

**ECE 445: SPRING 2026**

---

**CIRCLE OF LIFE: AUTOMATED  
DESKTOP AQUAPONICS SYSTEM**

By

Anjali Aravindhan

Aishwarya Manoj

Estela Medrano-Gutiérrez

Final Report for ECE 445, Senior Design, Spring 2026

TA: Manvi Jha

May 2026

Project 73

## **Abstract**

This final report is a short documentation that explains the process of designing, testing, verifying, and demonstrating the project Circle of Life: Smart Desktop Aquaponics System. This project is the senior design project for Anjali Aravindhan, Aishwarya Manoj, and Estela Medrano-Gutierrez and was completed in the Spring 2026 semester of ECE 445: Senior Design at the University of Illinois Urbana-Champaign under the supervision of TA Manvi Jha and graded by Professor Joohyung Kim.

# Contents

1. Introduction .....	1
2. Design.....	2
2.1. Design Procedure .....	2
2.2. Design Details.....	4
2.2.1 Physical Design.....	4
2.2.2 Subsystem Overview.....	7
3. Verification.....	13
3.1 Fish Feeder Subsystem.....	13
3.2 Water Quality Subsystem .....	14
3.3 Lighting Subsystem .....	15
3.4 Water Pump Subsystem.....	17
3.5 Power Subsystem.....	17
4. Costs.....	18
5. Schedule.....	19
6. Conclusions .....	19
6.1 Accomplishments & Future Work.....	19
6.2 Societal Impact & Ethic .....	19
7. References .....	21
Appendix A: Requirements & Verification Table .....	23
A.1. Water Quality R&V Table.....	23
A.2 Water Pump R&V Table .....	25
A.3 Lighting R&V Table.....	26
A.4 Power R&V Table .....	27
A.5 Fish Feeder R&V Table.....	29
Appendix B: Costs .....	31
B.1 Parts Table .....	31
Appendix C: Design .....	35
C.1 Original Design Documentation Block Diagram.....	35
C.2 LED Board Circuit Schematic.....	35
C.3 LED Driver Circuit Schematic.....	36
C.4 Water Pump Circuit Schematic .....	36

C.5 Power Input Circuit Schematic.....	36
C.6 Buck Converter Circuit Schematic.....	37
C.7 LDO Circuit Schematic.....	37
C.8 Sensor Inputs Circuit Schematic .....	37
C.9 Microcontroller Circuit Schematic .....	38
C.10 USB-to-UART Circuit Schematic .....	38
C.11 Servo Motor Circuit Schematic .....	38
C.12 First Round Main Power & Control PCB Layout.....	39
C.13 First Round LED PCB Layout.....	39
Appendix D: Weekly Schedule Table .....	40

## 1. Introduction

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 [2].

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, while a microcontroller processes sensor data and schedules feeding and lighting operations.

The system is designed for younger urban apartment residents and students with an interest in convenience, aesthetics, wellness, and/or sustainability but has 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.

Based on these functional goals, our system is divided into several major subsystems, including the fish feeder subsystem, water pump subsystem, the water quality and controls subsystem, the lighting subsystem and the power subsystem, as shown in Figure 1. Compared to our original design, shown in Appendix C.1, the only change we made is that we used LEDs with only one string rather than LEDs with two.

To ensure reliable and autonomous operation, the system is designed to meet several quantitative performance requirements. Our high-level requirements are thus the following:

- 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

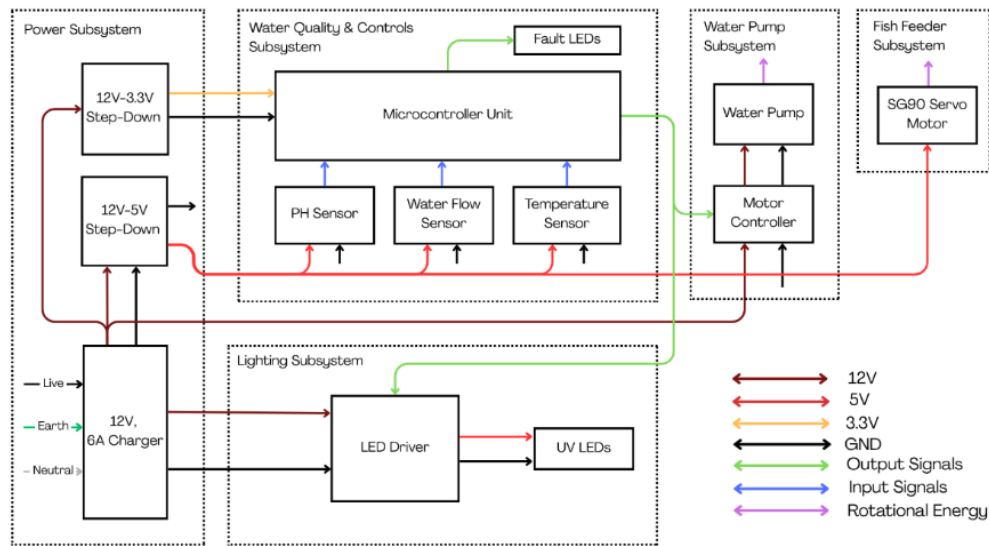


Figure 1: Block diagram with separate modules

## 2. Design

### 2.1. Design Procedure

The design was divided into two parts: the physical and electrical component. For the physical component, we originally decided to use a 5-gallon tank due to pricing reasons, but after talking to the machine shop, we decided to change it to a much smaller 0.8-gallon tank, since the other one would be over 10 lbs. with water. The plant bed is on top of the tank, with LED growth lights placed on top of the plants. We then have tubing going from the fish tank to the plant bed, and we utilize the water pump to move the water up to the plant bed. Next to the plant bed, our sensors are placed so that they hover over the tank bed, with only the required part submerged in the water.

For the electrical component, we decided to incorporate all power and control components into the main PCB shown in Figure 2. This PCB was placed on a water-proof board for protection on the base of the tank. The purpose of this PCB is to connect to all sensors, host the microcontroller to send signals to the sensors, and distribute power to all required components. Due to the need to have the LEDs on top of the plant bed, we decided to have an external PCB, shown in Figure 3, for the LEDs alone, which get its power and control signal from the main PCB.

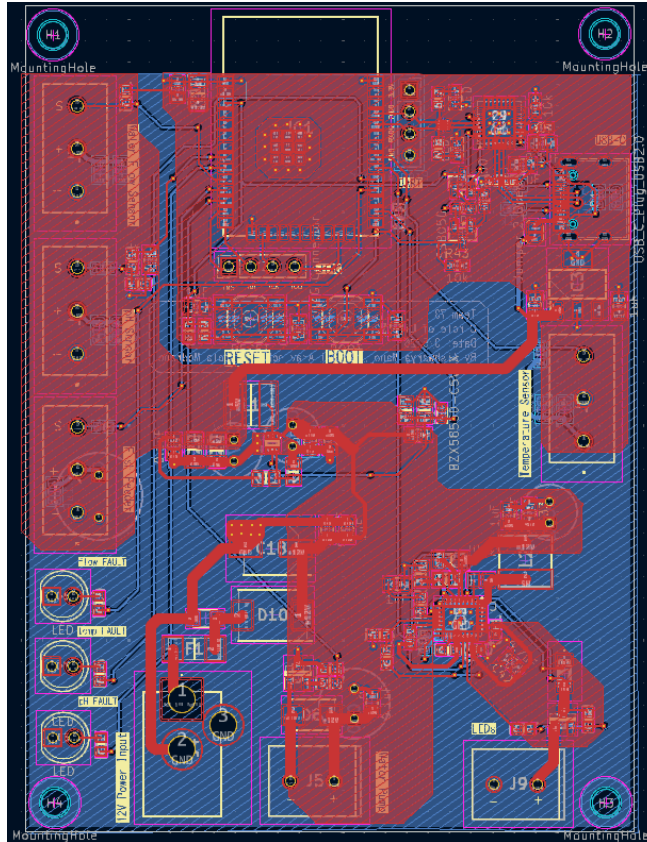


Figure 2: Final PCB Layout of Main Power & Control Board

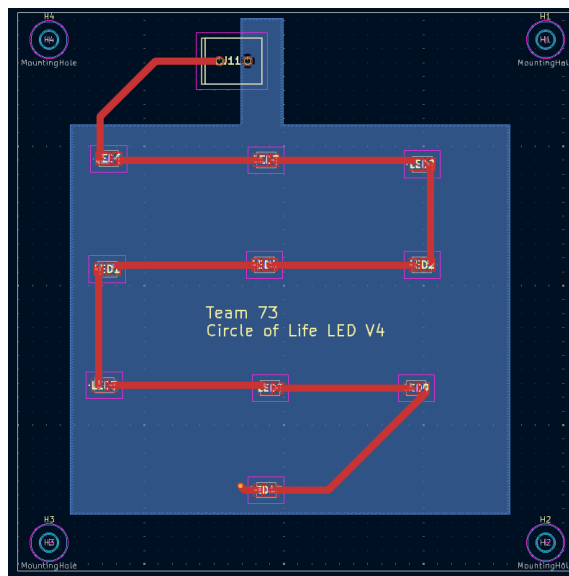


Figure 3: Final PCB Layout of LED Board

## 2.2. Design Details

### 2.2.1 Physical Design

#### 2.2.1.1 Aquaponics Fish Tank & Grow Bed

The physical design centers on a compact support structure made from T-slot aluminum extrusions, which mounts the grow bed, fish feeder, LED grow lights, and PCB enclosure. The grow bed consists of a 4" PVC pipe [11] configured as a nutrient film technique (NFT) channel, tilted at approximately 2% slope so that water flows continuously back into the fish tank by gravity. Water is pumped from the tank into the left side of the channel through an inlet hole, while the water flow sensor is mounted on a dedicated support adjacent to the structure, as shown in Figure 4. The PVC channel holds 3 plant cups of 3.4 cm diameter each, and the fish feeder is mounted at the back of the PVC holder on the support structure [17].

The final design shown above differs from the original concept whose preliminary 3D CAD draft is shown in Figure 5. The tank was reduced from the originally planned 16.5" × 8.75" × 10.75" (approximately 5 gallons) to a 0.8-gallon unit following consultation with the machine shop. The wooden support structure was replaced with T-slot aluminum extrusions for improved rigidity, and the PVC pipe is secured to the plastic top of the fish tank with hot glue rather than pipe clamps. Temperature and pH sensors are inserted through dedicated holes in the tank rather than being externally attached, and a plastic base was added to the bottom of the assembly to house the tank and controls.

