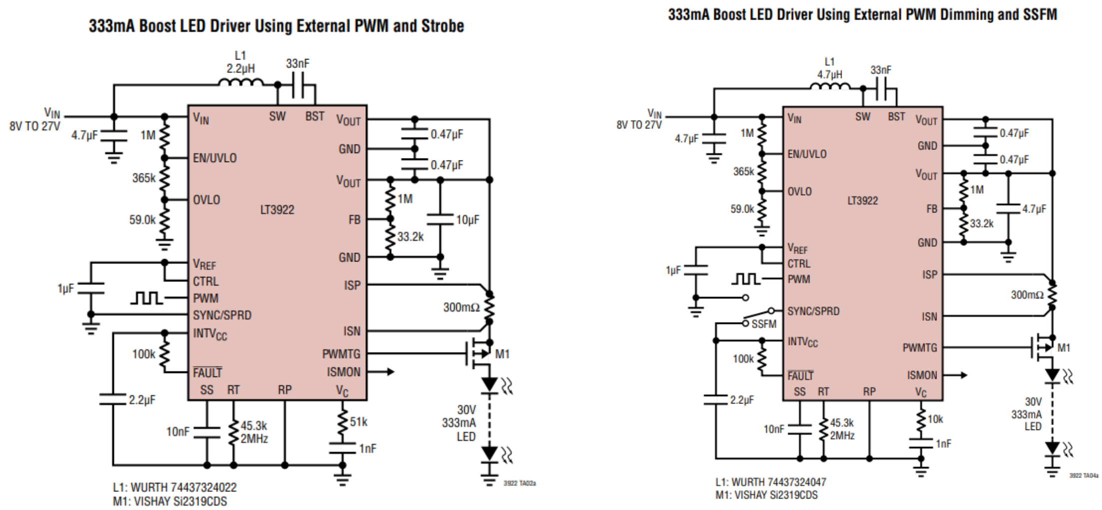


Figure 16: Application notes for LT3922

For VC pin, the recommended starting values in the datasheet are 1nF and 10k and thus were chosen as such. For the RT resistor, we choose a higher frequency because that would allow us to have a smaller inductor that fits into our PCB constraints. We chose the same frequency as in the application examples to be able to keep the inductor and decoupling capacitance similar.

Table 1. R_T Resistance Range

SWITCHING FREQUENCY	R_T
2.0 MHz	45.3k
1.6 MHz	57.6k
1.2 MHz	78.7k
1.0 MHz	95.3k
400 kHz	249k
200 kHz	499k

Figure 17: Frequency table for RT resistor

We are not using ISMON nor SYNC/SPRD so ISMON is unconnected and SYNC is GND. ISMON outputs a amplified current monitoring value which we don't need since we are not checking for that in our project, and SYNC is for driving the LED driver from an external clock, which is also not needed for our project.

FOR OVLO and UVLO, we choose values that fit the following equations according to our own desired ranges for UVLO and OVLO. We choose a minimum input voltage of around 10V and a maximum of around 15V, which led to the resistor values seen in the schematic above.

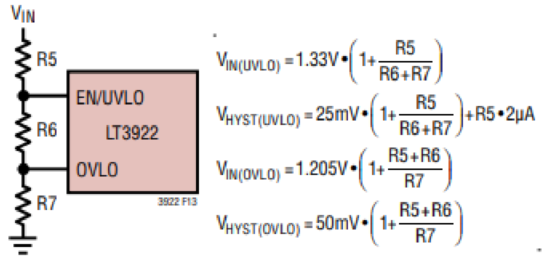


Figure 18: OVLO and UVLO resistor divider equations

Similarly for the FB resistors, these were chosen for VOUT to be the required output voltage calculated for the 10 LEDs in series. The resistance values were chosen to account for forward voltage variations possible in the LEDs. Using voltage maximums of our chosen LEDs we get the following:

- For the blue LEDs, 5 x 3.7 = 18.5V.
- For the red LEDS, 5 x 2.7 = 13.5V.
- In total, 18.5V + 13.5V = 32V.

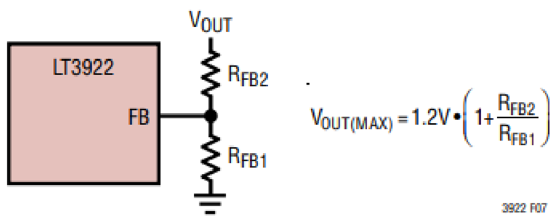


Figure 19: FB resistor divider equations

We still decided to go for lower (around 28V) for safety reasons since the FET chosen from the datasheet is only rated for 40V.

For the shunt resistor, we need 200mA LEDs instead of 333mA. If for 333mA they use a 300mOhm resistor, the new resistor value was chosen using the following equation:

$$\frac{333mA}{200mA} * 300m\Omega \approx 500m\Omega$$

Our LEDs can be found on a separate board, since we will put them at the top of the grow bed. The rest of the power systems will be at the base of the project for easy accessibility and cabling.

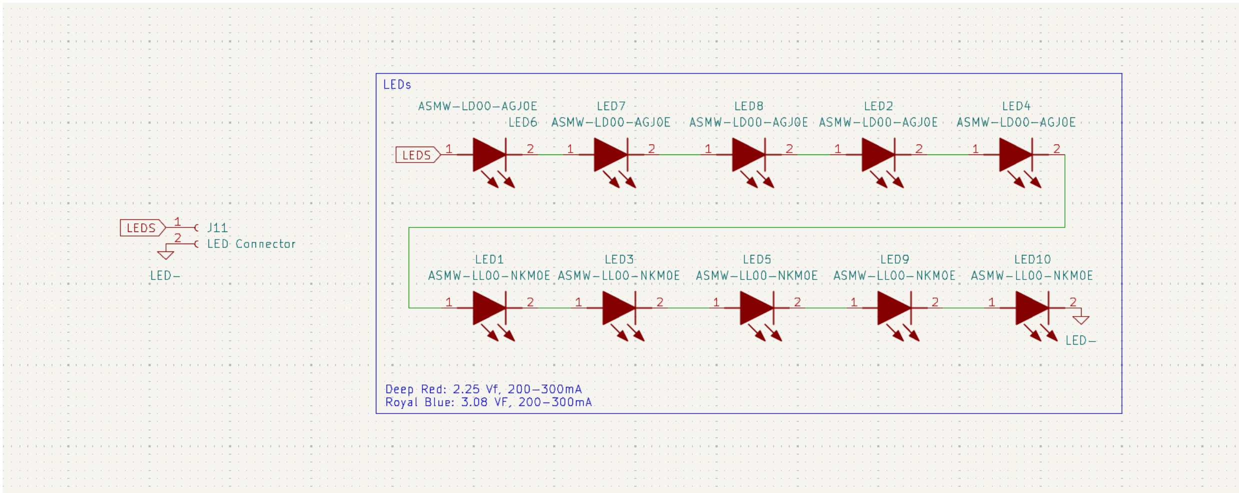


Figure 20: Schematic for LED board

Requirements	Verification
• The LED driver should supply at least the forward voltage in series to the blue and red growth LEDs as well as should not overheat after continuous time intervals of being lit.	• Ensure the board is in a ready state to receive the signal: ESP-32 is getting 3.3V and the LED driver is receiving 12V which shall be checked via multimeter. It should also be running for at least 60 seconds of error free before verification. • Use a multimeter to verify the grow LEDs are being powered correctly based on the LED driver output voltage of around 29V and test by measuring output voltage with a multimeter, then leaving the lights on for a fixed amount of time (30 mins) and making sure it is not overheating/experiencing any power surges. • Connect the LEDs to an oscilloscope for some time to observe any abnormalities such as voltage spikes.
• The lighting subsystem should be to dim the lights whoever we want according to the PWM input to the LED driver.	• Ensure the board is in a ready state to receive the signal: ESP-32 is getting 3.3V and the LED driver is receiving 12V which shall be checked via multimeter. It should also be running for at least 60 seconds of error free before

	verification. • The microcontroller should be sending a constant PWM signal to the LED driver. This should be verified by using an oscilloscope to measure the PWM output of the ESP32 combined with visual observation of the LED state change. • When duty cycle is changed, we should see a change in brightness to the LEDs.
• LED Driver should stop operation under UVLO and OVLO conditions	• Using an external power supply, power VIN with lower than 10V and test output of LED Driver. Similarly, tests with VIN higher than 15V and test output of LED Driver. In both cases, there should be no voltage output on VOUT and the LT3922 should stop switching. Using an oscilloscope, we can check the PWMTG output to see if there is a PWM output when a external PWM signal is coming in.

2.3.4 Water Pump Subsystem

The water pump will be in series with the water flow sensor, sending the water from the fish tank up to the plants. We will be using a circulation pump that is waterproof, and depending on the water flow sensor's outputs, we will be controlling the speed of the circulation pump by PWM modulating the supply voltage using a MOSFET [5]. To control the speed, we utilized an N-MOSFET as a low-side switch. When the gate of the MOSFET is driven high by a signal coming from the microcontroller, the MOSFET turns on and completes the circuit, allowing the water pump to run. The speed is controlled depending on the duty cycle of the PWM signal into the gate, since the motor only sees the average voltage which depends on the supply voltage and the duty cycle.

$$V_{avg} = D * 12V, \qquad where\ D\ is\ the\ duty\ cycle$$

There is also a flyback diode on the motor because when the motor is turned off, initially the current will try to keep flowing. If this current has nowhere to go, the voltage in the drain will shoot up, which can damage components. The flyback allows for that current to recirculate through there, preventing extreme voltage spikes. Other protections added include the bulk and decoupling capacitor to help with current spikes when the pump starts and to prevent high-frequency noise. We also added a series gate resistor to limit the gate current coming into the gate from the MCU. This value might be changed later If switching is too low. Lastly, we added a pulldown resistor to ensure that the gate is never floating, and the pump is always off when there is no operation.