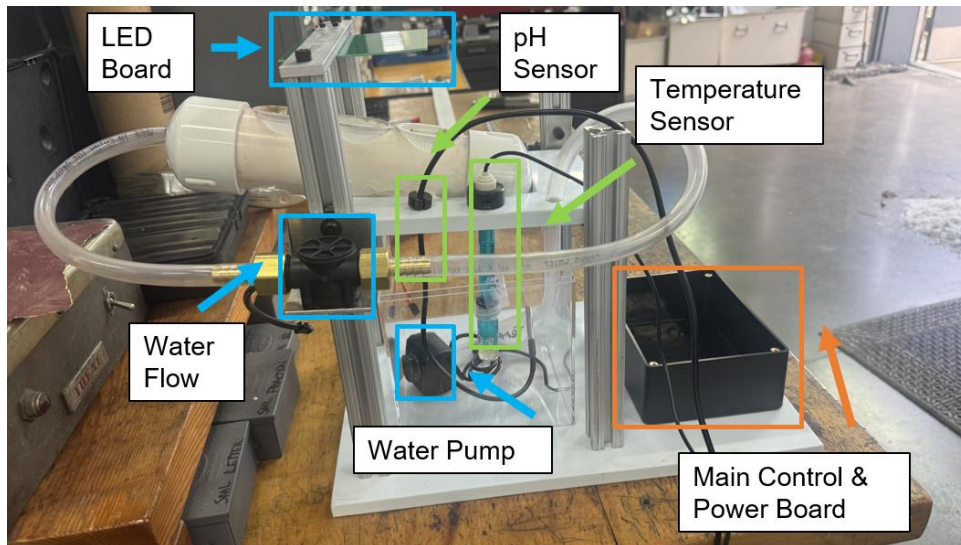


Figure 4: Complete physical design (left side) with labels for each subsystem placement

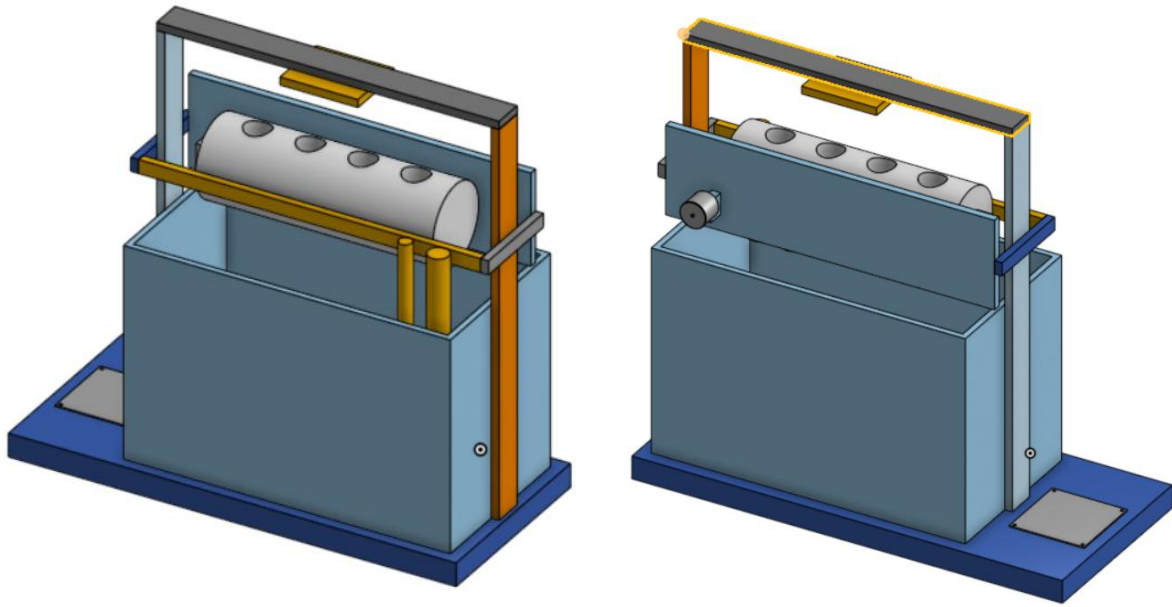
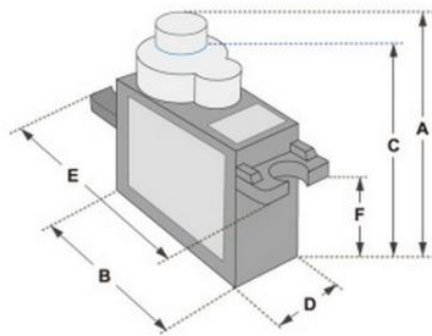


Figure 5: 3D CAD of preliminary original physical design for 5-gallon

### 2.2.1.2 Fish Feeder 3-D Printed Shaft



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

For the fish feeder, a 3D-printed enclosure attaches to the shaft of the SG90 servo motor using the dimensions shown in Figure 6. Common pellet sizes for small fish such as betta range from 0.5 mm to 1.5 mm, so the slit in the 3D-printed shaft is sized to accommodate this range [1]. The shaft consists of two parts: a lid that attaches to the servo motor's rotating structure, shown on the left in Figure 7, and the holder itself, shown on the right. Both parts are 3D-printed in PLA using a Prusa printer.

The servo motor mounts to the support structure, and a hole in the plastic tank top allows food pellets to fall through when the shaft rotates. Figure 8 shows the placement of the fish feeder on the assembled system.

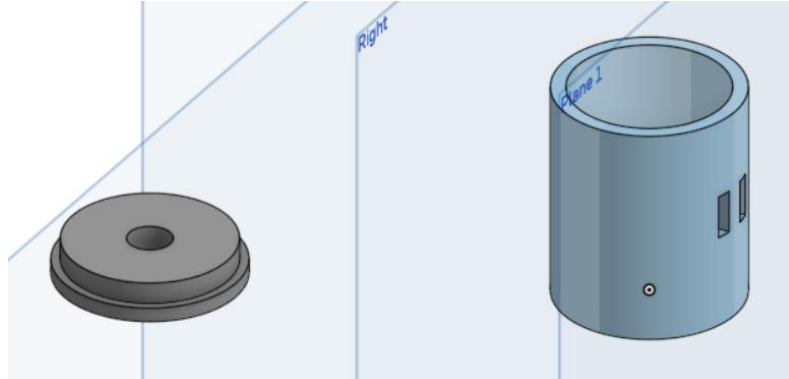


Figure 7: 3D-printed lid (to the left) and shaft (to the right)

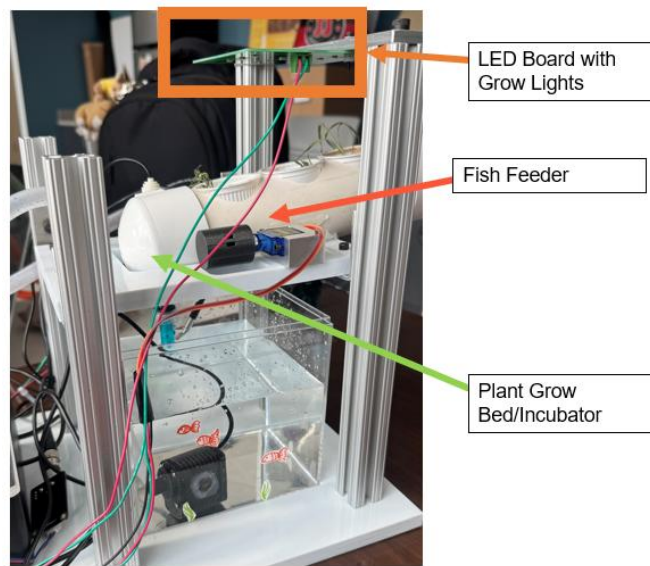


Figure 8: Complete physical design (right side) with labels for each subsystem placement

### 2.2.1.3 Main and LED PCB Board

Both the main power and control PCB and the LED PCB are smaller than 100 mm × 100 mm due to budget constraints. The main board includes 3 mm mounting holes on each corner, while the LED board includes mounting holes of the same size on the middle left and right sides, allowing it to mount to the aluminum extrusion above the system as shown in Figure 4. Both boards are housed in 3D-printed PLA enclosures. The main board enclosure is fully closed, with openings only for the DC barrel jack, USB-C port, and 2-wire to 3-wire through-hole terminal blocks. The LED enclosure is open on the LED-facing side, primarily to prevent direct contact between the PCB and the aluminum extrusion above if moisture

escapes from the tank. The enclosure body is 3D-printed, and the top panel is made from acrylic secured with screws.

All four rounds of PCB ordering were used over the course of the semester, producing three versions of the main power and control board and three versions of the LED board. The final designs are shown in Figures 2 and 3, and the original first-round designs are shown in Appendix C.12 and Appendix C.13.

## 2.2.2 Subsystem Overview

### 2.2.2.1 Fish Feeder Subsystem

The fish feeder is implemented using a servo motor that rotates between two angular positions: one away from the tank and one toward it for dispensing food. A custom 3D-printed food container couples to the servo shaft via screws and includes a small outlet opening that allows food to fall through when the container rotates downward.

In Appendix C.11 you can see our schematic design for the servo motor circuit. The servo motor operates at 5V, within its 4.8–6V range and is controlled via a PWM signal from a GPIO pin on the microcontroller [7][8], which also manages the other subsystems [7]. Since the microcontroller outputs a maximum of 3.3V but the servo expects 5V PWM signals, a level shifter steps the signal up accordingly. After breadboard testing, we realized this level shifter was not necessary since the servo worked correctly with a 3.3V PWM signal, so we removed it from our original design.

To ensure stable operation, an electrolytic capacitor on the 5V line suppresses current spikes during startup and load variation, a 0.1  $\mu$ F capacitor filters high-frequency noise, and a series resistor reduces ringing and overshoot on the PWM line caused by wire length. The resistor value may be adjusted based on testing results. We did not include the 470 $\mu$ F capacitor in the actual PCB because we thought it was excessive for the application, however it was kept in the PCB in case it was needed.

### 2.2.2.2 Water Quality & Control Subsystem

This subsystem monitors water quality and circulation using a water flow sensor, a pH sensor, and a temperature sensor. Because the pump lifts water to the grow bed, monitoring flow confirms that water is reaching the plants and helps detect faults such as clogged tubing, a disconnected hose, or an empty tank. The controller measures flow rate once per hour to verify that it stays within 3–5 $\times$  the total tank volume per hour. If the measured flow deviates from this range, the controller either adjusts the pump's PWM signal within safe limits or triggers the fault indicator LED and halts the pump [13]. The flow sensor is powered at 5V, with its output pulled up to 3.3V to keep the logic-high level safe for the ESP32.