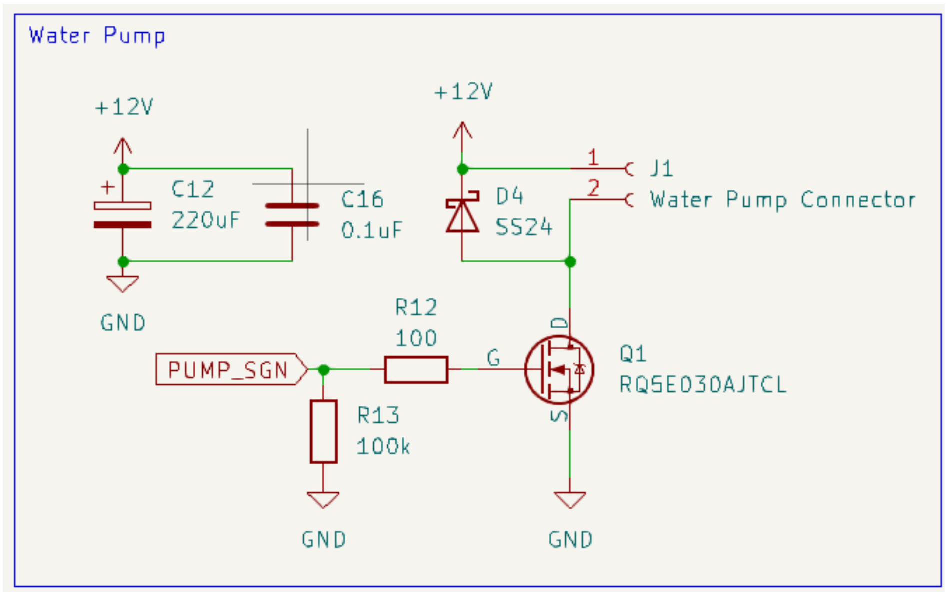


Figure 21: Schematic for water pump

Requirements	Verification
• The water pump will circulate water which will result in the tank volume being replaced at least 3-5 times per hour.	• Ensure the board is in a ready state to receive the signal: ESP-32 is getting 3.3V and the water pump is receiving 12V from the motor controller which shall be checked via multimeter. It should also be running for at least 60 seconds error free before verification. • Fill the fish tank with clean water. When the water is circulated to the plants and falls back down, collect the water in an external container and see how long it takes to empty the fish tank. From there, extrapolate the rate of the tank volume being replaced to confirm it is within the bounds.
• The pump should be operational for at least 24 hours without issue.	• Ensure the board is in a ready state to receive the signal: ESP-32 is getting 3.3V and the water pump is receiving 12V from the motor controller which

	shall be checked via multimeter. It should also be running for at least 60 seconds error free before verification. • Set up the water pump and ensure there is water flow between the fish tank and the plants on top. • Then, leave the pump running for 24 hours and record any abnormalities in the water flow by using the water flow sensor and making sure the rate is always above 0.
• The water pump will change speed depending on the duty cycle of the PWM going into the MOSFET.	• Check the PWM signal going into the water pump with an oscilloscope, then check if speed has changed through visualization.

2.3.5 Power Subsystem

The power subsystem's main goal is to provide power to the other subsystems in this project, including but not limited to the lighting, fish feeder, water quality, and pump. The project will be powered using an external AC-DC 12V, 6A charger. We will be using different power conversion methods to step down the 12V to 5V and 3.3V for their corresponding components [4].

Our first power part is for the 12V input. The main components are a fuse rated for 10A for over current protection, a TVS diode to clamp spikes of voltage from the input, and a Schottky diode for reverse polarity protection in case needed. It will most probably not be needed thanks to the nature of the DC Barrel Jack connector for our input charger supply, in which case it will be shorted. Lastly, we added decoupling and bulk capacitance for the same reasons as above. In this case, this is very important because it will keep the 12V for the rest of the board stable, since load changes in the pump and LED driver could affect this voltage.

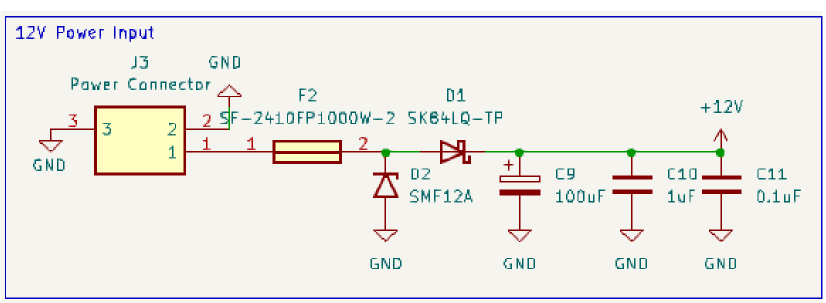


Figure 22: Schematic for12V power input

To step down our voltage from 12V to 5V, we utilized a buck converter with output current capabilities up to 5A. This current output is the main reason we used a buck converter instead of other power conversion methods for this part, since this area supplies many of our higher current components. Buck converters are also better for handling load variations since the voltage remains stable through the feedback network.

We added decoupling capacitance for similar reasons as stated above. R4 is a pulldown when using the EN signal from a GPIO of the microcontroller, whereas R3, R7 and D3 are for when you just want EN to be always high.