The pH sensor measures water acidity to ensure conditions suitable for both the fish and plants. Betta fish require a pH of roughly 6.8–7.5, which also falls within the slightly acidic to neutral range that the plants require. If the pH falls outside this range, an LED alerts the user to change the water [1][6]. The sensor outputs an analog signal up to 5V, where the voltage in millivolts corresponds directly to pH

level[12]. This signal is read through one of the MCU's ADCs and stepped down from 5V to 3.3V via a voltage divider before reaching the microcontroller.

The temperature sensor monitors water conditions to ensure the tank remains habitable for the fish, which requires a temperature between 76 and 85 °F for betta[1]. If the temperature falls outside this range, an LED alerts the user to adjust the water or room temperature [16]. The sensor outputs a digital signal with selectable resolution between 9 and 12 bits, connected directly to a GPIO pin pulled high to 3.3V, so the line is only pulled low when the sensor actively drives it. Decoupling capacitors are included to filter high- and mid-frequency noise and keep the sensor supply clean and stable. Appendix C.8 shows the circuit schematic of the sensor inputs.

The microcontroller, whose circuit schematic is shown in Appendix C.9, is an ESP32-WROOM-32E-N8R2, shown in Figure 12. JTAG is included for debugging, and a combination of BOOT and RESET buttons along with a UART connector allows the MCU to be programmed per data sheet. Sensor signals, fault LEDs, and enable signals connect to their respective GPIO and ADC pins, while the programming connectors use their dedicated pins. While testing, we realized we mis-wired some of the fault LEDs by putting them in pins that were input only, making it impossible to turn them on. This is one of the things we fixed in our second operation.

A USB-to-UART bridge, shown in Appendix C.10, allows the MCU to be programmed via USB-C rather than directly through the UART pins. While the datasheet informed this portion of the schematic, the design is largely based on the ESP32 DevKitC V4 schematic. Given that programming the MCU is critical to the project's functionality, using a proven design was the most reliable approach. The datasheet was reviewed thoroughly to confirm that the dev kit's functionality applied to this design, and ESD protection diodes were added to protect the MCU from static discharge.

The DevKitC V4 schematic uses an auto-reset and boot circuit for flashing the ESP32. The DTR and RTS signals from the USB-to-UART bridge drive two transistors that pull down the EN and BOOT pins. When EN transitions from low to high while BOOT is held low, the ESP32 enters programming mode. The truth table for this circuit is shown in the bottom left corner of Appendix C.10. LEDs are also included to indicate when programming is active.

The primary difference from the DevKitC V4 is the use of a USB-C connector rather than Micro-USB. For USB-C, the CC line is pulled to ground with a 5.1 kΩ resistor as required by the standard to negotiate current availability from the upstream port.

Unfortunately, our UART-to-USB bridge never worked. Since it wasn't part of our requirements, we went most of the course without utilizing it. Instead, we used a USB-to-UART programmer and directly utilized the UART pins for programming.

### 2.2.2.3 Lighting Subsystem

The lighting subsystem provides artificial light for the plants in the desktop aquaponics system. Its purpose is to deliver the correct light intensity and duration each day to promote plant growth in the absence of natural sunlight. The subsystem uses horticulture LEDs in deep red and royal blue wavelengths to simulate sunlight and promote photosynthesis, mounted above the grow bed and fish tank to shine directly onto the plants [3].

The final LED configuration uses 5 red LEDs and 2 blue LEDs in series, with the following forward voltage characteristics:

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

Adding the forward voltages gives a total string voltage shown below:

- For the blue LEDs,  $2 \times 3.08 = 6.16V$
- For the red LEDs,  $5 \times 2.25 = 11.25V$
- In total,  $6.16V + 11.25V = 17.41V$

Originally, we decided on 10 red LEDs and 10 blue LEDs as in Appendix C.2., but we noticed that the blue LEDs were much brighter than the red LEDs at the same current level, so we decided to reduce the number of blue LEDs instead. The original output voltage was around 26.65V

With PWM dimming, the average voltage measured across the string is lower than this theoretical value. A constant-current LED driver was chosen to boost the input voltage to the level required to drive the string at a controlled current, while also supporting PWM dimming for scheduling and over-voltage and over-current protection for safety.

The original LED driver selected was the LT3922, a step-up converter supporting LED strings up to 34 V and input voltages from 2.8 to 36 V. The schematic, shown in Figure 15, follows the application notes in the datasheet, with component values adjusted for the design's requirements. It was designed for the original 10 LEDs instead of 7. The schematic, shown in Appendix C.3, was designed largely following the datasheet application notes, with custom values calculated where needed. Figure 9 shows the application example used as a reference, and the relevant equations are taken directly from the datasheet.

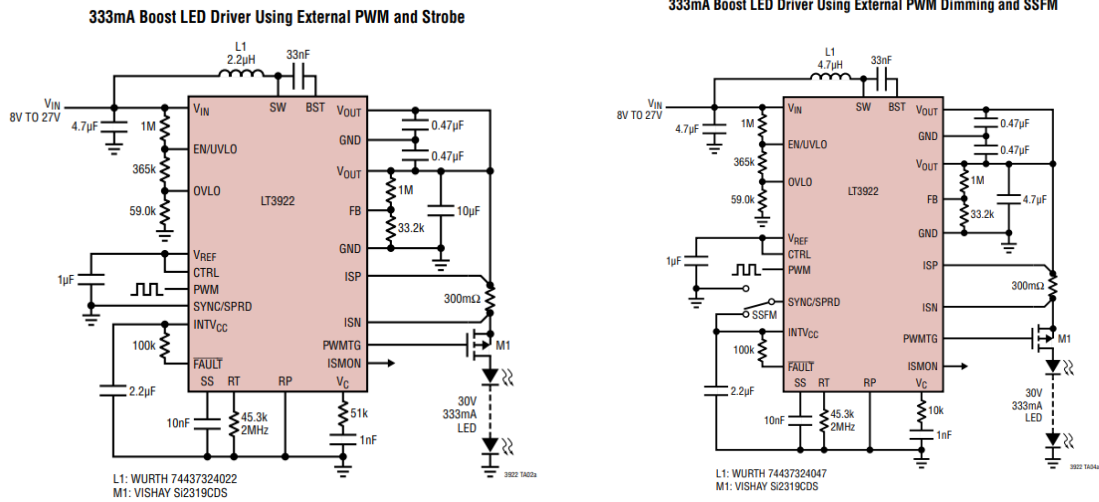


Figure 9: Application notes for LT3922

For the VC pin, the datasheet's recommended starting values of 1 nF and 10 kΩ were used. For the RT resistor, a higher switching frequency was chosen to allow a smaller inductor that fits within the PCB footprint, using the same frequency as the datasheet application example to keep the inductor and decoupling capacitance consistent. The frequency table is shown in Figure 10.

**Table 1. RT Resistance Range**

SWITCHING FREQUENCY	RT
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 10: Frequency table for RT resistor

ISMON is left unconnected, as current monitoring is not required for this application. SYNC is tied to ground, as an external clock is not used. For OVLO and UVLO, resistor values were selected per the datasheet equations, shown in Figure 11, to set a minimum input voltage of approximately 10 V and a maximum of approximately 15 V.

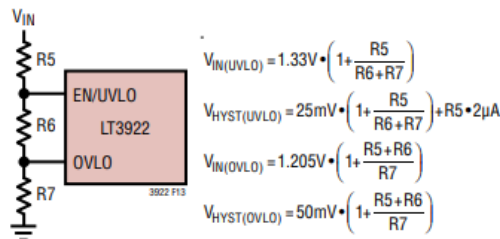


Figure 11: OVLO and UVLO resistor divider equations

The FB resistors, shown in Figure 12, set the output voltage based on the following maximum forward voltages. Keep in mind, these were calculated for our original planned forward voltage of 26.65V.

- For the blue LEDs,  $2 \times 3.7 = 18.5V$
- For the red LEDs,  $5 \times 2.7 = 13.5V$
- In total,  $18.5V + 13.5V = 32V$

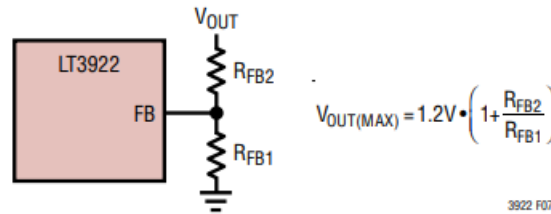


Figure 12: FB resistor divider equations

The target output voltage was set to approximately 20 V to provide margin for forward voltage variation while staying within the 40 V rating of the FET. For the shunt resistor, the value was scaled from the datasheet's 300 mΩ resistor for 333 mA to set the current to 200 mA using the following equation:

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

The LED board is mounted separately at the top of the grow bed, while the remaining power circuitry is housed at the base of the system for accessibility and cable management.

During bring-up, LT3922 was found to be non-functional due to a mismatch between the schematic symbol and the PCB footprint of the PMOS used to dim the LED voltage, which will be explained further in the verification section. Both available ICs were damaged during debugging, and the design was switched to a backup driver, the LDH-25-250W [10], which has a current output of 250 mA. Using a 3.3 V PWM signal, the current is limited to approximately 40% of the total output current, as shown in Figure 13.

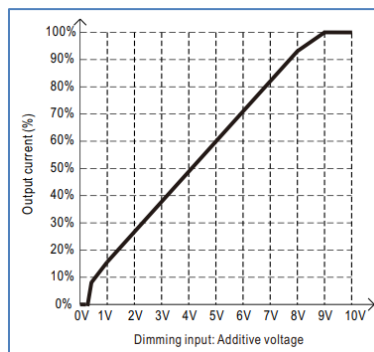


Figure 13: Graph of Output current (%) Vs Dimming input, applying additive 0 ~ 10VDC

#### 2.2.2.4 Water Pump Subsystem

The water pump, as seen in Appendix C.4, sits in series with the water flow sensor, circulating water from the fish tank up to the grow bed. A waterproof circulation pump is used, and its speed is controlled by PWM-modulating the supply voltage through an N-MOSFET configured as a low-side switch [5]. When the microcontroller drives the gate high, the MOSFET turns on and completes the circuit, allowing the pump to run. Speed is determined by the PWM duty cycle, since the motor responds to the average voltage, which depends on both the supply voltage and the duty cycle.

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

Several protection and filtering components are included to ensure reliable operation. A flyback diode across the motor provides a recirculation path for the inductive current when the pump switches off, preventing voltage spikes on the drain that could damage other components. A bulk capacitor and a decoupling capacitor help manage current spikes during startup and filter high-frequency noise from the supply. A series gate resistor limits the current driven into the gate from the MCU, and its value may be adjusted if switching speed proves insufficient. A pulldown resistor ensures the gate is never left floating, keeping the pump off when the system is not actively driving it.

An error that we noticed here is that, similarly as with the LED driver, there was a mismatch between the symbol and the footprint of the NMOS, making it so that the gate was at the source, the source at the drain, and the drain at the gate. We were able to fix it by rotating it in the PCB.

#### 2.2.2.5 Power Subsystem

The power subsystem supplies regulated voltage to all other subsystems, including the lighting, fish feeder, water quality monitoring, and pump. The system is powered from an external AC-DC 12 V, 6 A adapter, which is stepped down to 5 V and 3.3 V as needed by each subsystem [4].

The 12 V input stage, shown in Appendix C.5, includes a 10 A fuse for overcurrent protection, a TVS diode to clamp input voltage spikes, and a Schottky diode for reverse polarity protection. The Schottky diode is included as a precaution, though the DC barrel jack connector makes reverse polarity unlikely in practice, in which case it can be shorted. Bulk and decoupling capacitance are also included on the 12 V rail to maintain stability under load changes from the pump and LED driver.

A buck converter, shown in Appendix C.6, steps the voltage down from 12 V to 5 V with a current output capability of up to 5 A. The high current output is the primary reason a buck converter is used here rather than other conversion methods, as this rail supplies several higher-current components. Buck converters also handle load variations well, since the feedback network keeps the output voltage stable. Decoupling capacitance is included for the same reasons as mentioned in the previous sections, but not all of it was utilized in the physical design, specifically the electrolytic capacitors were not used. R4 serves as a pulldown on the EN signal when driven by a GPIO from the microcontroller, which was not

utilized. Instead, R3, R7, and D3 were used to tie EN permanently high. All component values were selected by the datasheet, and the table used to select values for the 5 V output is shown in Figure 14.

Finally, an LDO, shown in Appendix C.7, steps the 5 V rail down to 3.3 V for the MCU and temperature supply. The LDO was chosen for its simplicity and suitability for the MCU's lower current requirements, and its efficiency is adequate for the small voltage difference involved.

**Table 8-2. Recommended Component Values**

Output Voltage (V)	R1 (kΩ)	R2 (kΩ)	Minimum L1 (μH)	Typical L1 (μH)	Maximum L1 (μH)	Minimum C <sub>OUT</sub> (μF)	Typical C <sub>OUT</sub> (μF)	Maximum C <sub>OUT</sub> (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 14: Graph of Output current (%) Vs Dimming input, applying additive 0 ~ 10VDC

### 3. Verification

#### 3.1 Fish Feeder Subsystem

```

09:15:46.210 -> -----
09:15:46.334 -> -----
09:15:46.399 -> Temperature: 74.19°F
09:15:46.399 -> Voltage:1.94 pH value: 7.44
09:15:51.072 -> Flow rate: 120.00 L/hr
09:15:51.203 -> -----
09:15:51.354 -> -----
09:15:51.418 -> Temperature: 74.19°F
09:15:51.449 -> Voltage:1.94 pH value: 7.44
09:15:56.068 -> Fish feeder rotation started.
    
```

First Rotation

```

09:17:00.543 -> -----
09:17:00.825 -> -----
09:17:00.857 -> Temperature: 74.07°F
09:17:00.889 -> Voltage:1.94 pH value: 7.45
09:17:05.494 -> Flow rate: 120.00 L/hr
09:17:05.526 -> -----
09:17:05.859 -> -----
09:17:05.890 -> Temperature: 74.07°F
09:17:05.923 -> Voltage:1.94 pH value: 7.45
09:17:10.488 -> Fish feeder rotation started.
    
```

Second Rotation

Figure 15: Fish Feeder Rotations

The fish feeder subsystem has three main requirements found in Appendix A Section A.5: the servo motor should receive a PWM signal from the microcontroller, once the signal is received and it's nonzero it should rotate 180° to dispense fish food and then rotate back 180° to it's original position, and rotate once every 24 hours. In order to verify these requirements we used the main unit test of sending a nonzero PWM signal to the servo motor through utilizing the ESP32Servo library to the write() functions to move the servo 1° at a time until the full 180° was completed in both directions. In Figure 15, we used the prints on the Arduino Serial Monitor to show that the motor runs once in a specified amount of time, in this case using once per minute as the unit test. Figure 15 show the first rotation at approximately 9:15 and 56 seconds and the next rotation of the servo motor at 9:17 and 48 seconds. Due to this being approximately a minute later, we concluded that we proved the verification of the servo motor running once every 24 hours. With visual confirmations, we were able to confirm that during these rotation we saw the fish feeder's 3D printed shaft move 180° in each direction to confirm the rotation and that the servo motor was receiving a PWM signal.

It is suspected that the rotations are not exactly one minute later from the original rotation due to the integration with the code for other sensors. The code follows a blocking structure, and often will have to complete tasks for other subsystems before checking how much time has passed for the servo motor, leading to the slight delay shown in Figure 15. In all, all of the requirements were met for the fish feeder subsystem and were verified visually and via the Arduino IDE serial monitor outputs.

### 3.2 Water Quality Subsystem

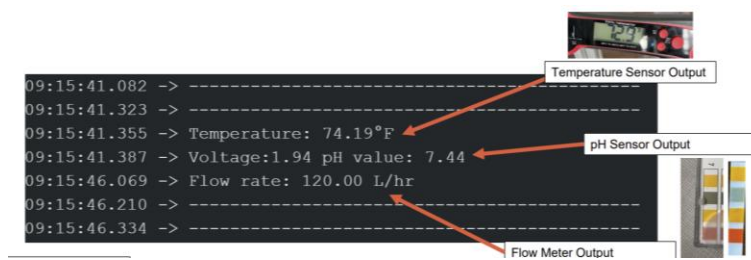


Figure 16: Water Quality Subsystem Serial Monitor Output

The water quality subsystem has general requirements for all three sensors that are included. All are required to be accurately showing the pH, flow rate, or temperature of the water and showing abnormal readings through sending a fault signal to the LEDs to alert the user of atypical readings. In Figure 16, there is a standard Arduino IDE serial monitor output and the corresponding verifications of accuracy for temperature and pH.

As shown in Figure 16, the Arduino IDE serial monitor was displaying a temperature of 74.19°F, when in reality it was 72.9°F. This falls in our requirements range of accuracy of the temperature sensor being within 5°C within the actual temperature of the water. This value is also outside of the temperature range that is ideal for a fish, which is 78-80°F, and we confirmed by visually seeing the fault LED light up at this temperature.

The pH verification is also shown in Figure 16. The tank was filled with tap water and the pH was verified as being around 7 through using VWR® pH indicator test strips as shown in the pH sensor output label in Figure 16. From there, the serial monitor output of a 7.44 pH was within the bounds of our verification of the pH being accurate within +/- 0.5 of the actual pH. For the fault LEDs, we were able to visually see the fault LEDs turn on when we tested the pH sensor in alkaline water, which has a pH of 9.5, since the normal pH for a fish ranges from 6-8.

The flow rate meter accuracy was checked through code and through using dimensional analysis. The code started off for the first 30 seconds with the PWM duty cycle set to 0. In this time, we visually verified that the flow meter fault LED was turned on, since at this point the flow is 0. To verify the accuracy, our original plan was to time how long it would take to move the water from the tank to an external container and use the time to calculate the overall flow rate. However, after the tank was built by the machine shop, this was no longer possible to do with the tubing being held in place on the tank. Instead, we opted to how long it would take to fill one cup of water and extrapolate that result using dimensional analysis to liters per hour. One cup for the flow rate shown in Figure 16 took about 9.8 seconds to fill completely. From there we converted to liters per hour by taking 1 cup/7.1 seconds =

0.236588 Liters / 7.1 seconds resulting in the total liters per second. This time 3600 results in total liters per hour as 119.96 L/hr. This is within our accuracy requirement of the error being within 3% with the error being  $|119.96 - 120| / 119.96 = 0.03\%$  error.

### 3.3 Lighting Subsystem

Area checked	What I tested/measured	Result	Conclusion
Switching node	Measured SW	Flat at 12V, no switching	No switching
Feedback divider	FB = 1M / 43.2k and measured FB	FB = 0.5V	Low value, meaning output is below target voltage, aka not boosting
Current control	CTRL tied to VRED with 1uF and measured CTRL	CTRL = 1.8V	Valid value
Enable/protection thresholds	Measured EN/UVLO and OVLO	In valid range	Normal operation
Fault output	Measured FAULT	High	No open/short LED fault
Soft start	Measured SS	High	Normal operation
Internal bias rail	Measured INTVCC	Around 2.3-2.4	A bit low
PWM input	Drove PWM from ESP32 and felt it high at 3.3V	No change	PWM_DIM is not a problem
PWMTG	Measure PWMTG gate	Gate is at 12V	This means its OFF
Schematic values	Checked RT, SS, FB, CTRL, EN/OVLO, INTVCC cap, BST cap, VOUT caps, PMOS path	Values match datasheet recommendations or equations	Nothing wrong with design

Figure 17: Table of LED Driver tests and results

When testing for functionality of the LT3922, the first symptom of it not working was that the boost converter was not switching when powered on. On the first IC, all pins were probed and checked against the datasheet, including adjusting the EN pin voltage divider after finding the enable voltage was near the off threshold, testing with and without load, and testing with PWM dimming held high and at 1 kHz. None of these changes had any effect, and all other pins appeared normal aside from the feedback pin reading low, which is expected when the converter is not switching. The set of measurements done on the LED driver to isolate the root cause of the failure can be seen in Figure 17 below:

The first IC was replaced with a second, and probing revealed that the FAULT pin was being driven low. Based on the datasheet, this indicated an open-circuit condition. When doing a diode-test of the PMOS, we discovered that the PMOS was broken, since it was short circuiting from drain to source. Further inspection of the PMOS in the schematic showed that it had been placed with incorrect orientation in the footprint. Once corrected, the converter was audibly switching, but the IC failed shortly after. Probing showed that the internal regulator (INTVCC) was no longer outputting any voltage, preventing the IC from completing its soft-start sequence. With both ICs damaged and insufficient time to order replacements, the design was switched to the backup solution.

The replacement driver, the LDH-25-250W [10], includes short-circuit protection (SCP) and overvoltage protection (OVP) in place of the LT3922's feature set, and stops operating when the input voltage drops

below approximately 8–9 V, which we found by connecting the LED driver to an external power supply at 12V, and then reducing that voltage to lower.

In terms of our requirement in which we had to show OVLO and UVLO, this requirement was not exactly met mainly because the change of IC made so that these conditions changed. For the LT3922, the OVLO and UVLO settings are custom, whereas in the LDH-25-250W, this is set at a specific voltage [10]. When talking to the professor and our TAs, they understood that this requirement had to be changed or discarded since it did not apply to our new solution.