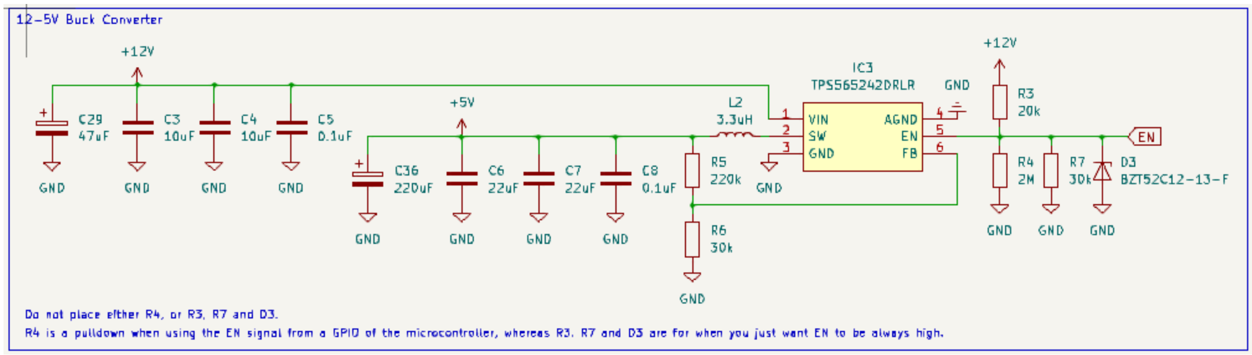


Figure 23: Schematic for 12-5V buck converter

All values were selected as per the data sheet and attached we can find table used to select values for 5V output directly taken from it.

Table 8-2. Recommended Component Values

Output Voltage (V)	R1 (kΩ)	R2 (kΩ)	Minimum L1 (µH)	Typical L1 (µH)	Maximum L1 (µH)	Minimum C_{OUT} (µF)	Typical C_{OUT} (µF)	Maximum C_{OUT} (µF)	CFF (pF)
0.6	0	10.0	0.42	0.82	2.2	44	88	220	—
1.05	7.5	10.0	0.68	1/1.5	2.2	44	66	220	—
1.8	20.0	10.0	1	1.5	2.2	44	66	220	10–470
2.5	95.0	30.0	1.2	2.2	4.7	44	66	220	10–470
3.3	135.0	30.0	1.5	2.2	4.7	44	66	220	10–470
5	220.0	30.0	2.2	2.2/3.3	6.8	44	66	220	10–470
7	320.0	30.0	2.2	3.3	6.8	44	66	220	10–470

Figure 23: Recommended components value table from datasheets

Lastly, we utilized an LDO to step down 5V to 3.3V for the MCU supply. It was chosen because it's simple, it's rated for lower current which works for the MCU and is efficient enough for small voltage differences such as in this case.

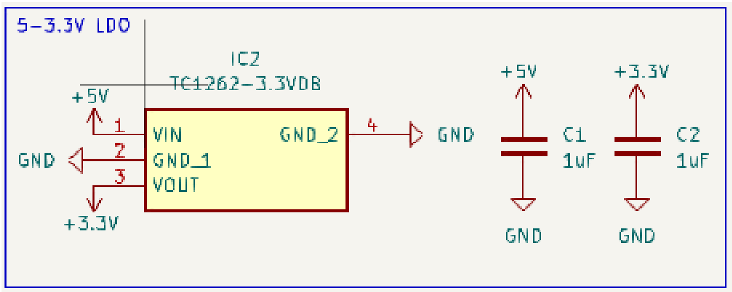


Figure 24: Schematic 5-3.3V LDO

Power Supply To:	From:	Minimum:	Maximum:	Units:
Servo Motor for Fish Feeder	12-5V Buck Converter	4.8	6	V
Water Pump	12V Power Input	6	18	V
LED Driver	12V Power Input	2.8V	26V	V
LED String	LED Driver Output	26.65V	N/A	V
Fault LEDs	MCU	1.8	3.6	V
Water Flow Sensor	12-5V Buck Converter	5	24	V
Water pH Sensor	12-5V Buck Converter	5	5	V
Temperature Sensor	12-5V Buck Converter	3	5.5	V
MCU	5-3.3V LDO	3.0	3.6	V

Figure 25: Table of input voltages and system power connections

Requirements	Verification
• The 12V to 5V buck converter should be supplying at least 5V (+/- 1V) to the water feeder servo, temperature sensor, pH sensor, and water flow sensor.	• Ensure the board is in a ready state: confirm there is 12V coming from the wall power supply. It should also be running for at least 60 seconds error free before verification. • Use a multimeter to check the output of the buck converter, and that the 12V has been stepped down to 5V. • Also ensure the 5V is within the bounds at the sensors and servo, and that they are operational via using a multimeter and manual readings/testing.

• The LDO should step down the 5V to at least 3.3V to the microcontroller.	• Ensure the board is in a ready state: confirm there is 12V coming from the wall power supply. It should also be running for at least 60 seconds error free before verification. • Using a multimeter, make sure the voltage is 3.3V at the ESP32 power pin.
• The circulation pump should receive at least 6V with a maximum of 18V directly from the power unit.	• Ensure the board is in a ready state: confirm there is 12V coming from the wall power supply. It should also be running for at least 60 seconds error free before verification. • Using a multimeter ensure the wall power source is 12V and that the circulation pump is 12V.
• Ensure all components receive the appropriate input voltage according to the table above.	• Ensure the board is in a ready state: confirm there is 12V coming from the wall power supply. Check with a multimeter all component's inputs.

2.4 Tolerance Analysis

Our main requirement for the system is that it must operate reliably for the selected AC-DC adapter, which is 12V, 6A (72W). To ensure that this AC-DC adapter is appropriate, we checked the worst-case power requirements for all our components. The following information can be found in the data sheets for each component.

3.3V Loads

- Temperature sensor
  - Takes max 1000nA for current.
  - Utilizes 3.3V supply.
  - P = 3.3 (1000nA) = 3.3uW
- ESP32
  - High peak current: 0.5A
  - Power: 0.5A * 3.3V = 1.65W
- LDO:
  - For an LDO, the power dissipation can be calculated as (VIN-VOUT) * Load Current. Since the main load current is the ESP32, our approximate dissipated power is (5-3.3V) * 0.5A = 0.85W.
- Total 3.3V Load: 1.65W + 0.85W = 2.5W

5V Loads

- SG90 Servo:
  - Stall current: 650 +- 80mA; Worst case: 0.73A
  - Worst case power: 5V * 0.73A = 3.65W
- Water flow sensor:
  - Max current: 15mA
  - Max power: 5V * 0.015A = 0.075W
- pH sensor:
  - No power information, but takes in 5V and using a similar example, it should take between 5-10mA [17]
  - Max power: 5V*10mA = 0.05W
- Accounting for the 2.5W load coming from the LDO and its corresponding loads, we get The total load power is 3.65W + 0.075 + 0.05W + 2.5W = 6.275W

12V Loads

- Water Pump:
  - Maximum input current: 0.5A
  - Maximum power: 12V * 0.5A = 6W
- LEDs and LED Driver:
  - Deep Red: Number of LEDs * Max LED current * VF Red Max = 5*0.3A*3.7V= 5.55W
  - Royal Blue: Number of LEDs * Max LED current * VF Blue Max = 5*0.3A*2.7V = 4.05W
  - Total power by LEDS: 9.6W
  - Overhead power: VIN quiescent current * 12V = 12V4mA = 48mW
  - Efficiency for VIN = 12V and switching frequency of 2MHz, we get around 91%.

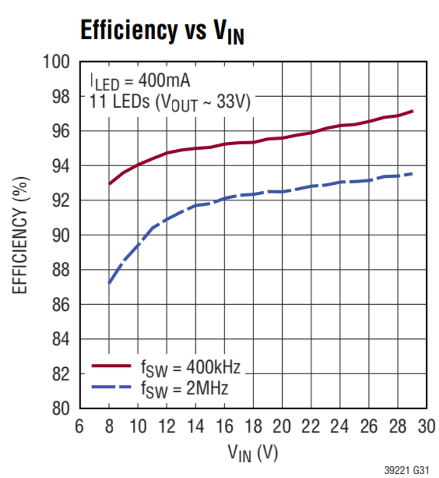


Figure 26: LED Driver Efficiency vs VIN table

- For the 500mOhm shunt resistor measuring the current through the LEDs, we can calculate the power it dissipates using P = I^2*R = (0.3) ^2*(0.5) = 0.045W
- Total power consumed: PIN = 9.6 / Efficiency + Overhead Power = (9.6W+ 0.045W)/0.91 + 48mW = 10.64W

- Lastly, we consider the load from the 5V calculated and its load of 6.275W, we can calculate the efficiency according to the following graph, in which the efficiency should be around 95%. Thus, the total power consumed by this is 6.275/ (0.95) = 6.6W.

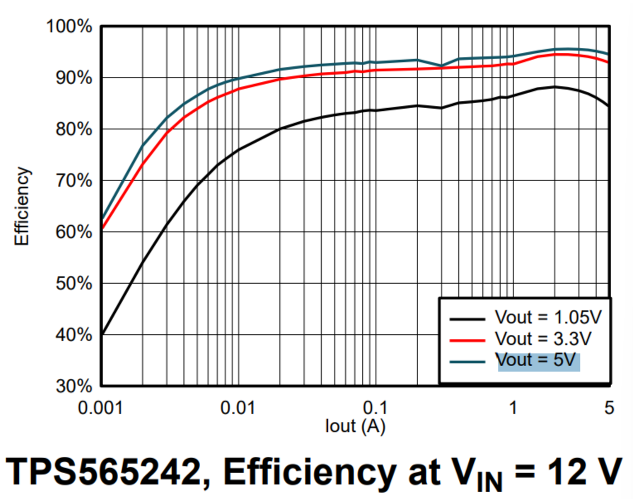


Figure 27: Buck converter Efficiency vs Iout

- Thus, summing everything up, we get total power for the main components of our board to come up with this:
  - Power total: 6.6W + 10.64W + 6W = 22.64W

Accounting for this, our power supply with 72W gives us plenty of overhead to account for any future expansion, any efficiency variation, and transient peaks from pump and servo startups and stall. This should also allow us to use Wi-Fi + Bluetooth in the future, which might require more power.

3. Cost & Schedule

3.1 Cost Analysis

3.1.1 Labor

A general estimate for each partner in the remaining weeks of this project is as follows. I am assuming a general salary for an ECE major based on the national average of $80,000 for entry level. This equates to about $39 (rounded up) per hour. We are estimating that this project will take about 10-15 hours per week for the 7 weeks remaining in this class. This comes to a calculation of 39 * 2.5 * 10 * 7 = $6825 per member. This time 3 comes out to a total of $20475.