For the requirement in which we had to match the forward voltage desired, we tested it by measuring the output voltage going to the LEDs using a multimeter. The voltage measured was around 14.51V, which originally seemed kind of low compared to the calculated 17.41V. However, we know this voltage is still correct, because at 40% of the output current, we are running the LEDs at 100mA, which is less than the DC nominal current of the LEDs. When LEDs are run at less than the nominal current required, this does not mean that they will not turn on, it instead means that the forward voltage going through them will be lower. This explains why the voltage seen is lowered, if we calculate the forward voltage using the minimum, we get the following:

- Deep Red: 1.8 V, 200 – 300 mA
- Royal Blue: 2.9 V, 200 – 300 mA

Adding the forward voltages gives a total string voltage shown below:

- For the blue LEDs,  $2 \times 2.9 = 5.8V$
- For the red LEDs,  $5 \times 1.8 = 9V$
- In total,  $5.8V + 9V = 14.8V$

Based on this, we know that the forward voltage is around that range when at a minimal condition. Thus, this requirement is still met, especially since the lights were still decently bright.

To simulate sun and allow the plants to grow, and to meet our requirement of having the lights dim, we decided to write code to create a cycle of sunrise, daytime, sunset, and nighttime for our grow LEDs. Since the microcontroller sends out a PWM signal, we can write only a portion of the duty cycle (from 0 to 255, which is full brightness) to the LED driver and thus to the LEDs, to control how bright they are. The code is set up in a continuous loop so that for the first 15 seconds of time, the LEDs remain off (this is nighttime), then for the next 45 seconds there is a gradual brightening to sunrise using a linear function relating the portion of the duty cycle to the amount of time that had elapsed in the sunrise interval. Then the lights would remain on for 15 seconds (a duty cycle of 255) and finally the lights would fade out as sunset (using a similar function as for sunrise but with opposite slope). Finally, the LEDs remain off for 45 seconds to represent nighttime. We were able to verify the LEDs' functionality by visually confirming that the lights were dimming in the sunset and sunrise cycles. We also had print statements that confirmed when sunrise and sunset began and what percent of the duty cycle was

being used in that instance, which showed us that brightening and dimming were occurring as the full duty cycle had not been reached yet.

### **3.4 Water Pump Subsystem**

For the water pump, we wanted to be able to pump at least three times the volume of the tank every hour to allow for the fish to receive oxygen and to circulate the nutrients from the tank to the plants. To do this, we knew that the microcontroller has a PWM cycle output that controls the speed of the water pump through the motor controller. We want the pump to speed up its rate of circulation if the flow rate is below three times the volume of the tank and for the fault LED to be lit up in that case, and we want to maintain the flow rate to be above the minimum flow rate. Since our tank is 0.8 gallons (roughly 3.028 liters), the minimum required flow rate is about 9.084 L/hr. To do this, within our overall control loop we added some code to control the water pump's speed based on the current flow meter reading. In the first 30 seconds of the aquaponics system being powered, the water pump speed would be forced to 0 L/hr to demonstrate fault LEDs turning on. The main control for the water pump is within one if statement, where if we are not within the first 30 seconds of powering the system, and the flow rate is below our minimum requirement, then the fault LEDs should turn on and the pump should receive a duty cycle equal to 10% of the maximum duty cycle, bringing the flow rate up above the desired minimum. We verified this to be a good signal to send as we first verified the accuracy of our flow meter, and then we used the flow meter readings to make sure that the 10% duty cycle sent resulted in a flow rate above 9.084 L/hr. If the flow rate is above the required minimum, the fault LED is turned off. An example of the reading from the flow meter can be seen in Figure 16.

### **3.5 Power Subsystem**

To verify the power subsystem, we used a multimeter to take voltage measurements across various electrical components on our PCBs. We have compiled them onto the table below. There is a small offset from the theoretical value of the voltages compared to their actual values, but that is due to natural variation in theory and applications. Everything matches for correct functionality.

Power Supply To:	From:	Minimum:	Maximum:	Actual:	Units:
Servo Motor for Fish Feeder	12-5V Buck Converter	4.8	6	5.1	V
Water Pump	12V Power Input	6	18	12.41	V
LED Driver	12V Power Input	2.8V	26V	12.41	V
LED String	LED Driver Output	26.65V	N/A	14.51	V
Fault LEDs	MCU	1.8	3.6	3.34	V
Water Flow Sensor	12-5V Buck Converter	5	24	5.1	V
Water pH Sensor	12-5V Buck Converter	5	5	5.1	V
Temperature Sensor	12-5V Buck Converter	3	5.5	3.34	V
MCU	5-3.3V LDO	3.0	3.6	3.34	V

Figure 18: Table of expected voltage readings compared to actual voltage readings using a multimeter

## 4. Costs

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 originally estimated that this project would take about 10-15 hours per week for the 7 weeks remaining in this class, in the end we ended up working for 139 hours on this project, 49 hours more than our original estimates. This comes to a calculation of \$39 per hour \* 2.5 \* 139 hours = \$13,522 per member. This time 3 comes out to a total of \$40,657.5.

Our additional parts costs are in Appendix B on the parts table. From there, we concluded that we spent close to our original estimate, with the main change being in the chosen fish tank and the new LED driver module costs. The total parts came to \$138.96.

In total, the machine shop worked on this project for about 3 days and an estimated 12 hours, which equates to \$39 \* 12 hours = \$486. This, all together with our working hours, the machine shop, and parts, comes to around \$41,264.46.

## 5. Schedule

From our initial schedule we made a few changes when working on our project. The main changes just had to do with giving ourselves more time to debug our sensor code and LED driver as we decided to just go with our third round PCB design instead of our fourth round and thus, we had less soldering to do. The detailed weekly schedule can be found in Appendix D.

## 6. Conclusions

### 6.1 Accomplishments & Future Work

By the end of this project, we successfully accomplished almost all our requirements and verifications through the process of designing, testing and integrating each subsystem. We were able to run the water quality, water pump, fish feeder, lighting and power subsystems together consistently for at least 24 hours. All sensors have us usable readings that were within our required tolerances, the fault LEDs turned on appropriately during abnormal conditions, the fish feeder rotated every 10 seconds for our test setup, the pump maintained constant water circulation, and the lighting followed the night-time and day-light schedule properly.

The two main parts that did not work as expected were the LED driver and the USB-to-UART programmer IC. For the LED driver, we were able to narrow down the issue to the PMOS symbol/footprint mismatch in the dimming path. After correcting this, the converter finally began switching but the IC failed shortly after. Since both ICs were damaged and there was not enough time to order replacements, we switched to our backup LED driver, which worked well enough to complete the lighting requirement. In the future, we would prefer to have the LED driver fully integrated on the PCB.

The USB-to-UART programmer also did not work as intended. It was an extra feature, rather than a core project requirement, so it did not affect our requirements and verification. After debugging, I believe the issue may be related to the LEDs added to the TX/RX traces, which might interfere with the signal. Since the ESP32 could still be programmed externally, in the future it might be best to discard this system overall. Making programming and debugging easier is not a requirement for production, since users should have no need to program the board easily.

For future work, we hope to improve upon this project by adding an LCD display to the front of the fish tank and an app component to allow the user to see when the fish tank has abnormal readings faster and easily compared to the fault LEDs. We would also like to add some additional oxygenation to the tank more than falling water breaking the fish tank surface to allow for more varieties of fish to be housed in tank.

### 6.2 Societal Impact & Ethic

Over 28% of the global population live in a state of medium or high-risk food insecurity. Our project, a desk top automated aquaponics system addresses the needs for many who lack access either to fresh produce or may not have the space/climate to grow plants. With monitoring and alerting the user to abnormal pH, temperature, and flow, it is accessible and has limited human monitoring required. The

monitoring combined with the horticulture LEDs which are also programmable, this product can be adjusted for different plants and climates and is easy to use in general.

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. This is in line with the IEEE standard, “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” [9]. Due to potential harm from unsealed electronics, we opted not to have a fish in the tank and to keep sustainable development practices without exposing wildlife to any potential harm. With this, we also built a protective casing around the controls circuit board to prevent any water from spilling onto the board.

## 7. 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/](http://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](http://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](http://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](http://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](http://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](http://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](http://www.digikey.com/en/products/detail/espressif-systems/ESP32-S3/15822445). Accessed 13 Feb. 2026.
- [9] IEEE. (2020). *IEEE Code of Ethics | IEEE*. [www.ieee.org](http://www.ieee.org).  
<https://www.ieee.org/about/corporate/governance/p7-8>. Accessed 12 Feb 2026.
- [10] MEAN WELL USA Inc. "LDH-25-500W." *DigiKey*, <https://www.digikey.com/short/rvz82w03>. Accessed 13 Feb. 2026.
- [11] 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?srsId=AfmBOooB2E2\\_wTROrZeA-FXmLRjT-pVKQeHeAogrwlPC86nX5K8ByoA05ec](http://mullettools.com/products/4-inch-thin-wall-furniture-grade-pvc-pipe-light-gray?srsId=AfmBOooB2E2_wTROrZeA-FXmLRjT-pVKQeHeAogrwlPC86nX5K8ByoA05ec). Accessed 27 Feb. 2026.
- [12] Robocraze. "pH Sensor Working Principle, Types and Applications." <https://robocraze.com/blogs/post/all-about-ph->



## Appendix A: Requirements & Verification Table

### A.1. Water Quality R&V Table

Requirement	Verification	Pass Criteria
<p>The pH sensor must accurately measure the pH of the water in the fish tank within <math>\pm 0.5</math> 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.</p>	<ul style="list-style-type: none"> <li>• 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.</li> <li>• 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.</li> </ul>	<ul style="list-style-type: none"> <li>• pH strip vs. pH print on the Serial Monitor</li> </ul>
<p>Test MCU is working</p>	<ul style="list-style-type: none"> <li>• Attempt to program it using both UART pins and BOOT/Reset buttons. Wait for the serial terminal message to say that code has been uploaded properly.</li> <li>• Next, test using USB-UART connector, and check for message saying code has been uploaded through the serial terminal</li> </ul>	<ul style="list-style-type: none"> <li>• Show the Serial Monitor uploading code prints &amp; hit reset to run code</li> </ul>
<p>The fault LEDs will indicate when an abnormal pH is sensed by the pH sensor</p>	<ul style="list-style-type: none"> <li>• 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.</li> <li>• 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</li> </ul>	<ul style="list-style-type: none"> <li>• Alkaline water should show an abnormal pH and the fault LEDs light</li> </ul>

	be lit.	
<p>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 from. If the temperature is wrong, MCU must send out a fault signal to the LED indicator.</p>	<ul style="list-style-type: none"> <li>• 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 for at least 60 seconds of error free before verification.</li> <li>• 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).</li> <li>• 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</li> </ul>	<ul style="list-style-type: none"> <li>• Check with thermometer and compare against the Serial Monitor</li> </ul>
<p>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 within the correct range, MCU must be able to communicate with the water pump to change flow rate or send out a fault signal to the LED if there is a fault condition.</p>	<ul style="list-style-type: none"> <li>• 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.</li> <li>• 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 [13]. 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.</li> <li>• We will test that the fault is working correctly by slightly</li> </ul>	<ul style="list-style-type: none"> <li>• Fill 1 cup of water and time it, to calculate the flow rate in liters per hour.</li> </ul>

	blocking the tube and verifying that the LED turns on if no water is allowed to come through the tube.	
--	--	--

## A.2 Water Pump R&V Table

Requirement	Verification	Pass Criteria
The water pump will circulate water which will result in the tank volume being replaced at least 3-5 times per hour.	<ul style="list-style-type: none"> <li>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 of error free before verification.</li> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>Show the flow rate from the Serial Monitor (previous requirement in flow meter should show that flow is accurate)</li> </ul>
The pump should be operational for at least 24 hours without issue.	<ul style="list-style-type: none"> <li>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.</li> <li>Set up the water pump and ensure there is water flow between the fish tank and the plants on top.</li> </ul>	<ul style="list-style-type: none"> <li>Leave pump running for the full demo period &amp; we have tested pump running for several hours while debugging</li> </ul>

	<ul style="list-style-type: none"> <li>Then, leave the pump running for the system for a smaller interval test and record any abnormalities in the water flow by using the water flow sensor and making sure the rate is always above 0.</li> </ul>	
The water pump will change speed depending on the duty cycle of the PWM going into the MOSFET.	<ul style="list-style-type: none"> <li>Check the PWM signal going into the water pump with an oscilloscope, then check if speed has changed through visualization.</li> </ul>	<ul style="list-style-type: none"> <li>Start with everything off, and then the water pump turns on.</li> </ul>

### A.3 Lighting R&V Table

Requirement	Verification	Pass Criteria
The LED driver should supply at least the forward voltage in series to the blue and red growth LEDs as well as not overheat after continuous time intervals of being lit.	<ul style="list-style-type: none"> <li>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.</li> <li>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.</li> <li>Connect the LEDs to an oscilloscope for some time to observe any abnormalities such as voltage spikes.</li> </ul>	<ul style="list-style-type: none"> <li>Demonstrate with a multimeter that the constant voltage going to the LED driver is around the forward voltage required</li> </ul>

<p>The lighting subsystem should dim the lights whoever we want according to the PWM input to the LED driver.</p>	<ul style="list-style-type: none"> <li>• 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.</li> <li>• 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.</li> <li>• When the duty cycle is changed, we should see a change in brightness to the LEDs.</li> </ul>	<ul style="list-style-type: none"> <li>• Oscilloscope to show the PWM from the LED Driver</li> <li>• Lighting starts in the off section</li> </ul>
<p>LED Driver should stop operation under UVLO and OVLO conditions. (This requirement no longer applies since LED Driver got changed)</p>	<ul style="list-style-type: none"> <li>• 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 an external PWM signal is coming in. (This requirement no longer applies since LED Driver got changed)</li> </ul>	<ul style="list-style-type: none"> <li>• Changed to a LDH-25-250W, Instead, our new LED Driver contains OVP and SCP.</li> </ul>

#### A.4 Power R&V Table

Requirement	Verification	Pass Criteria
<p>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.</p>	<ul style="list-style-type: none"> <li>• 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.</li> <li>• Use a multimeter to check the output of the buck converter, and that the 12V has been stepped down to 5V.</li> <li>• 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.</li> </ul>	<ul style="list-style-type: none"> <li>• Check with multimeter</li> </ul>
<p>The LDO should step down the 5V to at least 3.3V to the microcontroller.</p>	<ul style="list-style-type: none"> <li>• 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 of error free before verification.</li> <li>• Using a multimeter, make sure the voltage is 3.3V at the ESP32 power pin.</li> </ul>	<ul style="list-style-type: none"> <li>• Check with multimeter</li> </ul>
<p>The circulation pump should receive at least 6V with a maximum of 18V directly from the power unit.</p>	<ul style="list-style-type: none"> <li>• 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 of error free before verification.</li> <li>• Using a multimeter ensures the wall power source is 12V and that the circulation pump is 12V.</li> </ul>	<ul style="list-style-type: none"> <li>• Check with multimeter</li> </ul>
<p>Ensure all components receive the appropriate input voltage according to the table above.</p>	<ul style="list-style-type: none"> <li>• 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</li> </ul>	<ul style="list-style-type: none"> <li>• With all the above verifications, this is satisfied</li> </ul>

## A.5 Fish Feeder R&V Table

Requirement	Verification	Pass Criteria
<p>The fish feeder subsystem must receive a PWM signal from the microcontroller to turn on the servo motor.</p>	<ul style="list-style-type: none"> <li>• 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.</li> <li>• 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</li> </ul>	<p>Since the fish feeder is moving, it is implied that the PWM signal is working</p>
<p>The servo motor should rotate approximately 180 degrees (+/- 20 degrees) based on the PWM signal sent from the microcontroller.</p>	<ul style="list-style-type: none"> <li>• 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.</li> <li>• 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.</li> </ul>	<p>Visually can see that the food dispenser holes rotate from the top to the bottom when servo is on</p>
<p>The fish feeder should dispense</p>	<ul style="list-style-type: none"> <li>• Ensure the board is in a</li> </ul>	<p>Servo motor rotates once every 30</p>

<p>food into the tank at least once every 24 hours.</p>	<p>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.</p> <ul style="list-style-type: none"> <li>• 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.</li> </ul>	<p>seconds</p>
---	---	----------------

## Appendix B: Costs

### B.1 Parts Table

Description	Manufacturer	Quantity	Extended Price	Link
ESP32-WROOM-32E-N8R2 (MCU)	Espressif Systems	1	5.38	<a href="#">Link</a>
LED Lighting Color Horticulture Red 731nm (720nm ~ 745nm) 1113 (2835 Metric)	Broadcom Limited	3	4.44	<a href="#">Link</a>
LED HORTICULTURE BLU 451NM 2835	Broadcom Limited	5	6	<a href="#">Link</a>
Liquid Flow Sensor 1 ~ 30 LPM Plastic	Seeed Technology Co., Ltd	1	9.50	<a href="#">Link</a>
ANALOG PH METER KIT	DFRobot	1	29.50	<a href="#">Link</a>
DC MINI IMMERSIBLE WATER PUMP 6V	DFRobot	1	9.20	<a href="#">Link</a>
Waterproof 1-Wire DS18B20 Digital temperature sensor	Adafruit	1	9.95	<a href="#">Link</a>
SERVOMOTOR RC 4.8V	DFRobot	1	3.62	<a href="#">Link</a>
TERM BLK 3POS SIDE ENTRY 5MM PCB	Würth Elektronik	1	0.61	<a href="#">Link</a>
12V 6A Power Supply Adapter	COOLM	1	14.59	<a href="#">Link</a>
TVS DIODE 12VWM 19.9VC SOD123F	Littelfuse Inc.	2	0.46	<a href="#">Link</a>
Zener Diode 12 V 500 mW ±5% Surface Mount SOD-123	Diodes Incorporated	1	0.10	<a href="#">Link</a>
CAP CER 1UF 35V X7R 0805 *	Taiyo Yuden	10	1.6	<a href="#">Link</a>
CAP CER 0.1UF 35V X7R 0603 *	Taiyo Yuden	15	1.65	<a href="#">Link</a>

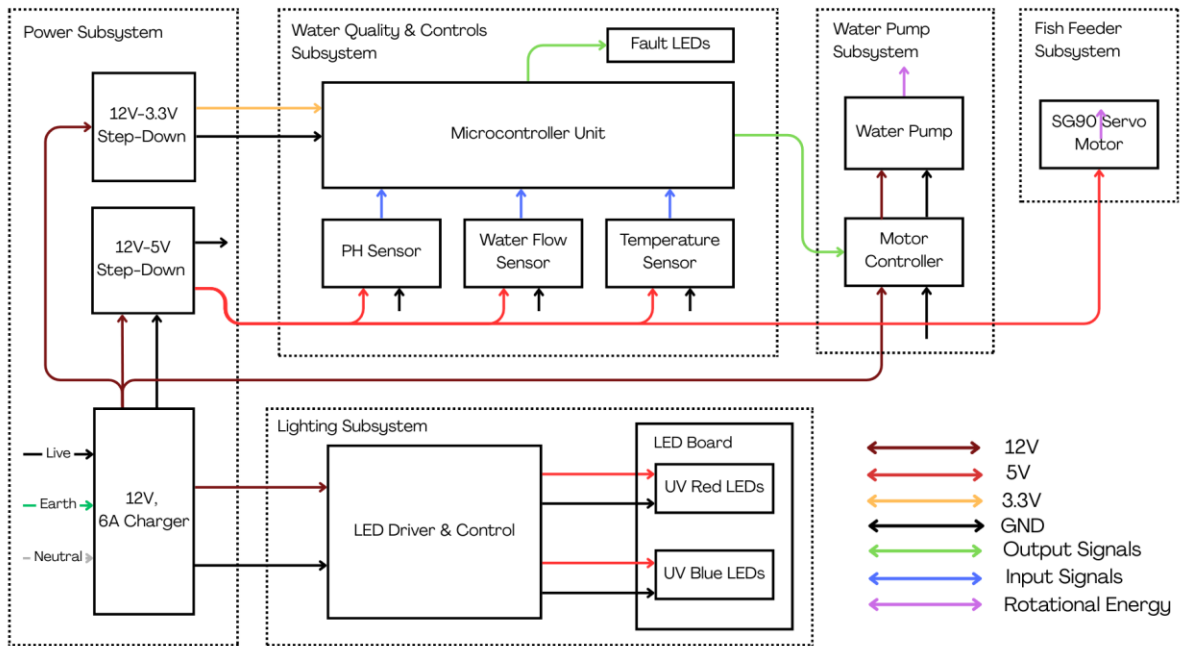
CAP ALUM 100UF 20% 35V SMD	KEMET	1	0.63	<a href="#">Link</a>
CAP ALUM 47UF 20% 50V RADIAL TH	Rubycon	2	0.56	<a href="#">Link</a>
CAP CER 10UF 25V X5R 1206 *	Taiyo Yuden	4	0.32	<a href="#">Link</a>
CAP ALUM 220UF 20% 35V RADIAL TH	Rubycon	2	0.86	<a href="#">Link</a>
CAP ALUM 470UF 20% 35V RADIAL TH	Rubycon	1	0.6	<a href="#">Link</a>
CAP CER 0.033UF 50V X7R 0805	KYOCERA AVX	1	0.21	<a href="#">Link</a>
CAP CER 0.47UF 50V X5R 0402	Murata Electronics	2	0.2	<a href="#">Link</a>
CAP CER 2.2UF 25V X5R 0603	Taiyo Yuden	1	0.08	<a href="#">Link</a>
CAP CER 10000PF 50V X7R 0603	KEMET	1	0.08	<a href="#">Link</a>
1000 pF ±10% 50V Ceramic Capacitor X7R 0402 (1005 Metric) *	Murata Electronics	1	0.08	<a href="#">Link</a>
CAP CER 22UF 25V X5R 1206	Taiyo Yuden	3	0.27	<a href="#">Link</a>
FIXED IND 4.7UH 3A 37.6 MOHM SMD	Bourns Inc.	1	0.42	<a href="#">Link</a>
FIXED IND 3.3UH 3A 30 MOHM SMD	Taiyo Yuden	1	0.23	<a href="#">Link</a>
CONN PWR JACK 2.5X5.5MM SOLDER	Würth Elektronik	1	0.94	<a href="#">Link</a>
DIODE SCHOTTKY 40V 2A DO214AA	Taiwan Semiconductor Corporation	1	0.38	<a href="#">Link</a>
MOSFET N- CHANNEL 30V 3A TSMT3	Rohm Semiconductor	1	0.72	<a href="#">Link</a>
FUSE BRDMT 10A 125VAC/125DC 2410	Bourns Inc.	1	1.18	<a href="#">Link</a>
RES 30K OHM 1% 1/8W 0805	YAGEO	2	0.20	<a href="#">Link</a>