On top of this, we are also using resources provided by the machine shop and estimate those services will be costing about $40 an hour and will take about 10 hours max. This adds an additional $400. So, the labor total for this project is about $20,875.

3.1.2 Parts

Description	Manufacturer	Quantity	Extended Price	Link
ESP32-WROOM-32E-N8R2 (MCU)	Espressif Systems	1	5.38	Link
LED Lighting Color Horticulture Red 731nm (720nm ~ 745nm) 1113 (2835 Metric)	Broadcom Limited	3	4.44	Link
LED HORTICULTURE BLU 451NM 2835	Broadcom Limited	5	6	Link
Liquid Flow Sensor 1 ~ 30 LPM Plastic	Seeed Technology Co., Ltd	1	9.50	Link
ANALOG PH METER KIT	DFRobot	1	29.50	Link
DC MINI IMMERSIBLE WATER PUMP 6V	DFRobot	1	9.20	Link
Waterproof 1-Wire DS18B20 Digital temperature sensor	Adafruit	1	9.95	Link
SERVOMOTOR RC 4.8V	DFRobot	1	3.62	Link
TERM BLK 3POS SIDE ENTRY 5MM PCB	Würth Elektronik	1	0.61	Link
12V 6A Power Supply Adapter	COOLM	1	14.59	Link
TVS DIODE 12VWM 19.9VC SOD123F	Littelfuse Inc.	2	0.46	Link
Zener Diode 12 V 500 mW ±5% Surface Mount SOD-123	Diodes Incorporated	1	0.10	Link
CAP CER 1UF 35V X7R 0805 *	Taiyo Yuden	10	1.6	Link

CAP CER 0.1UF 35V X7R 0603 *	Taiyo Yuden	15	1.65	Link
CAP ALUM 100UF 20% 35V SMD	KEMET	1	0.63	Link
CAP ALUM 47UF 20% 50V RADIAL TH	Rubycon	2	0.56	Link
CAP CER 10UF 25V X5R 1206 *	Taiyo Yuden	4	0.32	Link
CAP ALUM 220UF 20% 35V RADIAL TH	Rubycon	2	0.86	Link
CAP ALUM 470UF 20% 35V RADIAL TH	Rubycon	1	0.6	Link
CAP CER 0.033UF 50V X7R 0805	KYOCERA AVX	1	0.21	Link
CAP CER 0.47UF 50V X5R 0402	Murata Electronics	2	0.2	Link
CAP CER 2.2UF 25V X5R 0603	Taiyo Yuden	1	0.08	Link
CAP CER 10000PF 50V X7R 0603	KEMET	1	0.08	Link
1000 pF ±10% 50V Ceramic Capacitor X7R 0402 (1005 Metric) *	Murata Electronics	1	0.08	Link
CAP CER 22UF 25V X5R 1206	Taiyo Yuden	3	0.27	Link
FIXED IND 4.7UH 3A 37.6 MOHM SMD	Bourns Inc.	1	0.42	Link
FIXED IND 3.3UH 3A 30 MOHM SMD	Taiyo Yuden	1	0.23	Link
CONN PWR JACK 2.5X5.5MM SOLDER	Würth Elektronik	1	0.94	Link
DIODE SCHOTTKY 40V 2A DO214AA	Taiwan Semiconductor Corporation	1	0.38	Link
MOSFET N-CHANNEL 30V 3A TSMT3	Rohm Semiconductor	1	0.72	Link
FUSE BRDMT 10A 125VAC/125DC 2410	Bourns Inc.	1	1.18	Link

RES 30K OHM 1% 1/8W 0805	YAGEO	2	0.20	Link
RES 220K OHM 1% 1/4W 0805	YAGEO	1	0.11	Link
RES SMD 20K OHM 0.1% 1/8W 0805	YAGEO	1	0.10	Link
RES SMD 2M OHM 0.1% 1/10W 0805	TE Connectivity Passive Product	1	0.18	Link
RES 10K OHM 1% 1/5W 0603 *	YAGEO	6	0.66	Link
RES SMD 20K OHM 1% 1/10W 0603	YAGEO	1	0.11	Link
RES 1K OHM 1% 1/10W 0603 *	YAGEO	3	0.30	Link
RES 4.7K OHM 1% 1/10W 0603 *	YAGEO	1	0.10	Link
RES 220 OHM 5% 1/10W 0603	YAGEO	1	0.10	Link
RES 100K OHM 1% 1/10W 0603	YAGEO	2	0.22	Link
RES 100 OHM 1% 1/10W 0603	YAGEO	1	0.10	Link
RES 270 OHM 1% 1/10W 0603	YAGEO	3	0.30	Link
RES 1.1M OHM 1% 1/4W 1206	YAGEO	1	0.10	Link
RES 59K OHM 1% 1/8W 0805	YAGEO	1	0.10	Link
RES 100K OHM 5% 1/8W 0805 *	YAGEO	1	0.10	Link
RES 43.2K OHM 1% 1/10W 0603	YAGEO	1	0.10	Link
RES SMD 1M OHM 5% 1/8W 0603	YAGEO	1	0.28	Link
SWITCH TACTILE SPST-NO 0.05A 12V	Alps Alpine	2	0.62	Link
RES 0 OHM JUMPER 1/10W 0603	YAGEO	3	0.30	Link
RES 0.5 OHM 1% 1/2W 1206	Stackpole Electronics Inc	1	0.22	Link
RES 22.1K OHM	YAGEO	2	0.10	Link

1% 1/10W 0603				
RES 47.5K OHM 1% 1/10W 0603	Stackpole Electronics Inc	1	0.10	Link
RES 10K OHM 1% 1/10W 0603	YAGEO	3	0.30	Link
RES 5.1K OHM 1% 1/10W 0603	YAGEO	1	0.10	Link
Aqueon Standard Glass Rectangle Aquarium	Aqueon	1	14.99	Link

A star (*) means that these parts will be sourced from the EShop or the ECE Supply Center.

3.1.3 Grand Total

The total for this project is about $20,875 (labor costs) + 124.25 (parts cost) = $20,999.25.

3.2 Schedule

Week	Tasks	People
2/23 – 2/27	Finish PCB	Estela
	Order First Round PCB	Anjali
	Order Parts	Everyone
	Finish Design Document	Everyone
3/2 – 3/6	Prepare Design Review Presentation	Everyone
	Start Prototyping Board (Power & LED Drivers)	Anjali & Aishwarya
	Update PCB Design	Estela
	Order 2nd Round PCB	Anjali
3/9 – 3/13	Breadboard Demo	Everyone
	Third Round PCB Order	Anjali
	Update PCB	Estela
	Start MCU Code	Aishwarya
	Breadboard Sensors & Lighting	Anjali & Aishwarya
3/16 – 3/20	SPRING BREAK	
3/23 – 3/27	4th Round PCB Order	Estela
	Update PCB	Estela
	Connect Power subsystem to Sensors & Lighting subsystems	Everyone
	Work on water pump & fish feeder subsystems	Anjali & Aishwarya
3/30 – 4/3	Individual Progress Report	Everyone
	Start putting everything together	Everyone

4/6 – 4/10	**Progress Demo**	**Everyone**
	Continue working on code & debugging all subsystems	**Everyone**
4/13 – 4/17	**Continue working on code & debugging all subsystems**	**Everyone**
	Full assembly & Final bugs	**Everyone**
	Start working on Final Presentation and Final Report	**Everyone**
4/20 – 4/24	**Mock Demo**	**Everyone**
	Mock Presentation	**Everyone**
	Continue working on Final Presentation and Final Report	**Everyone**
4/27 – 5/1	**Final Demo**	**Everyone**
	Final Presentation	**Everyone**
5/4 – 5/8	**Final Papers**	**Everyone**
	Lab Notebooks	**Everyone**

4. Conclusion

4.1 Ethical considerations

4.1.1 Societal Impact

Our project creates an impact in various aspects of society. With the rising costs of fresh fruit and vegetables in the country, aquaponics provides an alternative to families to grow fresh produce with limited space and nutrient costs. [2] On top of providing a way to grow plants that can sustain families, it also generates a source of protein in the fish located in the aquarium underneath. Using aquaponics promotes a clean-living system and addresses worldwide concerns of water pollution and plastic waste that is affecting wildlife in the ocean today. Aquaponics, specifically on the smaller scale like our project, directly brings nutrient rich food to climates that might not be naturally suitable for that kind of plant/animal growth [13].

4.1.2 IEEE Code of Ethics

As a group, we are committed to following the IEEE Code of Ethics and incorporating it into our project. We are going to uphold ethical professional conduct standards as we conduct our senior design project, especially the following from the IEEE Code of Ethics:

1. “To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest, and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others” [9]

   To uphold this ethical standard, our senior design group has various avenues for taking in honest feedback and using it to correct our errors. First off, our team is committed to peer-reviewing each others’ work. When one teammate designs something or writes up a draft, the remaining teammates read over the work so that we have multiple people verifying validity and accuracy. Furthermore, we regularly communicate with our assigned project TA, who provides us with detailed feedback on our writing and project progress. Again, this acts as a peer-review source that ensures that our project has accurate technical information. Alongside this, we are committed to properly crediting contributions and claims from external sources. We regularly cite all of our sources and ensure that we keep a running record of external sources used so that we can clearly identify where we are getting our information from.

2. “To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations” [9]

   To uphold this ethical standard, our senior design group is committed to improving and growing our technical skills and only conducting technical work if we have appropriate training or supervision. Before first entering the lab, all our team members completed lab safety training to ensure that we knew how to use lab materials safety. Furthermore, all members of our group have completed our soldering training, and we are committed to following lab safety guidelines while soldering our PCB. We will wear goggles, use the flux fan, and conduct ourselves with care so that nobody is harmed or injured. Before using any new tools or technologies we have not used before, we will reach out to our TA to ensure that we are appropriately trained to use them and know how to use them in safe ways.

3. “To treat all persons fairly and with respect, and to not engage in discrimination based on characteristics such as race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression” [9]

   Our team is extremely committed to treating each other, and everyone around us, with respect and kindness. When we conduct our team meetings, we make sure every person’s voice is heard. We give each team member space to voice their opinions on decisions we are making, through asking for feedback from each member. At the end of our group meetings, we do a quick check-in to see how each member is feeling. If there are concerns or a teammate feels that they were overlooked, we talk about what changes can be made for next time and how we can improve our communication.