RES 220K OHM 1% 1/4W 0805	YAGEO	1	0.11	<a href="#">Link</a>
RES SMD 20K OHM 0.1% 1/8W 0805	YAGEO	1	0.10	<a href="#">Link</a>
RES SMD 2M OHM 0.1% 1/10W 0805	TE Connectivity Passive Product	1	0.18	<a href="#">Link</a>
RES 10K OHM 1% 1/5W 0603 *	YAGEO	6	0.66	<a href="#">Link</a>
RES SMD 20K OHM 1% 1/10W 0603	YAGEO	1	0.11	<a href="#">Link</a>
RES 1K OHM 1% 1/10W 0603 *	YAGEO	3	0.30	<a href="#">Link</a>
RES 4.7K OHM 1% 1/10W 0603 *	YAGEO	1	0.10	<a href="#">Link</a>
RES 220 OHM 5% 1/10W 0603	YAGEO	1	0.10	<a href="#">Link</a>
RES 100K OHM 1% 1/10W 0603	YAGEO	2	0.22	<a href="#">Link</a>
RES 100 OHM 1% 1/10W 0603	YAGEO	1	0.10	<a href="#">Link</a>
RES 270 OHM 1% 1/10W 0603	YAGEO	3	0.30	<a href="#">Link</a>
RES 1.1M OHM 1% 1/4W 1206	YAGEO	1	0.10	<a href="#">Link</a>
RES 59K OHM 1% 1/8W 0805	YAGEO	1	0.10	<a href="#">Link</a>
RES 100K OHM 5% 1/8W 0805 *	YAGEO	1	0.10	<a href="#">Link</a>
RES 43.2K OHM 1% 1/10W 0603	YAGEO	1	0.10	<a href="#">Link</a>
RES SMD 1M OHM 5% 1/8W 0603	YAGEO	1	0.28	<a href="#">Link</a>
SWITCH TACTILE SPST-NO 0.05A 12V	Alps Alpine	2	0.62	<a href="#">Link</a>
RES 0 OHM JUMPER 1/10W 0603	YAGEO	3	0.30	<a href="#">Link</a>
RES 0.5 OHM 1% 1/2W 1206	Stackpole Electronics Inc	1	0.22	<a href="#">Link</a>
RES 22.1K OHM 1% 1/10W 0603	YAGEO	2	0.10	<a href="#">Link</a>
RES 47.5K OHM	Stackpole	1	0.10	<a href="#">Link</a>

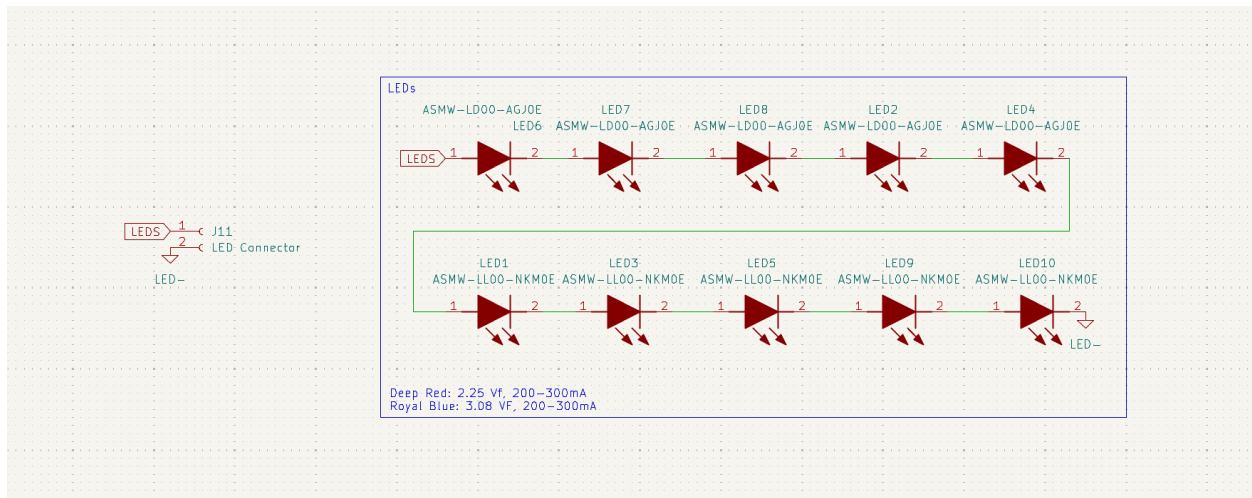
<b>1% 1/10W 0603</b>	<b>Electronics Inc</b>			
<b>RES 10K OHM 1% 1/10W 0603</b>	<b>YAGEO</b>	<b>3</b>	<b>0.30</b>	<a href="#">Link</a>
<b>RES 5.1K OHM 1% 1/10W 0603</b>	<b>YAGEO</b>	<b>1</b>	<b>0.10</b>	<a href="#">Link</a>
<b>0.8-Gallon Cube Small Fish Tank, Shatterproof Plastic Fish Tank with LED Lighting, Temporary Fish Bowl for Fry, Isolation, Education &amp; Decoration</b>	<b>SW PRODUCTS</b>	<b>1</b>	<b>18.99</b>	<a href="#">Link</a>
<b>DC-DC STEP-UP CONSTANT CURRENT L</b>	<b>MEAN WELL USA Inc.</b>	<b>1</b>	<b>10.71</b>	<a href="#">Link</a>

# Appendix C: Design

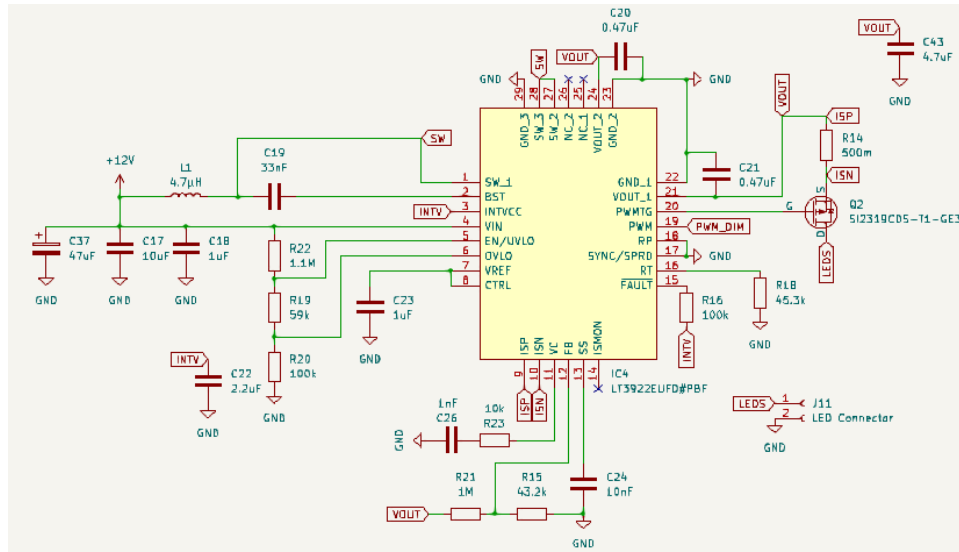
## C.1 Original Design Documentation Block Diagram



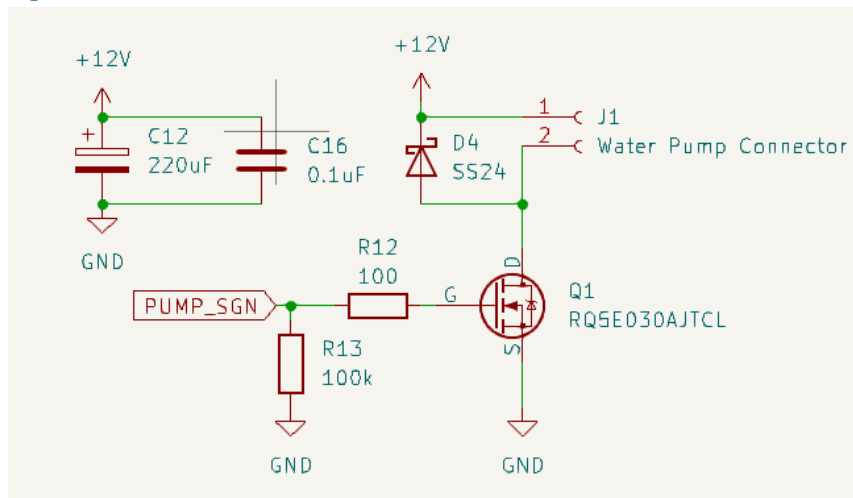
## C.2 LED Board Circuit Schematic



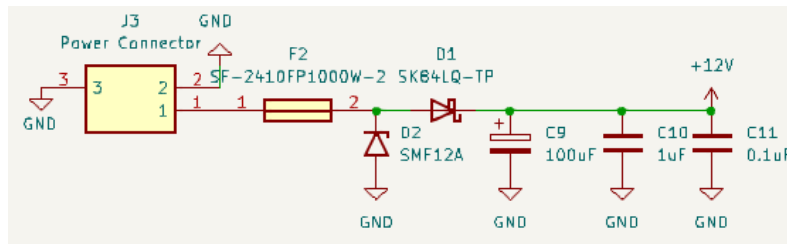
### C.3 LED Driver Circuit Schematic



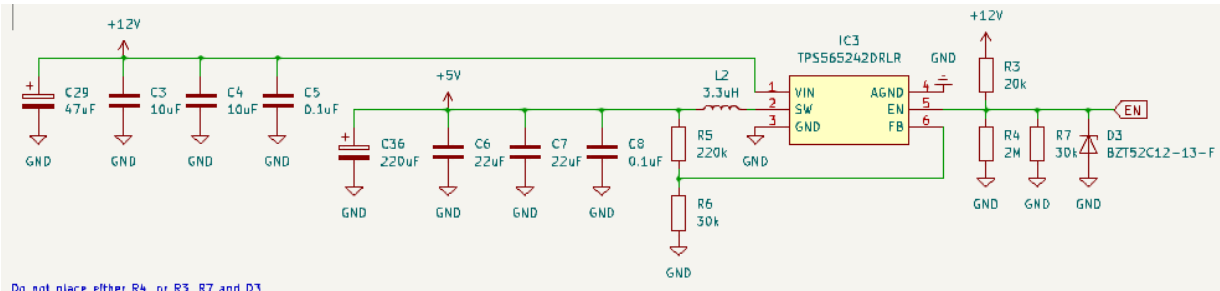
### C.4 Water Pump Circuit Schematic



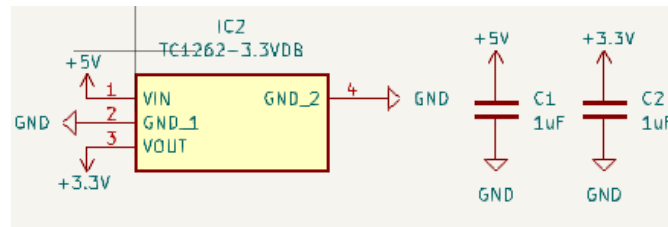
### C.5 Power Input Circuit Schematic



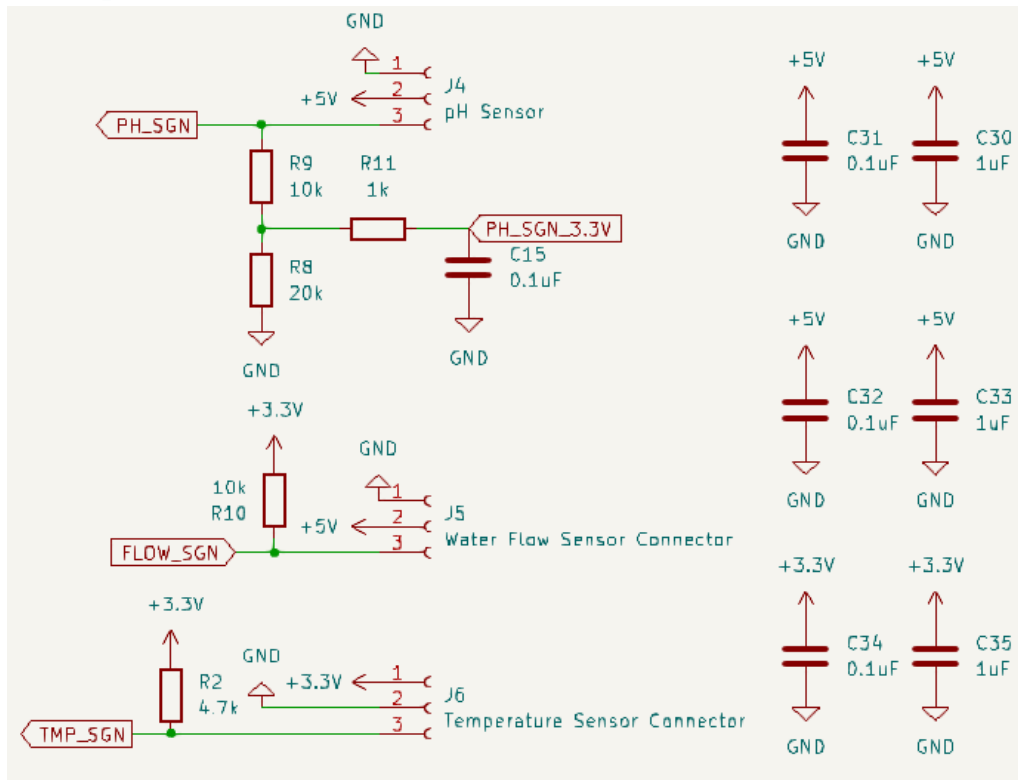
## C.6 Buck Converter Circuit Schematic



## C.7 LDO Circuit Schematic

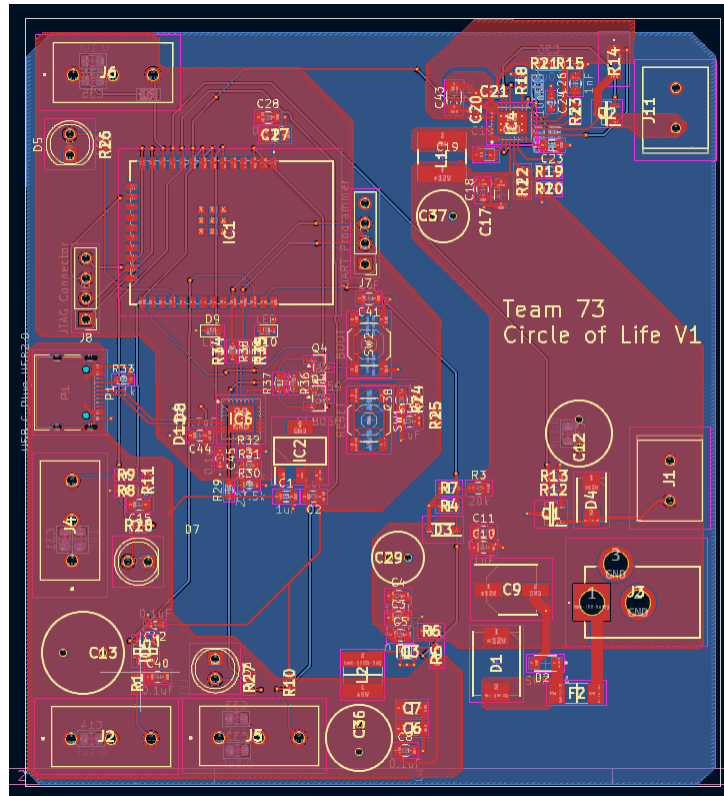


## C.8 Sensor Inputs Circuit Schematic

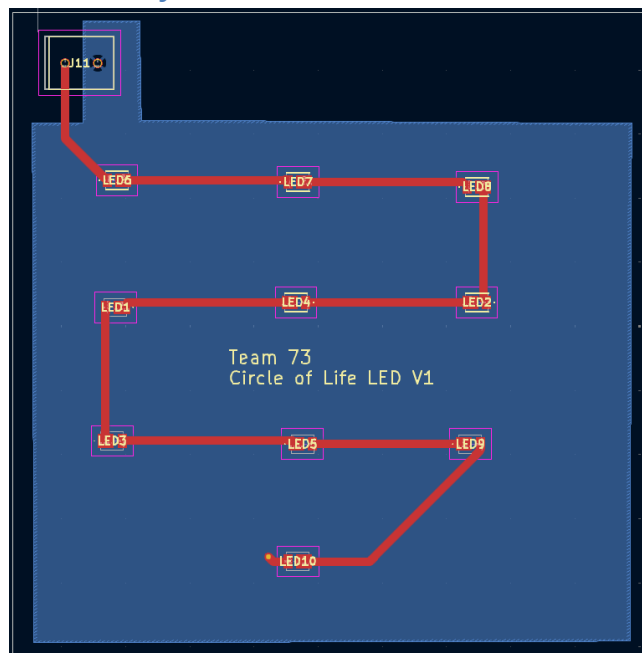




### C.12 First Round Main Power & Control PCB Layout



### C.13 First Round LED PCB Layout



## Appendix D: Weekly Schedule Table

Week	Tasks	People
2/23 – 2/27	Finish PCB First Round Design	Estela
	Order First Round PCB	Anjali
	Order Parts	Anjali & Aishwarya
	Speak with Machine Shop for physical design	Everyone
	Finish Design Document	Everyone
3/2 – 3/6	Prepare Design Review Presentation	Everyone
	Start Prototyping Board (Sensors & Fish Feeder)	Anjali & Aishwarya
	Update PCB Design	Estela
	Order 2 <sup>nd</sup> Round PCB	Anjali
3/9 – 3/13	Breadboard Demo	Everyone
	Third Round PCB Order	Anjali
	Update PCB	Estela
	Start MCU Code	Aishwarya
	Final updates for physical design with machine shop	Everyone
	Breadboard Sensors & Lighting	Anjali & Aishwarya
3/16 – 3/20	SPRING BREAK	
3/23 – 3/27	4 <sup>th</sup> Round PCB Order	Estela
	Continue debugging 3 <sup>rd</sup> round PCB	Estela
	Connect Power subsystem to Sensors & Lighting subsystems	Everyone
	Work on power & water pump subsystems	Anjali & Aishwarya
3/30 – 4/3	Individual Progress Report	Everyone
	Start putting everything together for progress demo	Everyone
4/6 – 4/10	Progress Demo	Everyone
	Work on LED Driver hardware issue	Estela
	Continue working on code & debugging all subsystems	Aishwarya & Anjali
4/13 – 4/17	Continue working on code & debugging all subsystems	Everyone

	<b>Full assembly &amp; Final bugs</b>	<b>Everyone</b>
	<b>Start preparing for Mock Demo and Mock Presentation</b>	<b>Everyone</b>
<b>4/20 – 4/24</b>	<b>Mock Demo</b>	<b>Everyone</b>
	<b>Mock Presentation</b>	<b>Everyone</b>
	<b>Continue debugging sensors</b>	<b>Anjali &amp; Aishwarya</b>
	<b>Start working on Final Presentation and Final Report</b>	<b>Everyone</b>
<b>4/27 – 5/1</b>	<b>Prepare final presentation slides and final demo practice</b>	<b>Everyone</b>
	<b>Final Demo</b>	<b>Everyone</b>
	<b>Final Presentation</b>	<b>Everyone</b>
<b>5/4 – 5/8</b>	<b>Final paper editing and submission</b>	<b>Everyone</b>
	<b>Lab Notebook submission and lab cleaning</b>	<b>Everyone</b>