4.1.3 General Ethical & Safety Concerns

General ethical concerns for our project are the well-being of the fish in the fish tank located underneath. Due to the system not having an oxygen pump for the fish, we are relying on both the fish breed being resistant to low oxygen environments and using the falling water from the plant trough to create movement of the water on the surface to promote oxygen flow. Also, due to the nature of the project there is also potential for the fish to face harm from any electrical components not being sealed off from the water properly. Safety concerns regarding our project stem from the use of electronic components near a body of water. With any incorrect sealing, it could present a risk to both the fish inside the tank or anyone handling the tank to replace the water, checking the status LEDs, etc. Over time, any sealant may also degrade, so that is also a concern we are considering when building our project.

4.1.4 Mitigating Safety Concerns

In order to mitigate the risks of our project, we are addressing it on several levels. Our main concern is the water with the circuit components and the risks that might occur in terms of the two of them interacting with each other. We are working to mitigate this risk by working with the machine shop to have the controls printed circuit board located underneath the fish tank so that there is a level of separation of the two. Everything that needs to interact with the water such as the water flow sensor, pH sensor, and temperature sensor, we have organized to have a side acrylic compartment that extends a little bit into the water. Another safeguard that we have put into place is due to the high voltage that is required by the growth LEDs. The LEDs are very prone to overheating and due to this, we have created an LED driver to regulate the voltage and have expanded more about this in our tolerance analysis. Our design follows standards set by IEC 60529.

References

1. Aqueon. "Betta." *Aqueon*, https://www.aqueon.com/resources/care-guides/betta. Accessed 13 Feb. 2026.

2. Beecher, Lance. "Introduction to Aquaponics." *Land-Grant Press | Clemson University, South Carolina*, lgpress.clemson.edu/publication/introduction-to-aquaponics/.

3. Broadcom Limited. "ASMW-LL00-NKM0E." *DigiKey*, www.digikey.com/en/products/detail/broadcom-limited/ASMW-LL00-NKM0E/9740413. Accessed 13 Feb. 2026.

4. COOLM. "12V 6A Power Supply Adapter - AC Adapter 100–240V 50–60HZ DC 12 Volt 6A 72W Power Converter Transformer Charger 6amp 5.5mm x 2.5mm DC Plug for LED Strips Lighting, Router CCTV Camera COOLM." *Amazon*, www.amazon.com/12V-Power-Supply-Adapter-Transformer/dp/B07DMFN2YN. Accessed 13 Feb. 2026.

5. DFRobot. "FIT0563." *DigiKey*, www.digikey.com/en/products/detail/dfrobot/FIT0563/8827828. Accessed 13 Feb. 2026.

6. DFRobot. "SEN0161." *DigiKey*, www.digikey.com/en/products/detail/dfrobot/sen0161/6579368. Accessed 13 Feb. 2026.

7. DFRobot. "SER0006." *DigiKey*, www.digikey.com/en/products/detail/dfrobot/SER0006/7597224. Accessed 13 Feb. 2026.

8. Espressif Systems. "ESP32-S3." *DigiKey*, www.digikey.com/en/products/detail/espressif-systems/ESP32-S3/15822445. Accessed 13 Feb. 2026.

9. IEEE. (2020). *IEEE Code of Ethics | IEEE*. Ieee.org.

   https://www.ieee.org/about/corporate/governance/p7-8. Accessed 12 Feb 2026.

10. Seeed Technology Co., Ltd. "314150005." *DigiKey*, www.digikey.com/en/products/detail/seeed-technology-co-ltd/314150005/5488047. Accessed 13 Feb. 2026.

11. STMicroelectronics. "STM32G431KBT6." *DigiKey*, www.digikey.com/en/products/detail/stmicroelectronics/STM32G431KBT6/10231564. Accessed 13 Feb. 2026.

12. Texas Instruments. "TPS565242DRLR." *DigiKey*, www.digikey.com/en/products/detail/texas-instruments/TPS565242DRLR/16585653. Accessed 13 Feb. 2026.

13. United Nations. "Aquaponics: Ancient Wisdom for New Food Production." *United Nations Western Europe*, 5 July 2021, unric.org/en/aquaponics-ancient-wisdom-for-new-food-production/.

14. "Waterproof 1-Wire DS18B20 Digital Temperature Sensor." *Adafruit*, www.adafruit.com/product/381. Accessed 13 Feb. 2026.

15. "10x Heavy Duty Mesh Pot Net Cup Basket Plastic Hydroponic Planting Mesh Net Pot Baskets Garden Plant Grow Cup." *Https://Www.amazon.com/CoscosX-Plastic-Hydroponic-Planting-Baskets/Dp/B071H51DRZ?Th=1*, Amazon. Accessed 27 Feb. 2026.

16. Mullet Tools. "4-Inch Thin Wall Furniture Grade PVC Pipe (Light Gray)." *Mullet Tools*, 2026, mullettools.com/products/4-inch-thin-wall-furniture-grade-pvc-pipe-light-gray?srsltid=AfmBOooB2E2_wTROrZeA-FXmLRjT-pVKQeHeAogrwLPC86nX5K8ByoA05ec. Accessed 27 Feb. 2026.

17. https://robocraze.com/blogs/post/all-about-ph-sensor#:~:text=This%20pH%20sensor%20is%20a,today%2C%20some%20of%20them%20include: