PoolWatch

Optical Chlorine and Phosphate Concentration Analyzer with Particle Imaging for Classification

Group 4

Authors:

Ning Dim	Dylan Hughes	Austin Naugle	Jason Ser
CPE	CPE	PSE	PSE

Reviewer Committee:

Dr. Thames	PHY
Dr. Pang *(Only SD1)*	PSE
Dr. Salamin	PSE
Dr. Xin Xin	ECE

Mentors:

Dr. Chung Yong Chan

Dr. Aravinda Kar

UNIVERSITY OF CENTRAL FLORIDA

Table of Contents

List of Tables

List of Figures

List of Equations

Chapter 1 Executive Summary

In ancient Rome and Greece, public bath houses were early centers of exercise, hygiene, and relaxation–precursors to the modern swimming pool. The term “swimming pool” itself gained traction in the nineteenth century. Since then, advances in science and technology have allowed homeowners across the United States to own a commercial-grade swimming pool in their own backyards.

Despite our human innovation, maintaining pool health has become an expensive and daunting task. Monitoring temperature, regulating chlorine levels, and preventing algae growth still need manual intervention. If someone were to neglect these factors, then this will lead to expensive repairs and unsafe swimming conditions.

Poolwatch aims to eliminate these issues by being cost-effective, environmentally safe, and user friendly. Our system uses advanced optical sensors to provide real-time automated measurements of Chlorine and Phosphate levels extracted from a test cuvette filled with a mixed reagent and pool water, while all data acquisitions such as chemical concentrations, temperature, and calibration metrics–are logged and stored on a secure web server. The user receives timely notifications and has intuitive access to all recorded information.

By automating pool monitoring, Poolwatch reduces maintenance costs, follows chemical safety, and gives users serenity–all while contributing to more sustainable pool management.

Chapter 2 Project Description
2.1 Motivation and Background

Beating the hot sun of the south with a relaxing pool day is a common sight amongst Floridians, however maintaining a pool takes a great deal of effort. Private pool owners tend to be too preoccupied to be constantly watching the condition of their pool, which often leads to a less than ideal swimming experience. If pool owners are not aware of the chemical composition of their pools, they can end up with algae infestations or other microorganisms making swimming dangerous. Since there exist many detriments to not properly monitoring pools, we have decided to design a pool monitor device for senior design.

The University of Central Florida offers its students the opportunity to complete a senior design project. This is designed to give the students experience and skills with working on an engineering project before going into the professional workplace. We hope to learn from this class project planning, researching, designing, implementation, and manufacturing. To achieve this we decided on a project that would have aspects of all of our areas of studies involved. We also wanted to involve technology we have not yet studied to provide ourselves with opportunities to learn and grow.

Our team was made out of 2 CpE majors and 2 CREOL majors, which means we were required to choose a project that involved at least 2 optical systems. We brainstormed several ideas before we landed on the pool monitoring device, but found they could not achieve all of our requirements. Collecting statistics of a pool requires several different optic devices, which allows us to fill the requirement that the CREOL students have. Since this device would need some form of MCU and power supply at the least, we are also able to meet the CpE requirements. With all of our requirements considered we found that the pool monitoring system would be the best choice for our senior design project.

2.2 Existing Products

There is no one system on the market currently that can provide all the functionalities of our proposed system, thus we have compiled a list of the four closest comparable technologies to our proposed Pool Monitoring System, each of which shares certain features with our design. Our system represents a hybrid approach that integrates the key strengths of these individual solutions to encompass a design that favors portability, accuracy of readings, long battery life, and ease of use.

[1] The first is Pentair's IntelliChem Commercial Controller System which automatically monitors pool pH , ORP , and temperature and based on those readings, administers the correct dosage of chemicals to bring the water chemistry back into balance. ORP stands for oxidation-reduction potential and is a measure of the cleanliness of the water and is an indirect way to measure the chlorine levels; the system is unable to measure phosphate directly as it is not a test included in the system. The monitoring system is also password-protected and includes an app for remote system monitoring and control. The price tag is around $4,132.42 and the system works best with the Pentair pool system. This setup is the most expensive and complicated setup among the four systems, as a professional installer would be required to attach the device to existing pool infrastructure in order to operate correctly.

[2] The second system is the Hanna Instruments Swimming Pool and Spa Controller with Analog Outputs which can monitor pH and chlorine at regular intervals that is customizable and sends the information to the designated app to view alarms, history logs, trends, and more. The system costs $2,059.99 and must have the system's probes connected to either in-line after the pool filter or mounted in the flow cell close to the controller to sample the pool water directly, which would require a professional installer to correctly attach to existing infrastructure. The price, however, is about half that of the previous system, and it is able to directly measure the chlorine levels at selected intervals remotely.

[3] The third system is the WaterGuru Smart Pool Monitoring, which costs $349 and is a pool monitor that fits inside a pool skimmer of width 8.25" and 9.4". It monitors water temperature every 30 minutes and also measures pH and chlorine, with all the info being sent to an app, where pool care advice is also doled out based on information received about the pool status. This system is a lightweight design compared to the previous two, consisting of a pool skimmer cover connected to a monitor. No extra tools are needed for installation, as the monitor just gets placed into the pool skimmer and requires only 4 double AA batteries for operation. One downside besides the rigid pool skimmer width requirement is that the system is making use of the skimmer basket so if any large debris gets into the skimmer, it will run into the monitoring system and potentially cause damage. Another potential issue is that chemistry pads are used to test pool chemistry, so the data readings are not as accurate, resulting in outputs that are more ballpark readings, which can lead to inaccurate dosing.

[4] The last system is the Sutro Water Monitoring System which consists of three parts, the Monitor which is placed in the pool and conducts the tests for pH, chlorine and alkalinity, the Hub that is used to connect to a user's wi-fi as well as receive information from the Monitor, with all the info being sent to the app, and finally the Charger. The system performs chlorine and pH tests three times a day, the alkalinity once a day and water temperature hourly, sending all the information to the app and sending notifications and either weekly or monthly updates as set by the user. This design consists of more parts than previous designs, but the readings are more accurate as real liquid reagents are being used to test pool chemistry in the Monitor, just like in manual testing. The rechargeable battery also allows for easier replacement when power is running low. The main drawback is that more work is required in setup than previous systems to correctly link the Hub and Monitor using wi-fi as connectivity issues can result for those not technically inclined. Another drawback is that since liquid reagents are being used and are packaged in cartridges inside the Monitor, if the cartridge runs out of reagents, the system cannot work or if the cartridge installation results in an error, the system will not connect and will require troubleshooting, complicating the process.

The primary advantage shared by all four systems is their Wi-Fi connectivity, enabling real-time data access through a dedicated app. These apps usually offer features such as trend analysis, historical logs, and remote monitoring. However, the limitations observed across these existing solutions such as lack of portability, complex interfaces, high cost, or inconsistent accuracy, have created a clear market demand for a more comprehensive product. Our proposed system aims to address these gaps by delivering a portable, user-friendly, cost-effective, and accurate solution with robust remote monitoring capabilities.

2.3 Goals and Objectives

2.3.1 Goals

The overall goal of this project is to design a device that gathers data such as the chlorine concentration, phosphoric acid concentration, temperature, and depth of a pool. The drive would then communicate with a website to display the results to the end user. The goals in ascending order are categorized below.

2.3.1.1 Basic Goals

The bare minimum of this project would be to collect the chlorine concentration, phosphoric acid concentration, temperature, and depth of a pool. Then communicate the data to a web server, in order to do that the project would need to achieve the following goals.

- Web server goals
  - Receives data from the device.
  - Display current pool statistics on a browser.
- Device goals
  - Measure pool temperature.
  - Has a power system capable of handling basic operations.
  - Sends data to the web server.
- Optical goals
  - Achieve an accuracy rating 0.5 ppm and R^2 > 0.95 for chlorine and phosphate concentration measurements.
  - Achieve a resolution of 10 μm while imaging particles.
  - Image particulate with a field of view of 1 mm by 1 mm.

2.3.1.2 Advance Goals

To make this project a challenge and a learning experience for us, as well as to make the device more useful, we decided on several advance goals to also achieve for this project, which are as follows.

- Web server advance goals
  - 2 way communication between the device and server.
  - Notification system.
  - Account system.
  - Storing previous pool statistics.
  - A professional website.
  - Displaying meta statistics.
  - Modern web app.
- Device advance goals
  - Monitors meta statistics.
  - Draws water from the pool.
  - A reasonably long lasting power supply.
  - Easily maintainable.
  - Easy troubleshooting.
  - Reasonably sized.
- Optical Advanced Goals
  - Achieve an accuracy rating 0.25 ppm and R^2 > 0.98 for chlorine and phosphate concentration measurement.
  - Reduce the response time by 5% of the PIN photodiode.
  - Achieve a resolution of 10 μm while imaging particulates.
  - Collimate light from the illuminating LED so that it does not diverge more than 1 degree.

2.3.1.3 Stretch Goals

Since we are limited in engineering specialties and time, there are several stretch goals that we would like to achieve for this project, but might not be able to, which are as follows.

- Web server stretch goals
    - Secure communication.
    - Weather monitoring and warning.
    - Phone compatible.
- Device stretch goals
    - Interchangeable web server.
    - Password system
    - Splash resistant.
    - Auto changing chemical test chambers.
- Optical Stretch Goals
    - The device is able to optically measure Chlorine and Phosphate with an accuracy of 0.20 ppm in two separate cuvettes.
    - Achieve a resolution of 1 μm while imaging particulates.
    - Image with a field of view of 10 mm by 10 mm.
    - Capture a 3 dimensional image of the particles using two separate cameras.

2.3.2 Objectives

2.3.2.1 Hardware Objectives

In order to achieve the hardware goals mentioned above the project met the following objectives.

- A power supply capable of lasting for at least a month without the battery needing to be recharged or replaced while running tests at least once a day.
- The MCU communicates with the PIN photodiode on the phosphate detection system, PIN photodiode on the free chlorine detection system, the thermometer, and the Avalanche photodiode.
- The device has a system for pumping water out of the pool and delivering the water into a preloaded N,N-diethyl-p-phenylenediamine (DPD)**[5]** powder and ammonium molybdate with ascorbic acid **[6]** into a cuvette for each respective testing chamber.
- The device has LEDs for every system to visually show that each system is functioning. At the least there is an LED for every regulator, the chlorine sensor, the PIN photodiode, the Avalanche Photodiode, and the MCU.
- To preserve power the aforementioned LEDs remain off unless a switch is flipped on the device.
- Separate major systems such as the regulators or sensors, are easily interchangeable to allow for easy maintenance on the device.
- The device's MCU is capable of completing every test calculation within a small time interval.
- The MCU has WiFi capabilities in order to communicate to the web server.

To achieve the stretch goals mentioned above, we would also need to achieve the following objectives.

- The device would need a keyboard interface for inputting setting
- The device would need a LCD display to show what settings are being edited and what they are currently set to.

- The device would have a pump system for detecting the free chlorine and phosphate test chambers with pre-loaded reagents.
- The device would have sensors for measuring the waste container's capacity.

2.3.2.2 Software Objectives

In order to meet the software goals mentioned above the project met the following objectives.

- The system has a web server on a cloud provider
- The system has a database to store user credentials and data collected from the device
- The web server has a front end website that allows the user to communicate with the device
- The website allows the user to create a new account and login to that account.
- The website allows the user to add devices to their account based off the device's serial number
- The website displays the device's current battery charge, time from last test, and if any non-power systems are down.
- The website displays the results from the free chlorine detection system, phosphate detection system, depth test, and temperature test, with timestamps.
- The website displays the results of the previous tests.
- The website allows the user to change the sampling rate of the device.
- The website allows the user to request tests to be run on the device.
- Measurements on the website are displayed in common American units.
- The website has a domain name to facilitate easy access.
- The website allows the user to set up automatic notifications for email and other servers, based on thresholds being crossed, such as temperature exceeding 100 degrees.
- The web server is fully containerized to allow for immediate deployment on other computers.
- The webserver is hosted on a kubernetes cluster and configurable using environment variables to allow modification in future deployments.
- The device generates HTTP messages to the web server to facilitate outbound communication.
- The device is able to parse HTTP messages to facilitate inbound messages.
- The server controls when a test is run and is able to compute the outcome of the test into common American units.
- The device manages the water drawing system.
- The device monitors the battery level and updates the site on the charge.
- The device is able to diagnose non-power system failures and display them on the debug LCDs, as well as update the webserver of their failure.
- The device has a hard coded serial number.

To achieve the stretch goals mentioned above the software would need to meet the following objectives.

- Web server communication is done using ssl to secure the data transferred.
- The web server would pull weather statistics and notify the user of selected events such as heavy rain.

- The website would be fully phone compatible.
- The device would facilitate an LCD display and communicate configuration variables.
- The device would be able to handle keyboard inputs and the changing of configuration variables.
- The device would handle the changing of the molybdophosphoric acid and chlorine test chambers.
- The device would notify the web server for when the waste container needs to be changed.
- The web server would notify the user for when the waste container needs to be changed.

2.3.2.3 Optical Objectives

In order to meet the optical goals specified above the following objectives must be met.

Basic Objectives

- Design lens system with a photodiode that can detect Phosphate and Chlorine using a micro-cuvette to plot various concentrations on a calibration curve.
- Design a lens system with a photodiode that can detect Chlorine.
- Design a magnifying lens system using singlet lenses with a camera to image particulate.

Advanced Objectives

- Use Achromatic lenses to minimize comatic aberration in order to capture a larger field of view.
- Use a multi lens system to minimize spherical aberration and collimate the light.
- Optimize the alignment so the optical path length increases.
- Change out the capacitor and resistor to larger values.

Stretch Objectives

- Use UV-Vis spectroscopy to select various wavelengths using a pin hole to detect chlorine and phosphate concentration in a singular cuvette.
- Use specialized lenses with a large imaging sensor to achieve a high magnification system with a wide field of view.
- Use a partially transparent mirror system to direct some light upwards to a secondary imaging system while still allowing some light to reach the chlorine measurement system.

- The device has an optical system that measures the free chlorine concentration after reacting with a DPD in the pool water.
- The device has an optical system that measures the molybdophosphoric acid concentration after reacting with Phosphate, Molybdate, and Ascorbic acid in the pool water.
- The device has an optical system for measuring the amount and size of particulate in the pool sample
- The sensors communicate with the MCU.
- The sensors are power efficient.

2.3.2.4 Mechanical Objectives

In order to meet the mechanical goals stated above, the project met the following objectives.

- The device could fit within a non disruptive space by the side of the pool.
- The device supports drawing operations
- Accessing components of the device and changing them out is manageable with a common multitool, such as a Leatherman.
- The device has a debug section displaying the LEDs indicating the working systems.
- The device's power source is hand changeable.
- The device has an on switch and a debug switch section.

To achieve the stretch goals for the mechanical design the device would meet the following objectives:

- The device would be splash resistant.
- The device would have an LCD display and a keyboard.
- The device would support a storage container for the reagents including Molbudate, ascorbic acid, and free chlorine testing.
- The device would have a waste container for the molybdophosphoric acid and chlorine testing waste.
- The waste container would not need any necessary PPE to handle.

2.3.3 Optical Diagrams

To meet those goals and apply those objectives the following optical diagrams are used. Below is the optical design for the Particulate Imager and the Chlorine/Phosphate Detector. For the Particulate Imager (**Figure 2.1**), it is placed perpendicular to the cuvette being used for the chlorine measurement. It features the same LED which is collimated and sent through the cuvette. Within the cuvette there will be particles which will then cast a shadow onto the opposite wall. On the other side a set of lenses is used to focus down the light onto a camera. This camera leads to a Raspberry Pi where the image is processed, the number and size of the particulate within the cuvette is recorded. This is done several times and averaged for the most accurate results. For the Chlorine/Phosphate detector (**Figure 2.2**), a small amount of pool water is pumped up from the pool into two Eppendorf Cuvettes where one is mixed with N,N-diethyl-p-phenylenediamine (DPD) powder to react with chlorine, and the other mixed with Ammonium Molybdate and ascorbic acid to react with phosphate to detect the presence of algae. This causes a reaction to occur changing the color of the water, also changing its absorption coefficient for each respective wavelength. Light emitting diodes can then be used to send light in wavelength relating to the highest absorption of the respective products, after the light passes through the cuvette, the intensity of the light can be measured using a photodiode. Based on this measurement the Chlorine/Phosphate concentration can be found.

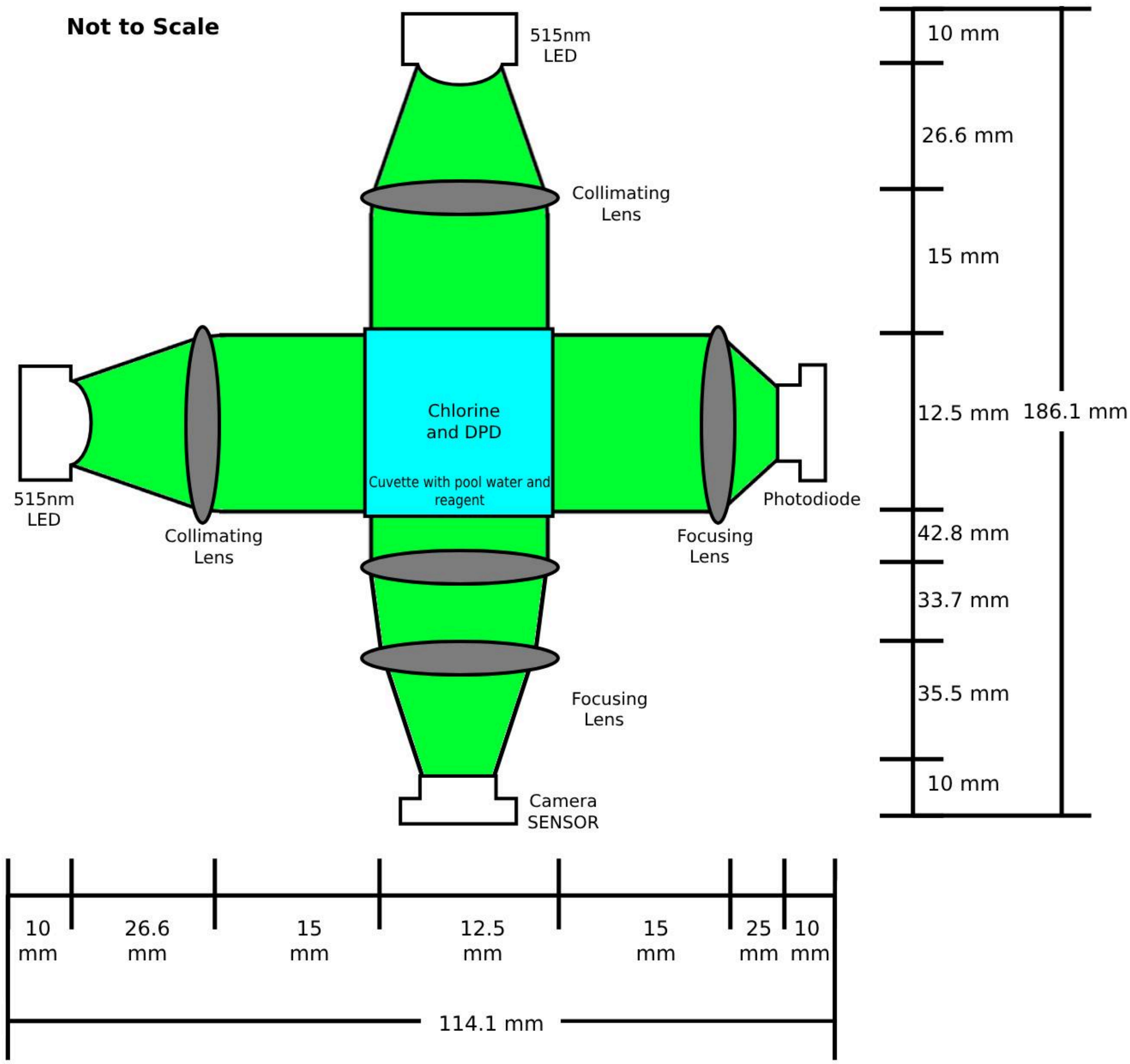


Figure 2.1: Particulate Imager Optical Design

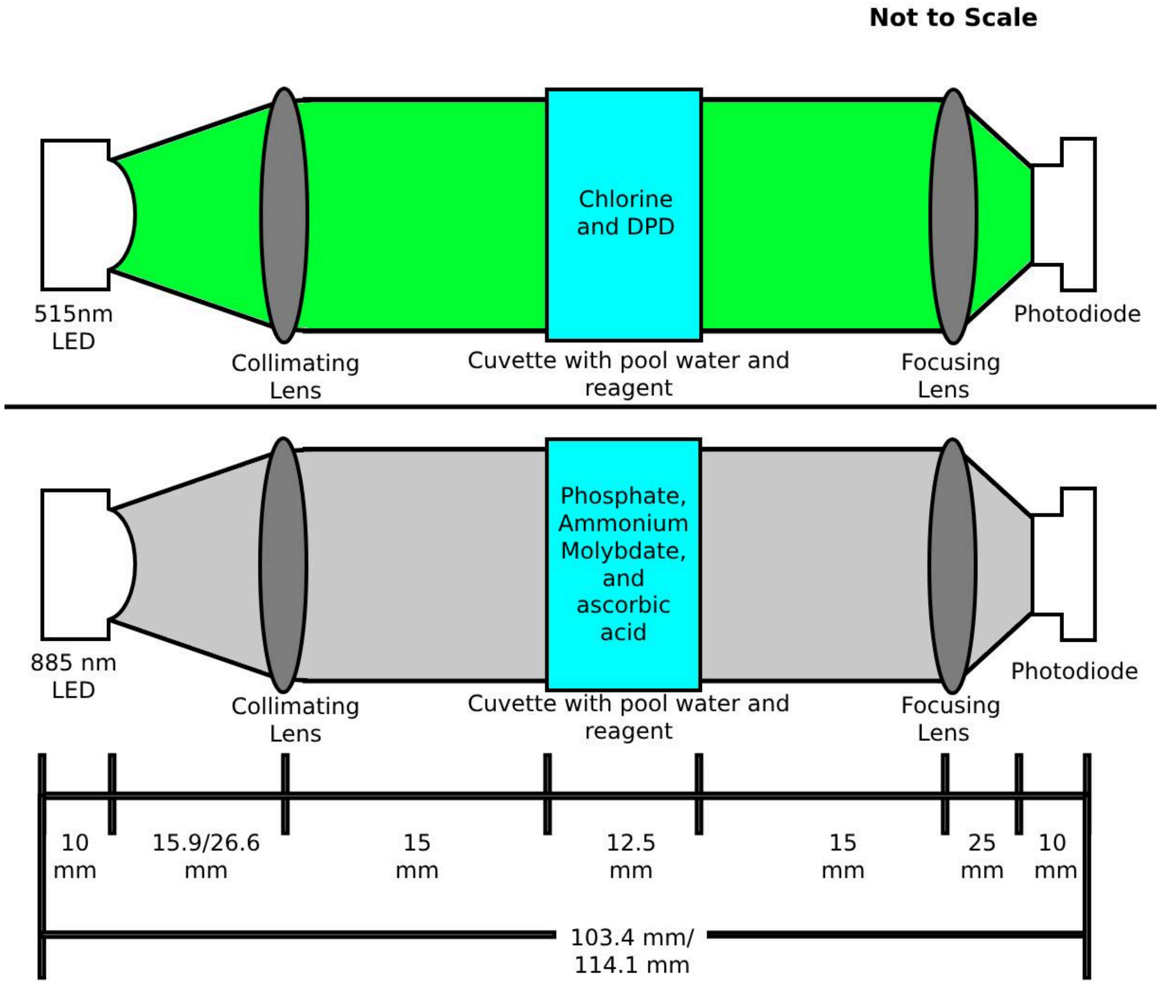


Figure 2.2: Chlorine and Phosphate Detector Design

2.4 Features and Functionalities of PoolWatch System

2.4.1 Web Server Features

The web server has the following features and functionalities.

- Account system.
- Containerized.
- Database.
- Ability to add devices.
- Device report display
- Notification configuration.
- Domain name.
- CRUD operations.
- Device management operations
- Device test requests

2.4.2 Electrical Features

The device has the following electrical features.
- Serial number.
- Temperature sensor.
- Robust power system.
- Troubleshooting panel.
- WiFi capabilities.
- Replaceable parts.
- Water pump system.
- Battery monitoring system.

2.4.3 Optical Features

The device has the following optical features.
- Light Emitting Diodes at various wavelengths.
- Collimating lens system.
- Focusing lens system.
- UV quartz cuvettes sampling system.
- Optical narrow bandpass filters.
- Achromatic lens imaging system.

2.4.4 Electrical Functionality

Some of the electrical functionalities include a serial number to identify each deployed device to support identification, usage tracking, and maintenance. The power supply system guarantees optimal performance and stability for managing a large number of operating procedures on a rechargeable battery. The thermistor collects pool water temperature for viewing in the web app. A troubleshooting panel is present for user friendly maintenance. The overall system is compatible with high-speed low latency wireless technology available in the United States. Individual parts of the system have reduced lead times and can be easily exchanged if broken or damaged, without requiring specialized, industrial tools. To bring pool water into the testing chamber automatically, a novel water pump system is used. Our system has the ability to monitor the health of the battery that is powering all systems included in the main chassis.

2.4.5 Optical Functionality

The device has the following optical functionalities in the final design. Light Emitting Diodes at 515 nm and 880 nm wavelengths for optical measurements and other visible wavelengths are used for indicators. The system has collimating lenses for minimal light loss and accurate measurement. Another lens system is present to focus the beam with minimum comatic aberration onto the photodiode for chlorine and phosphate measurements. UV quartz cuvettes with 90% transmittance are included in each optical system. Optical narrow bandpass filters with a 10 nm Full Width at Half Maximum are used to reduce any parasitic wavelengths and increase the SNR. Lastly, an imaging system consisting of achromatic lenses that is able to observe 10 µm feature sizes inside the cuvette.

2.5 Engineering Specifications

To ensure that the device is able to meet the objectives laid out above, it is important to specify the requirements that put us on the correct path to achieve them. Below is the specification table that details these objectives.

Table 2.1: Engineering Specifications

Component(s)	**Parameter**	**Specification**
Particulate Imager	Resolution	≤ 10 µm
Thermometer	Accuracy	± 0.4 F
Chlorine mass Concentration	Accuracy	± 0.5 ppm
Particulate Imager	Field of View	5 mm x 5 mm
Chlorine mass Concentration	Measurement Range	≥ 0.30 ppm
Chlorine LED Emission	Wavelength Range	515 nm ± 2 nm
Chlorine Optical Narrow Bandpass Filter	Central Wavelength	515 nm
Chlorine Concentration Analysis	Response Time	≤ 1 minutes 30 seconds
Phosphate Mass Concentration	Accuracy	± 0.5 ppm
Phosphate Mass Concentration	Measurement Range	≥ 0.30 ppm
Phosphate LED Emission	Wavelength Range	880 nm ± 2 nm
Phosphate Optical Narrow Bandpass Filter	Central Wavelength	880 nm
Phosphate Concentration Analysis	Response Time	≤ 1 minutes 30 seconds

Battery	Capacity	≥ 5000 mAh
Particulate Imager	Response Time	≤ 1 minute
UV Quartz Cuvette	Transmittance	90%
UV Quartz Cuvette	Allowable Volume Content	≤ 3.5 mL
Thermometer	Response Time	≤ 15 seconds
Container	Dimensions	≤ 18” x 12” x 6”
Power Unit	Max Power Consumption	≤ 24 Watts
Power Unit	Average Power Consumption	≤ 1 Watt
System	Battery Life	≥ 30 Days
System	Cost	≤ $1,500

These specifications helped us provide a great product for the potential customer. The House of Quality Diagram below details the marketing requirements and how they had been approached with engineering specifications.

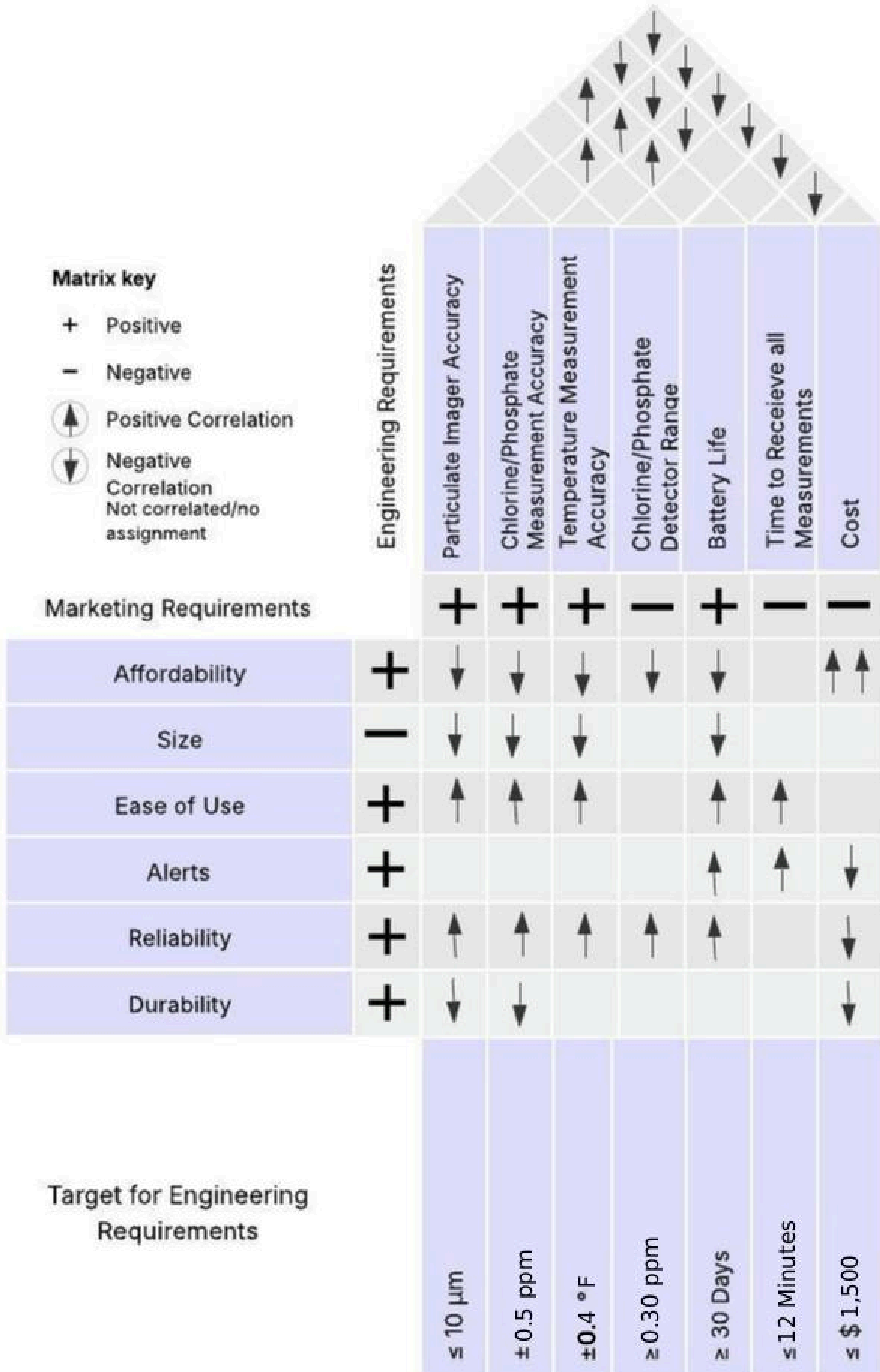


Figure 2.3: House of Quality

Below is the Hardware Block Diagram that illustrates the main hardware objectives we are trying to accomplish in this project.

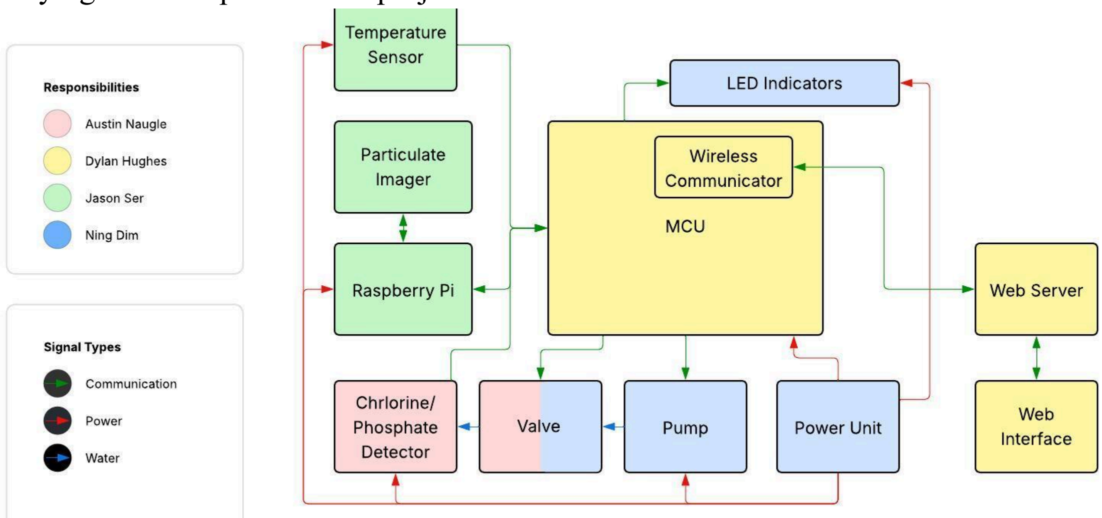


Figure 2.4: Hardware Block Diagram

Below is the Software Block Diagram that illustrates the main objectives we are trying to accomplish in this project.

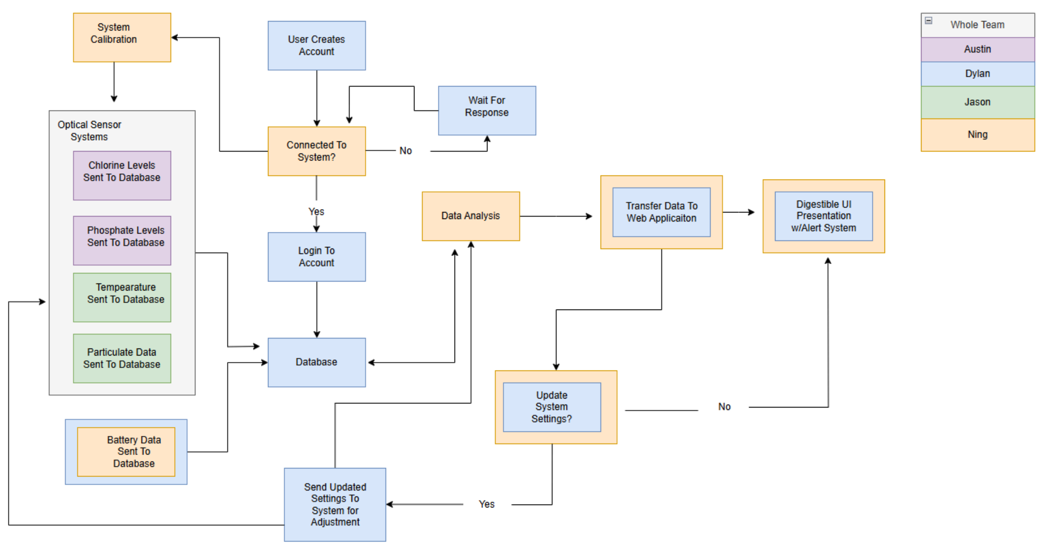


Figure 2.5: Software Block Diagram

The following is an illustration of the device showing the basic shape and layout.

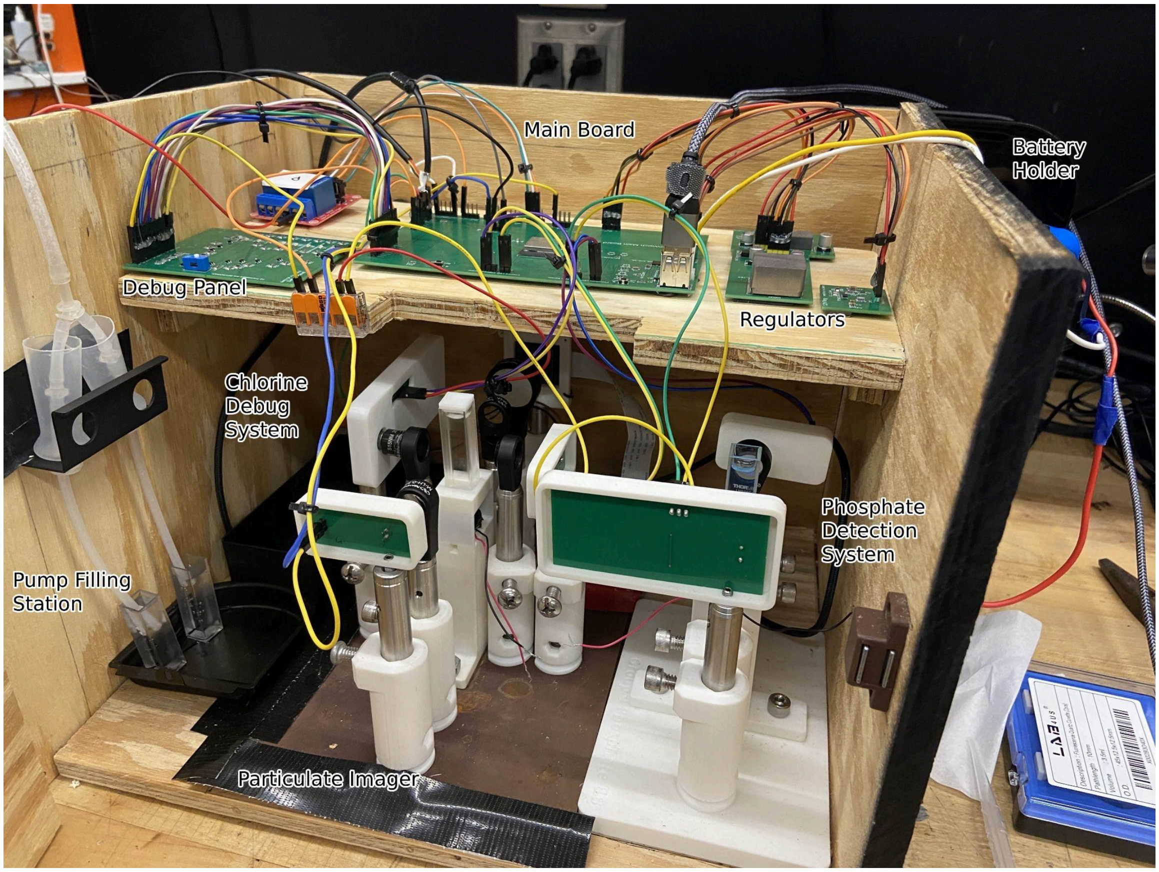


Figure 2.6: Shape and Layout Prototypes

Chapter 3 Research and Investigation

3.1 MCUs

When selecting the microcontroller unit (MCU) and corresponding development board, several factors were considered: familiarity, ease of use, low power mode capability, and cost. For this project, the MCU must support Wi-Fi to transmit data from the optical systems to a database for the web app. Two of the three optical systems require analog-to-digital conversion (ADC), so multiple ADC channels are necessary. Additionally, SPI or I2C bus capabilities are essential for processing data from connected sensors. The third optical system requires more advanced processing power to measure particulate concentration. Therefore, the selected MCU should have sufficient memory and processing power to perform the required calculations locally if needed, although offloading the computation to a remote database remains a viable alternative depending on final system constraints.

3.1.1 Arduino Uno Rev3

The Arduino Uno Rev3 is an 8-bit microcontroller that was considered due to its ease of prototyping, extensive library support (particularly for servo motors), and robust online documentation available. Familiarity with both the board and the Arduino IDE also makes it a user-friendly choice. It also features multiple ADC channels, which are vital for interfacing with the optical systems, and extensive peripheral support for future expansion if stretch goals are achieved.

A major advantage of this board is its availability, as it is already on hand for the project, eliminating the need to purchase additional hardware. Another significant benefit is the abundance of projects, resources, and documented use cases associated with the Arduino Uno. That large reference base provides support when applying best practices and implementation strategies, making it easier to integrate more advanced and autonomous functionality into the system.

Some drawbacks include the lack of built-in Wi-Fi, a key requirement for the planned IoT functionality. This limitation can be addressed by adding a Wi-Fi shield or integrating an external Wi-Fi module. Another drawback is the board's limited memory and relatively low processing speed. While sensor data is not stored locally, the limited memory means code must be highly efficient to ensure reliable operation within the constrained resources. A potential workaround would be to pair the Arduino with a more powerful board like the ESP32, which could handle the more demanding computational tasks and communicate with the Arduino as needed.

The ESP32 uses the ATmega328P MCU with a clock speed of 16MHz. It also provides extensive peripheral support, including more than 12 ADC input channels, 3 UARTs, 3 SPI, 2 I2C interfaces, and a 32 kHz crystal oscillator for timekeeping synchronization and sleep timer functionality, making it suitable for a sensor-rich, battery-powered IoT systems **[65]**.

3.1.2 Arduino Uno R4 WiFi

The Arduino Uno R4 WiFi is a 32-bit microcontroller with an integrated ESP32 module for Wi-Fi connectivity. It was considered because it retains many of the same capabilities as the Uno R3 while offering enhanced performance, greater memory, and additional features through its advanced Renesas processor and built-in wireless support.

The Uno R4 WiFi is built around the 48 MHz RA4M1 microcontroller with a floating point unit (FPU). It features 256 kB of flash memory, 32 kB of SRAM, and supports peripherals such as a 14-bit ADC, 12-bit DAC, operational amplifier, capacitive touch sensing, and a real-time clock(RTC). It offers 1 UART, 1 SPI, 2 I2C, and 1 CAN interface, making it suitable for projects that require precise calculations and analog sensor support **[66]**.

The R4 WiFi was a key consideration because it uses the Arduino IDE and supports many of the same libraries as the Rev3, while also offering built-in Wi-Fi for IoT applications. The main drawback is that, being newer than the other two options listed, there is currently less documentation and community support available if there are issues.

3.1.3 Adafruit HUZZAH32 ESP32 Feather

Another MCU considered was the ESP32, selected primarily for its built-in Wi-Fi, which makes it ideal for IoT applications. Notable hardware features include a 240 MHz dual-core Tensilica LX6 processor, 520 KB of SRAM, and 4 MB of flash memory, which is significantly more memory and processing power than the Arduino Uno Rev3. The ESP32 also provides extensive peripheral support, including over 12 ADC input channels, 3 UARTs, 3 SPI, 2 I2C interfaces, and a 32 kHz crystal oscillator for timing management and sleep functions **[67]**.

One key advantage is its compatibility with both the Arduino IDE and the lower-level ESP32 IDF, allowing for flexible development based on user experience. The widespread popularity among hobbyists and developers also ensures that there is large documentation available, for both troubleshooting and expansion. Another major benefit is the ESP32's ability to operate in three power modes: active, light sleep, and deep sleep. In light sleep mode, power consumption is significantly reduced by disconnecting Wi-Fi and maintaining only the internal clock and memory, allowing the MCU to resume operations right where it left off upon waking. In contrast, deep sleep mode conserves even more power by shutting down nearly all components except the real-time clock (RTC), which is used to trigger fast wake-up events. This feature is especially valuable for battery-powered systems aiming for long-term energy efficiency, as it allows the MCU to spend more time in sleep mode when not needed.

A disadvantage of the ESP32 is that it is more complex to work with than the Uno Rev 3, as the development process is not as straightforward and may involve more advanced coding. Another drawback is that the board runs on 3.3 V power and logic, which makes it more sensitive to overvoltage damage. Without proper precautions, this board is more vulnerable to damage than boards like the Arduino Uno, which can tolerate higher input voltages.

3.1.4 MCU Selection

Table 3.1 below lists the microcontrollers (MCUs) considered for the system. Based on our requirements, the Adafruit Huzzah32 was selected as the MCU that best meets our needs. Its numerous GPIO pins, low operating voltage, high clock speed, and compact form factor make it the most suitable choice for our system.

Table 3.1: MCU Comparison

Features	Arduino Uno Rev3	Arduino Uno R4 WiFi	Adafruit HUZZAH32 ESP32

Operating Voltage	5V operating, 7-12 V (rec'd)	5V operating, 6-24 V (rec'd)	3.0-3.3 V
Chip	Atmega328P	Renesas RA4M1	Tensilica LX6
Memory	32 kB Flash, 2KB SRAM	256 kB Flash, 36 kB SRAM	4 MB Flash ,520 kB SRAM
Interfaces	I2C, SPI, UART, USB	I2C, SPI, UART, CAN, DAC	3 UART, 3 SPI, 2I2C, 2 DAC, 12 ADC
GPIO pins	14	14	21
Clock Speed	16 MHz	48 MHz, ESP32 is 240 MHz	240 MHz
Dimensions	68.6 x 53.4 mm	68.85 x 53.34 mm	51.0 x 22.7 x 7.3 mm
Cost	$22.08	$22.00	$ 19.95
Quiescent Current (Iq)	45mA-80mA	82.86 mA - 124.04 mA	100 uA
Low Power Mode	yes	yes	yes
Wi-Fi	No	Yes	Yes

3.2 Processor Selection

To capture the photos a system is required to capture the photos and then process them. Our processor does not have enough space to take in a full photo so a Raspberry Pi will be used for all processing and image capturing. Once this processing happens it would then send the important data to the central processor. It is important to select the Raspberry Pi that is able to process the high resolution images in a reasonable amount of time.

3.2.1 Raspberry Pi 5

The Raspberry Pi 5 features the fastest processor, as well as the capability of 16GB of RAM which is more than all of the other Raspberry Pis. It also has two separate MIPI lanes which allows for the ability to capture with two separate cameras. The I/O on the Raspberry Pi are also the fastest of all of the other Pis. The drawbacks of the Raspberry Pi 5 are its high power draw of 27W and its higher price point **[97]**.

3.2.2 Raspberry Pi 4

The Raspberry Pi 4 is very similar to the 5 in terms of I/O but lags behind in the speed of all components. It also is limited to one MIPI lane limiting it to one camera. Though these drawbacks are slightly negated by its cheaper price point and lower power draw of 15W **[98]**.

3.2.3 Raspberry Pi Zero

The Zero differs greatly compared to the 5 and 4 in terms of available connections. It is a much more bare bones system. It still has the essentials needed for the system with a very cheap price point of 10 dollars. The power draw is also down to 13W, less than both other systems. However, this bare bone design creates drawbacks in terms of general speed. The Zero is down to only half a Gig of RAM and a slower processor. The board would also require more modification in order to connect the I2C needed to connect it to the MCU **[99]**.

3.2.4 Summary

Overall the Raspberry Pi 4 seems to be the best system for the job. Its middle of the road processing speed, cost, and power draw make it a great choice for our design. See **Table 3.2** for more detail.

Table 3.2: Comparison of Processing Units

Raspberry Pi 5	Raspberry Pi 4	Raspberry Pi Zero
+ + Processing Power + Connection - - Power	+ Processing Power + Connection - Power	- - Processing Power - Connection + Power

3.3 Fluid Handling System

Our aim was to design a system with easily replaceable parts to keep maintenance labor and costs as low as possible. Since automation and a month-long battery life are key goals of the project, low-power hardware selections will need to be balanced with smart mechanical design choices to minimize energy consumption during the operations.

The fluid handling subsystem consists of pumps and valves, each playing a critical role in transporting and controlling the flow of liquids throughout the device. Pumps enable the movement of fluids, while valves are used to direct, isolate, or prevent flow, allowing for automated switching between sources and precise timing during sample and reagent delivery. This section evaluates pumps and valve options for water and reagent handling, with reasons for the final selections.

3.3.1 Water Pump

In this section, we evaluate four common types of pumps under consideration for the main water pump: peristaltic, diaphragm, submersible and syringe pumps. These pumps were selected for their proven use in other fluid-handling applications such as aquariums and hydroponic systems, as well as for features like self-priming or immersion-ready designs, which supports reliable startup under many different operating conditions.

As part of the selection process, we considered several practical factors, including cost, setup complexity, voltage and current requirements, max head, and flow rate control. A critical parameter, the pump's maximum head or the maximum lifting height, is the highest vertical distance a pump can move water. The max head value is given in feet or meters and is measured from the pump's outlet to the discharge point. For the water flow to reach the desired height, the pump's max head must be greater than the vertical distance, since there is little to no flow at the max head. Therefore, pumps with a max head greater than 1 foot were considered for the main water pump.

Our fluid handling system requires delivery of pool water to two separate optical analysis subsystems. Each subsystem includes a cuvette in which the pool water must be mixed with a specific reagent, without splashing. To ensure timely and accurate results, the system must dispense precise volumes of water into each cuvette, enabling the full test sequence to complete within 4 minutes and 30 seconds. Furthermore, because the system is designed to support either parallel or independent test execution, the water pumping subsystem must be capable of selectively directing flow to each cuvette as needed.

A rough estimate of the required flow rate can be obtained by dividing the total volume by the available time to run the tests. When both tests are run in parallel, the total required volume is 7 mL over a duration of 3 minutes and 30 seconds (with 1 minute reserved for reagent dispensing and test execution), resulting in an estimated flow rate of approximately 2 mL/min (0.002 L/min). As long as the selected pump can operate within this range, it is suitable for delivering the necessary volume within the given time constraints.

3.3.1.1 Peristaltic Pump

The peristaltic pump operates using a rotary mechanism, typically with either rollers or shoes, to compress flexible tubing to move liquids through the system. Flow rate is controlled precisely by the rotation speed of the rollers, as each full rotation displaces a fixed volume of liquid through the tubing. This positive displacement action of the pump prevents backflow and reversing roller polarity enables bidirectional flow of liquid. Key advantages include precision dosing, bidirectional flow, and contamination avoidance, since only tubing touches the liquid. Another key advantage is that peristaltic pumps can be 3D printed due to their simple mechanical structure, allowing for budget customization and prototyping.

Peristaltic pumps are designed with specific flow rates which is important to consider when selecting the appropriate pump to use. Choosing one that is too high or too low could lead to poor performance and inaccuracy. Tubing is another critical factor when selecting a peristaltic pump, as it influences the flow rate and must be compatible with the system layout. Smaller diameter tubing for example, would result in a more consistent performance at low flow rates, since there is less pulsation **[100]**. The increased flow resistance of narrower tubing helps dampen pressure fluctuations as well as sudden flow changes **[100]**. Tubing is needed to be routed from the pool through various components before being deposited directly into the cuvettes, depending on the final system layout. Additional attachments or connectors may be required, which can directly influence which peristaltic pumps are well-suited for the final design. This type of pump may be referenced again in the dosing system section, which could utilize the same type of fluid delivery method.

3.3.1.2 Diaphragm Pump

A diaphragm pump is a positive displacement pump that uses a flexible membrane, called a diaphragm, which moves back and forth to create a temporary chamber. This chamber pulls fluid in through a one-way inlet valve and then expels it through a one-way outlet valve. There are two main types of diaphragm pumps, air-operated and electric. For use as the main water pump, the electric diaphragm pump was considered as they are more energy efficient, capable of more precise control, quieter, and are lower maintenance due to having fewer parts.

Electric diaphragm pumps are usually able to replace peristaltic pumps in most use cases but come with a few key limitations. One drawback of diaphragm pumps is the difficulty in achieving a smooth, continuous flow since their operating mechanism results in a pulsating output. In contrast, peristaltic pumps offer more consistent and controllable flow and their flow rate can be precisely adjusted using PWM signals or motor speed control, making them better suited for applications requiring accurate dosing or steady delivery. Furthermore, diaphragm pumps have more internal components, making maintenance more complex and replacement more involved in the event of a mechanical failure. Finally, the liquid being moved would touch the inner components of the pump, which might be an issue if contamination avoidance was needed.

Diaphragm pumps offer several advantages that still make them a strong contender for use as the main water pump. In the absence of mechanical failure, diaphragm pumps can be more cost-effective over the long term, as they do not require regular tubing replacement. The stronger forward stroke of a diaphragm pump also makes it more energy-efficient and better suited than a peristaltic pump for overcoming standing pressure in the flow line, should that be a concern.

3.3.1.3 Submersible Pump

A submersible pump must be fully submerged in liquid to operate, which limits its placement options and must be considered during system design. When powered, the impeller-a rotating fan-spins to generate centrifugal force, creating a low-pressure area at the inlet that draws in liquid. This force then propels the liquid outward through the pump’s outlet. Submersion allows the pump to maintain a fast, continuous liquid intake, resulting in a more efficient operation by preventing overheating, air locking, and cavitation (rapid creation then collapse of air bubbles in fluid). However, it also makes regular access for maintenance more difficult and less convenient, which is the main disadvantage.

The submersion design limits its use case in the water pump subsystem, as it must remain in the pool and can only draw water up to a mixing chamber or cuvette. This type of pump offers several advantages, mainly the ability to transfer large volumes of water quickly, self-priming capability, and a compact form in both vertical and horizontal configurations. A stronger pumping action might be useful if increased water pressure is required to split the incoming pool water and direct it to both optical system cuvettes. The submersible pump does not offer the same level of dosing control as the other two systems, which means it would likely need to be integrated with an additional system to fulfill the full range of operational requirements. An important point to address with this type of pump is how it would be positioned within the pool. Since it has to stay submerged in order to operate, brackets or a custom shelf would need to be designed to hold it at a specific depth below the water's surface. Since our demonstration would most likely be in a tank, this support structure would need to be incorporated into the final design.

3.3.1.4 Syringe Pump

A syringe pump offers the most accurate and precise dosing and is commonly used in laboratory and medical settings where reproducible, high-precision fluid delivery is critical. It was initially considered for our system due to these advantages. However, syringe pumps also have some large drawbacks, mainly speed and cost. Syringe pumps are typically the slowest of the three dosing mechanisms under consideration, and they require the most power and complex control requirements, as they rely on servo motors for actuation. While a DIY syringe pump was also considered as a lower-cost alternative, the combined material cost and labor required would still exceed that of the other prebuilt, lower-voltage options considered. Therefore, this type of dosing mechanism is not utilized in this system.

Table 3.3 shows the four main water pump contenders and the two most critical features considered for this application. The peristaltic pump was chosen for its ability to deliver precise volume control at low flow rates, its self-priming capability, low maintenance requirements, wide model options, and simple mechanical operation. The peristaltic pump also has backflow prevention to support accurate dispensing with minimal splashing when filling cuvettes. Overall, the peristaltic pump is a cost-effective and reliable choice that meets the system's key water pump requirements.

Table 3.3: Water Pump Type Comparison

Features	**Peristaltic**	**Diaphragm**	**Submersible**	**Syringe**
Control	High	Medium	Low	High

Flow Rate	Medium	Medium	High	Low
Self-Priming	Yes	Yes	No	No

3.3.2 Water Pump Selection

The water pump selection was based on two possible use cases that were considered in our selection criteria. In the first use case, the pump draws water from the pool and directly fills the cuvettes. In the second use case, the pump supplies water to a holding or mixing tank for further distribution. Having preliminary design sketches also helped narrow down suitable pump options and ensure the solution can support multiple use cases. **Fig. 3.1** shows an initial rough sketch intended to visualize potential implementations of water pumps within the system. The left and right sides of the sketch are separated to represent the two split optical chambers that need water to be delivered to the cuvettes located inside. The blue tube represents the water pump drawing pool water up and then being dispensed directly into the cuvettes on the left-hand side via a valve. On the right-hand side, it shows the water being dispensed into a cuvette using a peristaltic pump. Using this design as a basic guide, specific pumps would be chosen based on their ability to fulfill the two defined use cases.

As the design progressed, the system requirements shifted toward reliably filling only two cuvettes, rather than supporting both of the original use cases. With this change, pump selection became more targeted, focusing solely on pump types capable of consistently supplying water for these two cuvettes. Another consideration is that the pump is currently intended to operate using a simple on/off control, with no variable-speed functionality planned for implementation. As a result, the pump always runs at full speed, making it important to select models that offer naturally low flow rates or slow operating speeds. Other options will also be listed in case variable control needs to be implemented in later design stages.

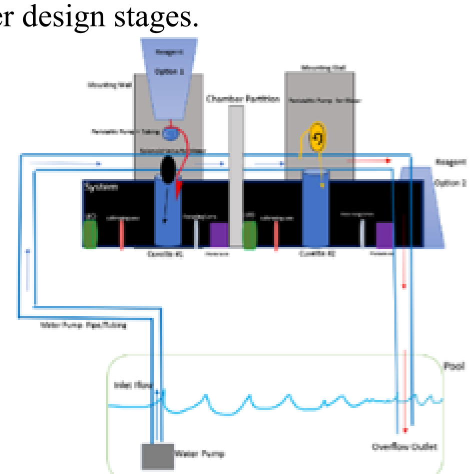


Figure 3.1: Initial Water Pump Design

Table 3.4 below summarizes the three main peristaltic pumps under consideration. The models may be referenced again in the dosing system, which could require the same type of fluid delivery method. The first pump in the table was selected for its lower flow rate, small tubing for low pulsation, perfect for applications with compact setups with minimal piping. The second pump delivers a higher flow rate and is preferred when a stronger flow is needed, depending on the total length and complexity of the piping system. The first pump was chosen for being able to be powered from a voltage range of 6-12V, extensive documentation, and low current requirements.

Table 3.4: Peristaltic Pump Comparison **[55,56,57]**

Features	**Gikfun Peristaltic Pump**	**Kamoer Peristaltic Pump**	**Micro Peristaltic Pump**
Voltage Requirement	12 V	12 V	6V-12V
Max Current	0.220 A	0.42A	0.5A
Flow Rate	max 0.5 L/min	0.7 L/min	30mL/min-100mL/min
Max Head Height	5m (16.5 ft)	5m (16.5 ft)	N/A (reviews say 5-6.5ft)
Tubing Dimensions	2mm ID and 4mm OD	3mm ID and 5mm OD	2.5mm ID and 4.5mm OD
Cost	$11.98 (Amazon)	$10 (Amazon)	$1.40(Aliexpress)

For the final pump design in Senior Design II, the Gikfun peristaltic pump was chosen because it provided the naturally low flow rate our system needed. Since we did not require variable-speed control in the final implementation, this pump was more than sufficient, as its fixed-speed operation consistently delivered the slow and steady flow needed to fill both cuvettes. The basic on/off control fit directly with our design approach, and the pump consistently delivered repeatable volumes without the need for extra hardware or adjustments.

3.3.2.1 Tubing Material Comparison

In this project, there must be an efficient method of transporting the pool water and reagent from the peristaltic pump and cartridges to a 45x12.5x12.5 mm cuvette. Any sort of 3-D hollow cylindrical object can transport a liquid, but a long flexible tube would make the cost and pressure best to use for this system. There are different properties associated with tubing that are available to transfer liquids with a flow rate from one place to another.

Table 3.5: Material Selection Needed for Transporting Water and Reagents

Material	Overall Cost	Is it flexible?	Chemical Resistance	Pressure Applications
Polyethylene	Low	Yes	Medium	Low & High
Steel	High	No	High	High
Nylon	High	Yes	High	High

In **Table 3.5** various materials were chosen to see how well they could be used in the pump/valve system. Steel and Nylon do not corrode as much at elevated temperatures, whereas polyethylene does. This was not included in the table since thermal conductors were used for optimal tubing conditions. Polyethylene was the best choice in terms of cost since it's cheaper to manufacture compared to nylon and steel. However, polyethylene does not resist various chemicals as well compared to steel and nylon. Although this system only uses an acid or dye, this would not negatively affect the material.

Moreover, each material listed does have some major drawbacks that led to the final decision. For this current system, the tubing needs to be somewhat elastic so it does not expand or conduct at very high temperatures. Also, since the overall shape of the packaging is going to be rigid, the tubing must be flexible. So, this leaves out steel because it would be highly disadvantageous to use a material with a high thermal conductivity and thermal expansion coefficients.

An alternative choice to polyethylene would be nylon since both can be observed to have a polymer structure. However, the cost of nylon is much higher compared to polyethylene, so the latter is used in the final design. The total cost was not considered because the individual contributed their own that they already paid for and used in other projects.

Table 3.6: Material Selection Needed for Transporting Water and Reagents

Material	Relative Cost	Chemical Resistance	Thermal Distortion
Plastic (Kynar)	$6.77	High	Low
Brass	$10.34	High	High
Stainless Steel	$123.70	High	High

Some of the materials that were chosen for comparison shown in **Table 3.6** can be laid out in various configurations. For example, there are push-to-connect, quick assembly, and compression tube fittings. All of these are made out of a different material, so selecting the best one for this system was required. The environment the tubing is going to be placed in is outside near a swimming pool, which means the tee-junction must be able function properly at high temperatures.

Some reagents change the chemical composition and become corrosive. To compensate for this the following comparisons were made so this would not happen. However, all materials that were selected would have a high tolerance for the reagents being used.

3.3.2.2 Tee-Junctions

To ensure the proper fittings on the polyethylene tubing a tee-junction is needed so any excess pool water or reagent will be separated from each other.

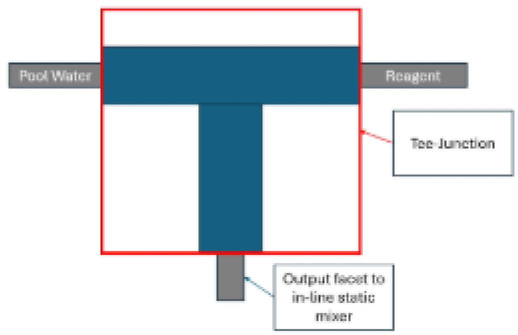


Figure 3.2: Potential Tee-Junction Design

When testing T-junctions in Senior Design II, we found that they did not distribute water evenly between the two cuvettes. One side consistently received a greater volume, which made them unsuitable for our setup. Because of this, we moved to Y-junctions instead, which provided noticeably more balanced flow and resulted in more even dispersal of water into both cuvettes.

3.3.3. Solenoid Valves

Valves are designed for many different applications, but the ones relevant to our system are those used to start or stop fluid flow, prevent reverse flow, and redirect fluid between pathways. In our system, solenoid valves are the primary component considered for the dosing mechanism. Solenoid valves were selected due to their significantly faster response time for start/stop operations and their minimal part requirements, which streamlines integration and saves assembly time.

Solenoid valves are electromechanical devices consisting of two main components: the solenoid and the valve body. The solenoid is an electric coil that, when energized, creates a magnetic field that moves a small metal rod called a plunger against the force of a spring to control the flow of a medium. Solenoid valves are available in either a normally open (NO) or normally closed (NC) configuration, referring to their default state when no power is applied. When energized, the valve switches to the opposite state, either allowing or stopping fluid flow.

Only normally closed solenoid valves were considered, as they help minimize power consumption in gravity-fed systems by remaining closed unless fluid flow is explicitly needed. As a result, power is used only during valve activation. Solenoid valves are generally classified into two main types based on their operating principle: direct-acting and pilot-operated.

In direct-acting valves, the solenoid directly moves the plunger to open or close the valve. These valves do not require any minimum fluid pressure to operate, making them suitable for conditions such as vacuum, zero, and negative pressure. In contrast, pilot-operated valves require a minimum pressure differential to assist in valve actuation. The solenoid opens a small pilot port, which allows system pressure to shift across a diaphragm or piston and move the main valve open or closed. Even though direct-acting solenoid valves consume more power than pilot-operated types, they offer faster response times and greater versatility in low-pressure environments.

Table 3.7 below provides a summary for the two types of solenoid valves discussed. While direct-acting solenoid valves are more reliable in low-pressure systems, they are significantly more expensive and would consume more of our budget. Even the lower-cost options often receive poor reviews citing leakage problems, which made it challenging to identify reliable models. Pilot-operated solenoid valves are typically more affordable and easier to source, but their minimum pressure requirement raised concerns for our system. Because the dosing setup is gravity-fed and the reagent volumes are small, meeting the activation pressure for a pilot-operated valve would not be ensured. For this reason, direct-operated valves were considered the more appropriate choice for our design.

Feature	Direct-Acting	Pilot-Operated
Minimum Pressure Needed	No	Yes
Complexity	Simple	More Complex
Response Time	Fast	Slower
Power Consumption	Greater	Lower
Cost	Higher	Lower

Table 3.7: Solenoid Valve Type Comparison

3.3.4 Solenoid Valve Selection

There are many types of solenoid valves to be considered for use in the dosing system for different functions and this section considers only normally closed valves, as they save battery life and help prevent backflow. In our system, we want to be able to independently run the chlorine detection and phosphate detection tests independently, for that reason, a solenoid valve is used to prevent the flow of liquid from one side to the other.

Two types of uni-port solenoid valves were considered in the tables below, both pilot operated and direct-acting valves, to serve as the primary solenoid valve in **Table 3.8**.

One option is the U.S. Solid 1/4" NPT Electric Solenoid Valve. Record in Table 3.12 is a direct-acting, normally closed, uni-directional solenoid valve with a rated voltage of 12V DC and a current draw of 0.4 A, resulting in a power consumption of 4.8 W. It features a waterproof rating of IP65, making it suitable for outdoor or moisture-prone environments. The inlet and outlet connections are both ¼" FNPT, matching the previous option, and the valve's compact form factor measures 2.25 × 0.8 × 1.25 inches. With strong user ratings on Amazon, it retails for $13.79, offering a balance of affordability and reliability **[61]**.

Table 3.8: Solenoid Valve Summary

Features	AOMG 12V ¼" Solenoid Valve	Beduan 12V ¼" Inlet Solenoid Valve	Adafruit Solenoid Valve
Input Voltage	12V	12V	12 V
Current Load	0.4A	460mA	320 mA
Min. Operating Pressure	0.02Mpa	0.02Mpa	0.02Mpa
Dimensions	2.25 × 0.8 × 1.25 inches	8.46 x 2.36 x 1.34 inches	3.3 x 1.69x 2.24 inches
Cost	$13.79	$9.57	$6.95

3.3.5 Reagent Mixers

To ensure effective mixing of reagents with pool water, two approaches were considered: integrating a static inline mixer or extending the length of the tubing to increase contact time before the mixture is dispensed into the cuvettes.

3.3.5.1 Static Inline Mixer

Static inline mixers are motionless devices that rely on the pressure of the incoming fluid to force liquids through a fixed mixing element housed within a pipe. As the fluid flows through, it is redirected and split several times, resulting in thorough mixing by the time the fluid exits the outlet. While mechanically simple, static mixers do have some drawbacks, primarily the cost. Most commercially available static mixers are made of metal and often exceed $100. Possible workarounds include using commercially available disposable mixers that can be retrofitted into piping or custom 3D-printed designs, allowing for integration in both vertical and horizontal configurations. Another limitation is their need for a minimum flow pressure to operate effectively, which restricts their use in low-pressure sections of the system and requires careful placement to ensure sufficient fluid pressure is available to drive the mixing. Static mixers inherently cause a pressure drop, so when integrated with a pump, this pressure loss must be accounted for, potentially requiring a more powerful pump to ensure the fluid reaches its destination.

An illustration of a typical static mixer setup design is shown below in **Figure 3.3**. In this system, the flow of water drives the mixing process as reagent is injected through a second inlet and exits the system through the outlet as a homogeneous mixture.

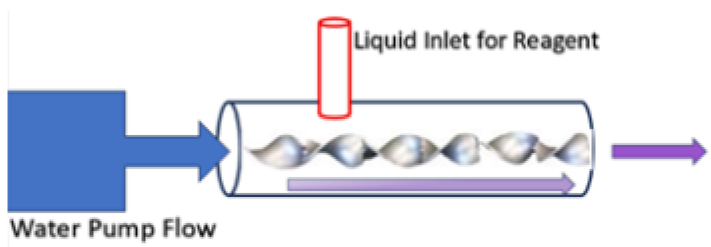


Figure 3.3: Static Inline Mixer Design

3.3.5.2 Static Inline Mixer Selection

One possible option is the Koflo PPT-.25-12-P Polypropylene Tube Disposable Static Mixer, which features a 1/4" outlet tubing diameter and a 1/8" inlet tubing diameter. It has plain tubing ends, allowing compatibility with various mounting configurations, which adds versatility to how it can be integrated into the system. It is also cost-effective, priced at $8.10 for a set of 12 from Cole-Parmer **[62]**.

Another option shown in **Table 3.9** is to purchase just the mixing element itself, such as the Koflo Disposable Static Mixer Element with 12 internal mixing stages. Sold unmounted, this version costs $4.20 on Amazon and may be a better value if a custom pipe casing can be designed to house the element **[63]**.

The final option considered is a PVC static mixer from CBTLSK, retailing for $40 on Amazon for a ¾" socket size. Although primarily designed for gas-liquid mixing, reviews indicate the static mixer performs effectively for liquid applications as well. This mixer was considered due to the lower cost compared to similar products, the transparent mixing chamber that allows visual confirmation of mixing, and ease of installation. The main drawback is the socket size and overall length limits tubing we can use and may require additional design considerations to properly integrate into the system **[64]**.

The main reason for considering inline static mixers is that they require no electrical power to operate, as mixing is driven entirely by fluid flow pressure, utilizing either laminar or turbulent flow regimes. They offer an efficient way to ensure reagents are properly mixed with pool water before being deposited into cuvettes, which aligns with the system's emphasis on smart, low-power design choices.

Table 3.9 Static Inline Mixer Comparison

Features	Koflo PPT Disposable Mixer	Koflo Disposable Mixer	CBTLSK PVC Static Mixer
Dimension (tubing)	1/4" OD, 1/8" ID	0.240” diameter, 3.08 “length, 0.050” end width	¾" socket size
Cost	$8.10	$4.20	$40

In Senior Design II, as testing progressed, we transitioned from using an inline static mixer to evaluating a magnetic stirrer built from a repurposed PC fan. The inline mixers did not provide adequate mixing, and the existing housing design required a compact solution. A small 5V PC fan with the blades removed was selected for testing as a magnetic stirrer and showed promising results for fitting within the system constraints.

3.4 Control Methods and Supporting Components

To manage fluid control components such as pumps and valves, MOSFETs, relays, and motor drivers were evaluated as viable options. These components vary in complexity, and selecting the appropriate one depends not only on the intended application but also on the cost and design constraints of the overall system. This section explores three common control methods for actuating pumps and valves, with the goal of identifying the most suitable approach for powering these components.

3.4.1 Relays

Relays are electromechanical switches used to make or break an electrical connection, to ensure a device has been turned on or off. Relays work by receiving a signal from a low-voltage input source that activates an electromagnet. The electromagnet pulls a switch, completing a higher power circuit on the other side of the relay. This enables a small voltage to safely operate a higher-power load. In practice, relays are widely used as a cost-effective and straightforward solution for controlling actuators that operate at higher voltages or currents than those typically supplied by microcontrollers such as 3.3V or 5V.

Relay, especially opto-isolated relay modules, provide complete electrical separation, protecting the microcontroller from high-voltage spikes or faults on the load side. Relay modules are also typically easier to use, with clear separation between load and control terminals, making them more beginner-friendly for rapid deployment.

3.4.2 MOSFETs

MOSFETs were the second option considered for use as electronically controlled switches for pumps and valves. Unlike relays, MOSFETs allow both simple ON/OFF switching and motor speed control through pulse-width modulation (PWM), making them suitable for more dynamic control applications. They also offer faster switching speeds and lower power consumption.

However, a key disadvantage of using MOSFETs is the lack of electrical isolation between the control (MCU) side and the load side. However their key strength is their small power consumption, and their fast switching speeds that are well-suited for high frequency applications. They are also small and the easiest to integrate compared to the other two control methods.

3.4.3 Motor Drivers

A motor driver was also considered as an alternative to relay-based switching, particularly when more precise and programmable control of a pump or valve is required. Motor control can be achieved through simple pulse-width modulation (PWM) signals generated by a microcontroller, which the motor driver interprets and converts into appropriate voltage levels to drive the motor with precision. Internally, motor drivers commonly use transistors, commonly MOSFETs or bipolar junction transistors (BJTs), as the switching elements to control current flow through the motor. Compared to relays, which only provide binary ON/OFF switching and have a slower mechanical response, solid-state motor drivers are faster, quieter, and allow more dynamic control with less power loss and no mechanical wear. However, their main drawback is the increased complexity in setup, requiring advanced control logic, such as an H-bridge configuration to control spin direction, and potentially more effort in troubleshooting.

3.4.4 Control Method Selection

To simplify control and accelerate project development, electromagnetic relays are used to operate the pumps and valves. Electromechanical relays were chosen because they are cheaper and reliably prevent current leakage when turned off, if opto-isolated relays are chosen.

There are many different types of relays with the two types of relays, electrochemical relays (EMR) and solid-state relays (SSR), is considered here. While both serve the same basic function, switching circuits, they operate differently. EMRs use a physical coil and mechanical components to open or close contacts, while SSRs use semiconductor components and are activated via control pins without moving parts.

3.4.1.1 Electromechanical Relays (EMR)

Electromechanical relays are composed of three main parts: the electromagnetic coil, the contacts, and a mechanical movable part called the armature. When the coil is energized, the magnetic field pulls the armature, causing it to move from the Normally Closed (NC) contact to the Normally Open (NO) contact. When the coil is de-energized, the armature returns to its default position, reconnecting with the NC contact. When contact is made, a "click" sound is heard. The contacts can come in different configurations based on the number of throws and poles in the relay: Single Pole Single Throw (SPST), Double Pole Single Throw (DPST), or Double Pole Double Throw (DPDT).

SPST (Single Pole Single Throw) is the simplest relay configuration, consisting of one Normally Open (NO) contact and one Common (COM) terminal to either make or break a single circuit. DPST (Double Pole Single Throw) relays have two independent sets of contacts that operate simultaneously, allowing control of two circuits with both switches opening or closing at the same time. DPDT (Double Pole Double Throw) relays contain two sets of contacts, with each set having a double-throw function. This means each pole can connect to one of two output paths, allowing flexible switching between two circuits for each pole.

An EMR response time is slower than a SSR, larger, nosier, require higher input power to operate and are sensitive to vibrations and shocks. Their main advantage lies in thermal performance and in their ability to support a broad range of electrical signals and loads, including both AC and DC at various voltages, currents, and frequencies. This flexibility makes EMRs easier to match to specific system requirements.

3.4.1.2 Solid-State Relay (SSR)

Solid-state relays (SSRs) have no moving parts and rely on semiconductor technology to switch devices on and off. They typically consist of three main components: an LED (on the input side), a light-sensitive detector, and a switching device such as a MOSFET, transistor, or in some designs, a phototransistor, which integrates the light-sensing and switching functions. When a control signal is applied to the input, it activates the internal LED. The emitted light is detected by the light sensor on the output side, triggering the switching device and allowing current to flow through the load.

Since there are no moving mechanical parts, SSRs are much quieter than EMRs. SSRs operate at high speed, are smaller, consume less power, operate over a wider range of input voltage, and have a longer lifespan since there are no physical contacts that can burn out. Their main disadvantage is that there is a voltage drop across the semiconductor which causes a lot of power to be lost in the form of heat.

3.4.1.3 Relay Summary

Table 3.10: Relay Comparison Summary

Features	**Electromechanical (EMR)**	**Solid-State (SSR)**
Switching Mechanism	Mechanical Contact +coil	Semiconductor
Response Time	Slower	Fast
Moving Parts	Yes	No
Power Consumption	High(coil)	Lower (optical trigger)
Electrical Noise	Higher; mechanical clicking	Lower; silent operation
Lifespan	Shorter; mechanical wear	Longer; no moving parts
Thermal Performance	Better Heat Dissipation	Generates More Heat
Load/Signal Compatibility	Supports wide AC/DC ranges	Limited; susceptible to voltage surges
Vibration/Shock Resistance	Sensitive; affected by mechanical stress	Excellent; vibration-resistant
Package Size	Larger	Smaller
Cost	Generally lower	Generally Higher

Based on the summary provided by **Table 3.10**, electrochemical relays were chosen because they provide a good all-around solution despite limitations in switching speed, mechanical lifetime, and package size. EMRs are often easier to design and operate, cheaper, and their thermal heat performance makes them the most practical option for our system.

One possible relay option is the 2 piece 2-Channel Relay Module by HiLetgo. This relay operates at 12 V, and each channel requires a trigger current of approximately 4 mA to be activated. Each relay can switch loads of up to 30 V DC at 10 A, supporting a maximum

DC load of 300 W per channel. Designed for durability, the module features diode current protection for fast, reliable response. With a compact overall size of 50 mm × 26 mm × 18.5 mm and a low cost of $8.59 for two pieces, this relay offers an efficient and budget-friendly solution for applications requiring multiple relay channels **[68]**.

Another option is the 5V Relay Module – 1-Channel Relay Switch Board with Optocoupler Isolation and High/Low-Level Trigger from Aediko. This module was selected for its low voltage control requirement, making it compatible with 3.3V and 5V logic signals. Its compact form also makes it suitable for systems with limited space, especially when not all relays need to be joined on a single board **[69].**

The HiLetgo 2-Channel Relay Module was selected because it works as a high level trigger with 12V to drive the relay coil, making it directly compatible with the ESP32's GPIO pins.**[70].**

Table: 3.11: Relay Voltage Comparison

Features	**HiLetgo 12V**	**Aediko 5V**	**Aediko 3.3V**
Operating Voltage	12V	5V	3.3V
Optocoupler Isolation ?	yes	yes	yes
Num. Channels per Board	2	1	1
Max Load per Channel	30V at 10 Amps	30 V at 10 Amps	10 A max

This relay module was chosen for its flexibility and ease of integration. It supports both high and low level triggering, making it compatible with 3.3V and 5V logic signals, which is ideal for systems using microcontrollers like the ESP32. Although it requires a 12V supply to power the relay coil, it can still be triggered by a 3.3V GPIO pin, avoiding the need for additional control circuitry. The optocoupler provides electrical isolation that helps protect the microcontroller from voltage spikes on the load side. Its compact size and single channel design make it a good fit for applications where relays do not need to be combined on a single board.

3.4.5 Flyback Diodes

Any component that uses a coil of wire, such as a relay, solenoid, or motor, acts as an inductive load. Flyback diodes are used to protect circuit components connected to these inductive loads by safely dissipating the back EMF generated when the current is suddenly interrupted (turned off). The diode is placed in reverse bias across the inductive load and is inactive under normal conditions. Once the load is turned off, then the diode becomes forward-biased and provides a low resistance path for the current to flow, preventing voltage spikes and arcing that could damage other components. The main advantage of using a flyback diode is that it protects sensitive circuitry and helps extend the lifespan of switching components.

In selecting an appropriate flyback diode, the following key criteria must be considered: reverse voltage rating, forward current rating, and switching speed. The voltage rating of the diode must surpass the maximum voltage present in the circuit to prevent breakdown. The current rating should be capable of handling the peak current generated by the inductive load **[69]**. Since relay modules were chosen, which often come with flyback diodes on the board, doesn't mean that a flyback diode still isn't necessary, especially across the load where arcing can still happen.

3.5 Chlorine Detection System

3.5.1 Chlorine Molecule Chemical Properties and Safety

This unique compound gas has found its usefulness in human history as a monumental achievement in scientific discovery and being used for military needs **[71]**. While being the second lightest out of the halogen elements, it can also be found in 0.017 percent of Earth's crust.

Chlorine is known for its high electronegativity and prominent electron affinity. This essentially means the amount of electrons orbiting around the atom are more likely to form an unbreakable bond with neighboring atoms. If the individual particle is found naturally in its neutral state an electron causes some of the energy to be dispersed once the molecule becomes an ion.

An ionic compound mainly consists of a cation and anion. Some general considerations to take note of is that a cation is positively charged. While the latter is negatively charged. These are critical in determining how well some metals, nonmetals, or semimetals react with Chlorine.

Table 3.12: Chemical Properties of Chlorine

Chemical Properties	
Electron Configuration	$1s^2 2s^2 2p^6 3s^2 3p^5$
Chemical Formula	Cl_2
Atomic Number	17
Atomic Weight	35.446 amu - 35.457 amu
Oxidation States	-1, +1, +3, +5, +7
Density (1 atm, 0 ℃ or 32 ℉)	3.214 g/liter (0.429 ounce/gallon)

Ionization Potential	11.48 eV
Color	Greenish-Yellow

After the Chlorine reacts, some of the oxidation states mentioned in **Table 3.12** except -1 are represented in hypochlorite (ClO^-), chlorite (ClO^-_2), chlorate (ClO^-_3), and perchlorate (ClO^-_4) which forms strong ionic bonds to remain in its stable state **[71]**. Whereas oxides such as chlorine monoxide (Cl_2O), chlorine dioxide (ClO_2), chlorine perchlorate (Cl_2O_4), dichlorine hexoxide (Cl_2O_6, and dichlorine heptoxide (Cl_2O_7) are very unstable and can easily react with other members.

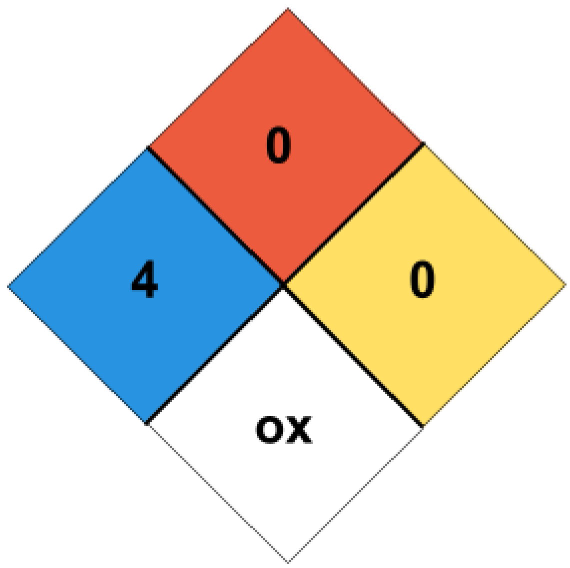


Figure 3.4: NFPA Diamond for Chlorine

Table 3.13: National Fire Protection Association

Hazard	Value	Description
Health	4	Hazardous to inhale.
Flammability	0	Will not combust easily.
Instability	0	Very stable.
Special	OX	Can react with Oxidizing reagents.

As some of the hazards illustrated in **Table 3.13 and Fig. 3.4** contains about less than one part per million is considered to be a non-lethal amount of Chlorine to inhale **[72]** .

One way to express the stoichiometry of Chlorine when combined to be dissolved in water can be shown in **Equation 3.7**.

$$Cl_2 + H_2O \Leftrightarrow HClO + HCl$$ (**Equation 3.1**: Chlorine Dissolved in Water)

The final product of adding the reactants is called Chlorine water since the hydrate itself disproportionately oxidizes and reduces to form an acid. This mostly affects the Chlorine water capability to act as an oxidizing agent because of the hypochlorous acid, which over time exponentially decomposes.

3.5.2 N,N- diethyl-p-phenylenediamine (DPD) Reagent and Chemical Safety

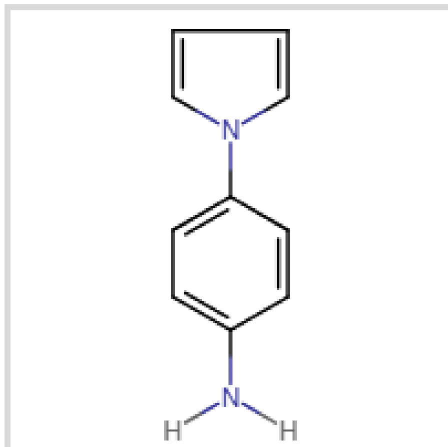


Figure 3.5: Molecular Structure of DPD

Table 3.14: Compound Properties of N,N- diethyl-p-phenylenediamine (DPD)

Chemical Properties	
Chemical Formula	$C_{10}H_{16}N_2$
Melting Point	79-82 ℉
Atomic Weight	164.25 g/mol
Solubility	<1 mg/mL at 66℉, very soluble in benzene.
Density	0.998 g/mL
Vapor Pressure	0.00298 mmHg
Color	Reddish brown or black liquid.

Some of the chemical properties mentioned in **Table 3.14** and **Fig. 3.5** includes the visible color to be Reddish brown or black liquid **[73]**. This is under the presumption that DPD doesn't undergo any chemical changes when placed in an inert atmosphere. With a vapor pressure that is relatively small compared with water, N,N-diethyl-p-phenylenediamine does not require a large amount of pressure to any surrounding surface that would interact with the molecule. Since the DPD density is also less dense than water, N,N- diethyl-p-phenylenediamine floats if placed into any type of container.

Additionally, when a small amount of DPD is used in a controlled environment there is not much risk involved. However, there are an abundance of accounts where someone was exposed to large amounts of DPD, and they would experience skin irritation, dermatitis, and hemorrhaging **[74]**. If placed into air or water, DPD does not induce a reaction to occur due to its low dissolve rate. If this compound is heated over time to decompose it emits toxic fumes. When DPD is introduced with an acid and it releases energy, after it reacts it precipitates salt and water.

3.5.3 Free Chlorine Reaction Time

In this current design, we must know and understand the reaction between chlorine and the N,N- diethyl-p-phenylenediamine (DPD) reagent **[75]**. Typical Amines consist of nitrogen, hydrogen, and an Ethyl group all bonded to a benzene. When chlorine is added to water the final product formed is called hypochlorous acid (HOCl), which can also be further broken down into a hypochlorite ion. This is synonymous to free Chlorine found in an aqueous solution.

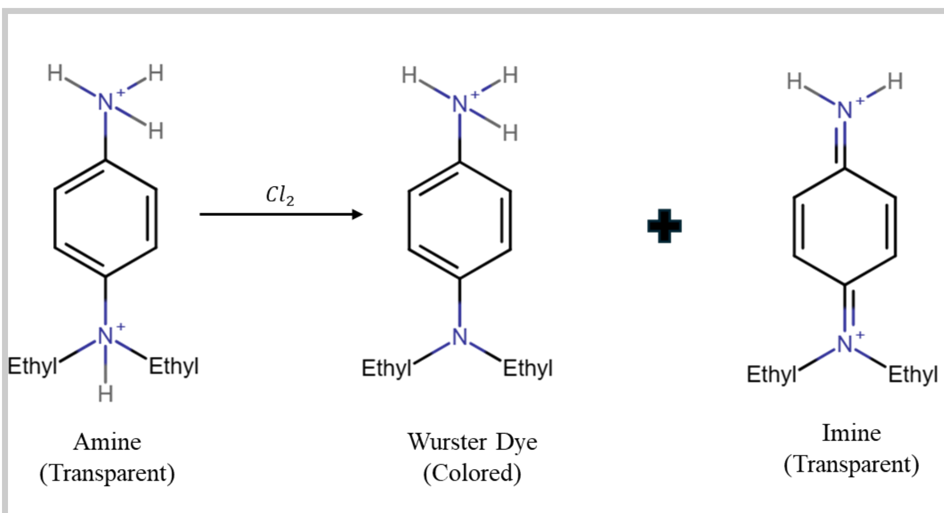


Figure 3.6: DPD Chlorine reaction **[76]**

With Free Chlorine already present, the DPD Amine introduced in **Figure 3.6** is added to the water and would immediately become oxidized **[75]**. Correspondingly, the Wurster dye pH level is neutral, which means it would appear as a mix of pink-magenta color. Whereas the Imine would only be able to form at higher oxidant ranks, which makes the overall stability to be very poor. This would mean a higher chlorine concentration would result in the Imine to be present in the final product, otherwise it would not be taken into consideration when collecting measurements.

Once the DPD reagent is added in with the water, there is a 30-second window to colorimetrically observe the Free Chlorine. Since the reaction is mostly pH-dependent, having a phosphate buffer causes the final product to be a neutral acid and would not be affected by the pH, which is 6-7 in water. This decreases the amount of time it takes to collect measurements and prevent any over or under oxidation from other elements present in the water.

DPD reagent has been used as a valuable detector for different oxidants in water treatment, such as hydrochloric acid, hydrogen peroxide, and persulfate **[78]**. If DPD is oxidized the appearance becomes a clear red color. The molar absorption coefficient once it stabilizes is typically 21,000 $\pm$270$M^{-1}cm^{-1}$ at 551 nm and 19,930 $\pm$400$M^{-1}cm^{-1}$ at 510 nm **[78]**. This falls into the wavelength range that's being used for this current detection and measurement system, so determining the range of values reduces systematic error and eliminates any large amount of uncertainty.

3.5.4 Concentration Test Hardware

3.5.4.1 Cuvette Comparison
After searching for various cuvettes here are some options and reasons why the UV quartz cuvettes were chosen for the final design.

Table 3.15: Various Cuvettes Comparison

Cuvette Comparison Chart					
Item Name	**Cost**	**Total Volume (mL)**	**Spectral Range (nm)**	**Transmittance**	**In stock?**
UV Quartz Cuvettes (2pcs)	$94.99	3.5	190-2500	~90%	✔
Standard Glass Cuvettes (4pcs)	$36.99	3.5	340-2500	~84%	✔
Polystyrene Cuvette (pack of 100)	$26.80	1.5-3	300-1100	~80%	✔

The list of items labeled in **Table 3.15** indicate a wide selection of various cuvettes to choose from. However, the UV Quartz Cuvette has by far the best choice due to its spectral range total transmittance. The values that were used for transmittance were included in a specifications sheet for the quartz, glass, and polystyrene cuvettes. Although the total volumes and spectral ranges would work with 515 nm and 880 nm, the most optimal option that can hold more liquid would make detecting the Free Chlorine much easier.

3.5.4.2 LED Selection

To enable the ability to detect Free Chlorine with DPD reagent in pool water the correct Light Emitting Diode (LED) should be selected accordingly. For accurate measurements selecting a 515 nm LED would fit best with the system, because of the absorbance peak after the DPD reacts with the Chlorine in the pool water. We can determine the concentration of free chlorine based on the intensity diminishing as the light from the LED passes through the UV Quartz Cuvette.

There are alternate mounting styles associated with LEDs. In the case where a diode's mounting style is classified as SMT (Surface Mount Technology) or SMD (Surface Mount Device) usually means the anode and cathode legs are configured for a PCB (Printed Circuit Board). The advantages of using this threading compared to through-hole is its compact form factor would not overlap with any additional components.

Although, one disadvantage of using SMD or SMT is soldering the device onto the board would be very difficult. There is a large risk of accidentally missing the copper pad and damaging the epoxy. If someone were to use a through-hole mounting style they would require less precise soldering but would need to fit the LED onto some 3-D printed mount or keep it suspended to avoid any metal contacts.

Table 3.16: Comparison Across Various LEDs at 515 nm Wavelength

Light Emitting Diode (LED) Comparison Chart				
Part No.	**Cost**	**Peak Wavelength (nm)**	**Output Power (mW)**	**In stock?**
WP7113ZGC	$9.68	515	3.23	✔
XSDG43MB	$8.35	515	2.47	✔
2231507	$15.11	515	2.76	✔

Further explanation of the highlighted portion made in **Table 3.16** was decided based on the mounting style that would best suit the design requirements for our project. The mounting style for the second and third option both have through-hole legs and are attached to an encapsulated shell. Primarily, it was significant that all LEDs were operated at max DC forward current to emit 515 nm wavelength. There was no issue in discovering these components, but is not common to find on popular optoelectronic vendor websites. Even though the second option is less expensive compared to the third option, the LED was considered because the mounting style would still work on a PCB.

All of the components are in-stock, but the second option needs a written inquiry to purchase the desired diode. Since the dominant wavelength was 515 nm at a peak forward current these diodes were going to be used in the Chlorine Detection System. While the power consumption is an important factor in the design, it is relatively small compared to other devices in the system that draw much more energy. There are ratings that provide accurate temperature and time durations for using lead soldering compared to other soldering methods in the specifications sheet. What makes the LED's contrast are their respective viewing angles, which is also known as the lambertian emission pattern.

3.5.4.3 Photodiode Selection

When deciding which photodiode to use for this system, one must have two requirements: a high responsivity at a desired wavelength, and the Noise Equivalent Power (NEP) to be much less than the signal-to-noise-ratio (SNR) of one. In the scope of this project, having an affordable efficient photodiode increases the likelihood of an accurate concentration measurement that was defined in a previous section.

There are two modes the photodiode can operate in. One is the photovoltaic mode which doesn't require an applied bias and the device acts as a battery. The latter uses a reverse bias to decrease the junction capacitance that allows for rapid optical detection. An external current called dark current is present in the circuit. Current across the diode ($I_{0)}$ can be expressed as:

$$I_d = I_0 \cdot (e^{\frac{V_j}{nV_T}} - 1)$$

(**Equation 3.2**: Ideal Diode Formula)

Where I_0 is the dark current, V_j is voltage produced by the diode current as it flows across the junction, V_T is threshold voltage that is dependent on temperature, and *n* is the ideality factor that indicates the net current that is present throughout the photosensor.

[78] There are also many different materials used to fabricate the semiconductor layers that are surrounded by metal contacts. Some examples of these include germanium, black silicon, silicon, gallium phosphide, and Indium Gallium Arsenide. The clear difference between silicon and black silicon is one has a darker appearance due to the small etched patterns that are on the surface. It is far too simple to say the color black comes from the hue of silicon, but rather a grey spectrum which sometimes appears to be saturated.

One advantage of black silicon is the surface area is much larger which allows more light incident on the surface to become trapped. The material properties of germanium is that the absorption coefficient is much larger compared to silicon, which increases the cost exponentially.

Table 3.17: Quantitative Photodiode Comparison

Photodiodes Comparison Chart				
Part No.	**Total Cost**	**Responsivity (@ 515 nm)**	**NEP ($W/Hz^{1/2}$)**	**In stock?**
S1223-01	$28.70	0.32-0.34	$1.3 \cdot 10^{-14}$	✔
FDS1010	$86.18	0.20-0.22	$2.1 \cdot 10^{-13}$	✔
S5973	$29.63	0.26-0.28	$1.1 \cdot 10^{-15}$	✔

What makes the **S1123-01** the optimal choice in **Table 3.17** for this project was the PIN photodiode's respective responsivity and total cost. Some other elements that are useful about the optical response is the receiving bandwidth associated with the diode. To know the minimum power that could be just detected above the noise level the following expression is written as:

$$P_{min} = NEP \cdot \sqrt{BW}$$ (**Equation 3.3**: Minimum Optical Power)

So, the minimum optical power (P_{min}), Noise Equivalent Power (NEP), and bandwidth (BW) was found using the specifications sheet provided by each respective PIN photodiode. Observing the operating temperature is very important, this photodiode has an accurate optical response at room temperature. Any deviation from this temperature ultimately diminishes or changes the optical and electrical response of the device.

The manufacturer of the photodiode selected is called Hamamatsu Photonics, which are known for their high-quality optoelectronic devices. There were other versions that are available and they were compared with the selected photodiode. This was done to see how realistic and compatible the device would be integrated into the final system. Although the various series would slightly contrast to one another, there was a distinct difference in overall performance at the 515 nm wavelength.

Lastly, the overall size of the effective photosensitive area was considered in this selection. Since the overall size of the system must be minimized to prevent any electrical, mechanical, or optical components from overlapping each other. The tradeoff between the bandwidth and effective photosensitive area would affect how fast the incoming signal can be detected as well the amount of light the device can collect. For this particular selection one would want to accumulate large amounts of light, but also at a relatively lower processing speed. As long as the power can be detected, then the baud rate can be further optimized using individual components on an external circuit.

3.5.4.4 Optical Lenses Selection

When selecting the best lens component for this project the divergence angle of the LED was used. This would provide what would fit the beam waist and would not be too small for the beam to pass through. Also knowing the exact surface area of the photodiode had provided valuable information which helped influence the chosen component.

Table 3.18: Quantitative Collimating Lens Comparison

Unmounted Plano-Convex Collimating Lens Comparison Chart						
Part No.	**Total Cost**	**Material**	**AR Coatings**	**Focal length (mm)**	**Optic Diameter (in)**	**In stock?**
LA1289-AB	$41.11	N-BK7	AB	30	0.5	✔
LA1608-AB	$66.50	N-BK7	AB	75	1	✔
LA1422-AB	$70.26	N-BK7	AB	40	1	✔

The **LA1289-AB** lens in **Table 3.18** was chosen for many significant reasons. Some of which are not shown in the table were heavily taken into consideration after a thorough search was conducted. The Anti-Reflection (AR) coatings contain the exact wavelength range but "AB" covers more of the mid-infrared region. Whereas the "A" coating only provides optimal optical throughput for this system. Also, the cost of the "A" coated lenses were much cheaper compared to the "AB" coated lenses.

Furthermore, if one wanted to use a shorter focal length the cost would also increase. It was chosen to compare with the other "AB" coating with a much longer focal length. What makes this specific lens fall short is the small focal length would greatly decrease the beam diameter. This would be advantageous as it would allow the beam to pass through the two interfaces of the cuvette without spilling outside the edges. However, the amount of collimated light that passes through would not get absorbed by the analyte which in turn greatly impacts the accuracy of the concentration measurements.

After the beam has been collimated and passed through the cuvette it needs to be focused onto the PIN photodiode to output an accurate read-out. The size of the beam once it has been collimated was considered at first, but the next step was to know the effective area on the photodiode to ensure all of the light would be focused into that space.

Table 3.19: Quantitative Focusing Lens Comparison

Plano-Convex Focusing Lens Comparison Chart						
Part No.	**Total Cost**	**Material**	**AR Coatings**	**Focal length (mm)**	**Optic Diameter (in)**	**In stock?**

LA1560-AB	$56.07	N-BK7	AB	25	0.5	✔
LA1560-AB	$61.52	N-BK7	AB	25	0.5	✔
LA1540-AB	$60.25	N-BK7	AB	15	0.5	**X**

The **LA1560-A** component in **Table 3.19** was chosen for many significant reasons. Since all optical components were paid out of pocket there would be no reason to choose the most expensive lens out there. So, the material remained the same as the collimating lens because there is no reason to use a different one for this type of optical design. The reason why the focusing lens needs to be smaller than the collimating lens is because of the specific focal lengths.

Additionally, there is a strong emphasis on maximizing the transmission from a lens in the optical design, one must consider the Anti-Reflection coating that enables this to occur. When considering the best AR coating, the spectral range determines how much of the light transmits through the two glass interfaces. Since the 515 nm wavelength of light coming from the LED was already known, both “A” and “AB” already include this in their respective wavelength ranges. Since there is a strong emphasis on maximizing the transmission from a lens in the optical design, one must consider the coating that enables this to occur.

3.5.5 Optical Filters

The Optical Narrow Bandpass Filter function in the Chlorine Detection system is to eliminate unnecessary wavelengths and improve the overall transmitted signal coming from the cuvette. There are a vast array of filters that exist for electronic and optical applications. Some include: low pass, high pass, band pass, band reject, etc. What a low pass filter does is block out all higher frequencies at specific cut-off frequency and allows any other frequencies below the pass through.

The reverse process of the low pass filter would be a high pass filter which attenuates signals with lower frequencies. If one were to combine the low pass and high pass filter a frequency band would only be allowed to pass through. By adjusting the individual electronic components such as the capacitor, resistor, inductor values reduce or expand the cutoff the range of frequencies. These specific filters can also be used to tune frequencies in the RF band but also the optical regime as well.

Here the review of a collection of filters that could potentially be used in the Chlorine Detection system. Dichroic Mirrors, Optical Bandpass Filters, Optical Narrowband Filters have similar features but are used for different applications. Dichroic mirrors are known for their individual characteristics such as reflecting or transmitting a singular wavelength.

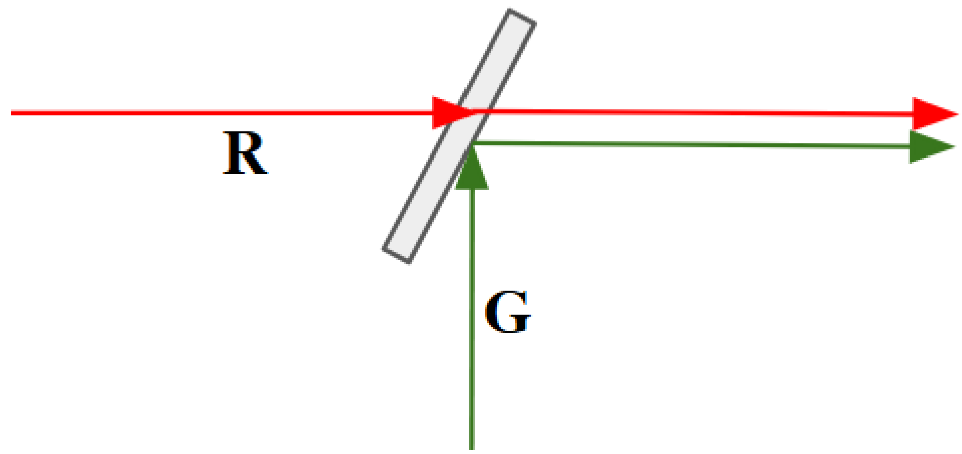


Figure 3.7: Dichroic Mirror Illustration

If someone were to want to use a Dichroic Mirror it would be to selectively choose what wavelengths are reflected while the other is transmitted. How this is accomplished is by applying a thin coating onto the surface of the mirror so the Multi-Layer Interference constructively interferes at only a half wavelength assuming the thickness is a quarter wave. What is demonstrated in **Figure 3.7** is green wavelengths reflecting off the surface of the mirror while the red wavelengths do not reflect, but rather only pass through the mirror.

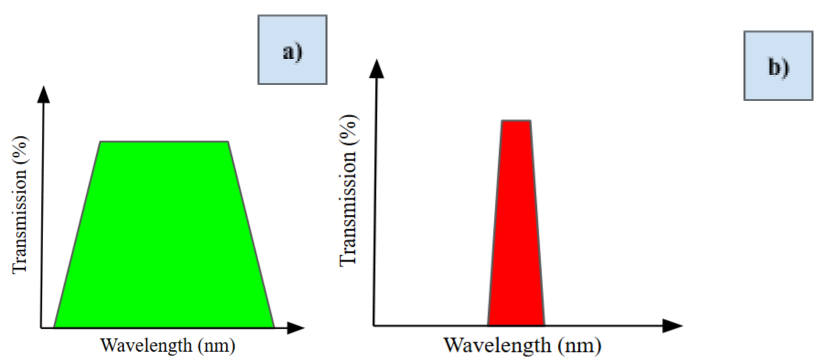


Figure 3.8: **a)** Optical Bandpass Filter and **b)** Optical Narrow Bandpass Filter

What makes the electrical and optical filters differentiate from one another are frequencies or wavelength ranges that are being cancelled out. Even though both are proportional to each other with the speed of light, they both refer to a different frequency regime. Since in all applications longer wavelength ranges meaning relatively smaller frequencies (kHz-GHz range) are very useful in Radio Frequency (RF) applications. While the Optical frequencies are typically found in the MHz-ZHz range.

Furthermore, **Figure 3.8 a)** and **b)** have noticeable passband widths that contrast to one another. While one lets in a very large amount of wavelengths pass through the filter, the latter does not allow many wavelengths to transmit through. In the context the wavelength range is in the visible spectrum, this filter would be very useful for those who want to single out a particular wavelength to eliminate any other parasitic wavelengths that exist outside the desired range.

Table 3.20: Optical Narrow Bandpass Filter Candidates

Optical Narrow Bandpass Filters Comparison Chart				
Part No.	**Total Cost**	**Central Wavelength**	**FWHM**	**In stock?**
SO3010225	$80.00	515 nm	20 nm	✔
102386637	$122.50	515 nm	10 nm	✔
#65-698	$193.31	515 nm	10 nm	✔

Why the **SO3010225** Narrow Bandpass filter in **Table 3.20** would be the optimal choice for this design due to its cost, shape factor, and Full-Width at Half-Max (FWHM). What this means is the filter can be placed in a tight compact area where the collimated beam that passes through the cuvette without any of the scattered light being allowed to be detected by the Photodiode.

Another reason for having a smaller FWHM is helpful in determining how much of the transmitted light ultimately reaches the detector. This would be significant because if the total concentration of the molecules in the pool water cannot be collected, the filter acts as a sanity check to make sure any of the light coming out of the cuvette is completely absorbed. Since the change in the irradiance being collected on the photodiode is tuned for a high responsivity at that particular wavelength.

3.6 Phosphate Detection System

3.6.1 Phosphorus Molecule and Safety

Before going into phosphorite, phosphorus acid, and phosphate they all derive from their ancestral elemental Phosphorus **[79]**. This mass of the element in nature has a mass that is around 0.10 weight percent which is located in the Earth's crust. There are about over 550 different minerals that have been discovered to have Phosphorus in their chemical composition. However, Phosphorus is mostly found in apatite series that are mixed together with calcium ions and have some slight traces of phosphorus include: fluoride, chloride, or hydroxide ions.

Another useful structure of phosphorus is phosphorite, which is also known as a sedimentary rock, and its material properties are very similar to each other **[79]**.

Researchers have reported a large amount of phosphite rocks that can either be found in large and small quantities in various natural ores. If someone were to crush this rock, and mix it with an acid the following formation would be named phosphorus acid. This forms unrefined calcium hydrogen phosphates, which are soluble in water and can be used for fertilizer or food supply for animals.

Table 3.21: Element Properties of Phosphorus

Chemical Properties	
Electron Configuration	$1s^2 2s^2 2p^6 3s^2 3p^3$
Chemical Formula	P_4
Atomic Number	15
Atomic Weight	30.9738
Oxidation States	-3, +3, +5
Density (white)	1.82 gram/cm3 @ 20℃ (68℉)
Color	White or Yellow

With the element properties mentioned in **Table 3.21** the outer shell orientation is easily comparable to nitrogen **[79]**. This is useful when forming covalent bonds as the outer orbital has an additional lone pair electron. There is also a case where phosphorus can react with other elements based on their respective electronegativity, which shares the same type of behavior with nitrogen.

What makes phosphorus different from nitrogen is its very low electronegativity and also consists of substantial atoms. Treating bodies of water with phosphorus forms orthophosphoric acid, which is also known as phosphoric acid. This includes the production of phosphates where some salts contain a phosphate ion and are used as an abrasive in toothpaste and additives in detergent.

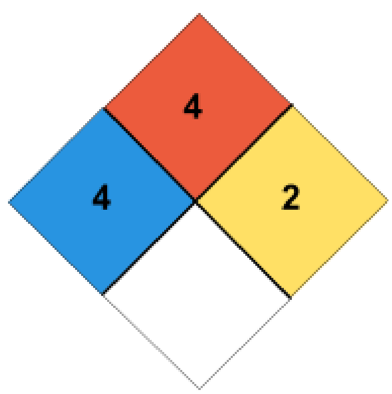


Figure 3.9: NFPA Diamond for Phosphorus

Table 3.22: National Fire Protection Association for Phosphorus

Hazard	Value	Description
Health	4	Severe burns on skin.
Flammability	4	Will combust easily.
Instability	2	Somewhat unstable.
Special	NA	NA

Some key features from **Figure 3.9** and **Table 3.22** include a very high health hazard meaning this chemical requires to be handled very carefully even with Personal Protection Equipment (PPE) **[80]**. If one were to be exposed to that element in a liquid or solid phase it causes severe burns on the skin. Phosphorus is also easily combustible meaning the element is very easy to burn, but can also completely vanish at normal atmospheric pressure and room temperature. Phosphorus is also known to undergo harsh chemical alterations at high temperatures and pressures.

Furthermore, something to keep note about this element is that the element can react commonly with oxidizing agents **[80]**. Phosphorus is also known to be insoluble in water and ethyl alcohol and soluble in carbon disulfide. The chance of white phosphorus reacting with the oxygen in the air is very high, and can be quickly terminated by placing the white phosphorus in water. Proper PPE is needed such as a positive pressure, pressure demand, full facepiece self-contained breathing apparatus (SCBA) or pre-calibrated supplied air respirator with escape SCBA, and a full coverage resistant chemical suit.

3.6.2 Ammonium Molybdate Compound and Safety

[81] One of the compounds that is critical to detect and measure phosphate is made up of Hydrogen, Nitrogen, Oxygen, and Molybdenum. Some other applications of Ammonium Molybdate are in photography, ceramics, and fertilizers. The appearance of Ammonium Molybdate in the solid phase is a white to greenish-yellow. The ratio of the ammonium salt made up of ammonium and molybdate ions is 2:1 and is considered to be a poison.

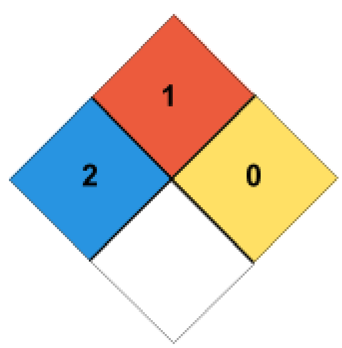


Figure 3.10: NFPA Diamond for Ammonium Molybdate

Table 3.23: National Fire Protection Association

Hazard	Value	Description
Health	2	Can cause irritation of nose and throat.
Flammability	1	Will generate toxic oxides of nitrogen on ignition.
Instability	0	Very stable.
Special	NA	NA

[81] Some of the key features in **Fig. 3.10** and **Table 3.23** show very low values for the various hazards. This compound should not be inhaled or be used without any PPE for proper handling. Even though the Flammability has a very low value it still can be very dangerous to inhale. The compound can also easily break down in water. According to the Safety Data Sheet, the required protective clothing is goggles to prevent any residue on a piece of dust that potentially could land directly on the human eye. The other requirement is rubber gloves to avoid any exposure to the skin after a leakage or spill.

[82] There has been numerous experiments and research completed to ensure that ammonium molybdate and antimony potassium tartrate can react with phosphorus to create an antimony-phosphomolybdate complex. The effect of reducing the complex with an ascorbic acid results in the external appearance to be a blue color. What makes this compound valuable for this system is the ability to only detect orthophosphates. This method is not able to and nothing else is such as polyphosphates or other organic phosphorus compounds.

3.6.3 Chemical Composition of Algae

The mystery behind how these small green creatures grow on various surfaces can be perfectly explained with Science. To begin, when there is a sufficient amount of orthophosphate ions present in the pool water, algae blooms at an exponential rate. To combat this a unique chemical compound with chlorine is used to exterminate the existing microalgae and prevent the potential growth as well.

[83] To better understand microalgae one must consider the individual components that make up the entire organism. There have been scientists and researchers who reported nitrogen and phosphorus make up the chemical composition of this green substance. Microalgae can also consume phosphorus in an alternate form such as polyphosphate or orthophosphate. There is also a strong emphasis on the amount of nutrients provided based on the lipids, proteins, and amino acid content present for fish and shrimps.

Whether different sub-species all share the same amount of nutrients was up to the environmental conditions where the production of algae takes place.

Other professionals who clean and maintain large swimming pools contain a plethora of methodologies to clean and prevent algae growth, but do not have the necessary equipment to detect the potential microalgae growth. That is why in this system where different locations of the swimming pool are tested may contain more orthophosphate than others. This occurs when the pool water is pumped into the cuvette with the already added reagents to form a blue color for colorimetric analysis.

There are numerous accounts researchers have experimentally found the absorption coefficients at various concentrations of molybdenum blue to be around 9,500 - 22,000 $M^{-1}cm^{-1}$ with an absorbance peak at 880 nm wavelength **[84]**. These values were used to determine the alignment of the system after the linear regression analysis.

3.6.4 Concentration Test Hardware

3.6.4.1 LED Selection

To enable the ability to detect orthophosphate in pool water the correct Light Emitting Diode (LED) should be compared to and selected accordingly. For accurate measurements selecting a 880 nm LED would fit best with the system because of the absorbance peak after the pool water. We can determine the concentration of free chlorine based on the intensity diminishing as the light from the LED passes through the UV Quartz window on the Cuvette.

There are alternate mounting styles associated with LEDs. In the case where a diode's mounting style is classified as SMT (Surface Mount Technology) or SMD (Surface Mount Device) usually means the anode and cathode legs are configured for a PCB (Printed Circuit Board). The advantages of using this threading compared to through-hole is its compact form factor won't overlap with any additional components.

Although, one disadvantage of using SMD or SMT is soldering the device onto the board would be very difficult. There is a large risk of accidentally missing the copper pad and damaging the epoxy. If someone were to use a through-hole mounting style they would require less precise soldering but would need to fit the LED onto some 3-D printed mount or keep it suspended to avoid any metal contacts.

Table 3.24: Comparison Across Various LEDs at 880 nm Wavelength

Light Emitting Diode (LED) Comparison Chart				
Part No.	**Cost**	**Dominant Wavelength (nm)**	**Output Power (mW)**	**In stock?**
XTHI30W	$8.59	880	8.83	✔

XTHI12BF	$8.29	880	2.29	✔
XZTHI54W	$8.88	880	6.75	✔

What could make **XTHI30W** the best choice in **Table 3.24** is the cost and output optical power based on chosen specifications for the system. The preferred mounting style to be used in the PCB could be any of the mentioned mounting styles, so this was not considered in the final selection. Now in order to determine the correct optical power the manufacturer specification sheet had a luminous intensity in watts per steradians. So, to find the optical power some assumptions had to be made. At some threshold current the peak wavelength for all components remained at 880 nm wavelength.

3.6.4.2 Photodiode Selection

There is utmost importance in this design to receive the light transmitted through the cuvette and analyze it using a device with a bandgap material. In this system, the necessary parameters were considered to optimize the minimum amount of light from the cuvette that would be needed for further analysis.

When selecting the correct photodiode for this system, the methodology was identical to the selection process used in the chlorine detection system. However, since the LED being used is selected at a different wavelength, the parameters must also change to enhance the optical response from the cuvette.

Table 3.25: Quantitative Photodiode Comparison

Photodiodes Comparison Chart				
Part No.	**Total Cost**	**Responsivity (@ 880 nm)**	**NEP ($W/Hz^{1/2}$)**	**In stock?**
S1223-01	$28.70	0.52-0.54	$1.3 \cdot 10^{-14}$	✔
FD11A	$52.94	0.53-0.54	$6.80 \cdot 10^{-16}$	✔
FDS015	$150.72	0.31-0.32	$8.60 \cdot 10^{-15}$	✔

What makes **BPW34FAS-Z** in **Table 3.25** the optimal choice for this system was the PIN photodiode's respective responsivity and total cost. Some other elements that are useful about the optical response is the receiving bandwidth associated with the diode. To determine the minimum power that could be just detected above the noise level **Equation. 3.8** was used again. The mounting style for the selection photodiode is SMD/SMT where soldering is required on the PCB.

So, the minimum optical power (P_{min}), Noise Equivalent Power (NEP), and bandwidth (BW) was found using the specifications sheet provided by each respective PIN photodiode. Observing the operating temperature is very important, this photodiode has an accurate optical response at elevated temperatures. Any deviation from this temperature ultimately diminishes or changes the optical and electrical response of the device.

Ams OSRAM is the manufacturer of the product, which is known for their high-quality optoelectronic devices. There were other versions that are available and they were compared with the selected photodiode. This was done to see how realistic and compatible the device would be integrated into the final system. Although the various configurations would slightly contrast to one another, there was a distinct difference in overall performance at the 880 nm wavelength.

Lastly, the overall size of the effective photosensitive area was considered in this selection. Since the overall size of the system must be minimized to prevent any electrical, mechanical, or optical components from overlapping each other. The tradeoff between the bandwidth and effective photosensitive area would affect how fast the incoming signal can be detected as well the amount of light the device can collect. For this particular selection one would want to accumulate large amounts of light, but also at a relatively lower processing speed. As long as the power can be detected, then the baud rate can be further optimized using individual components on an external circuit.

3.6.4.3 Optical Lens Comparison

When selecting the best lens component for this project the divergence angle of the LED was used. This would provide what would fit the beam waist and ensure the lens would not be too small for the beam to transmit through.

Table 3.26: Quantitative Collimating Lens Comparison

Unmounted Plano-Convex Collimating Lens Comparison Chart						
Part No.	**Total Cost**	**Material**	**AR Coatings**	**Focal length (mm)**	**Optic Diameter (in)**	**In stock?**
LA1074-B	$58.11	N-BK7	B	20	0.5	✔
LA1608-AB	$66.50	N-BK7	AB	75	1	✔
LA1422-AB	$70.26	N-BK7	AB	40	1	✔

The **LA1074-B** component in **Table 3.26** was chosen for this system after quantitative analysis. Some factors which are not shown in the table were calculated and were compared with other lenses after a thorough search was completed. The Anti-Reflection (AR) coatings contain the exact wavelength within the spectral range but “AB” reflects more light around 880 nm wavelength. Whereas the “B” coating only provides optimal optical throughput for this system. Also, the cost of the “B“ coated lenses were much cheaper compared to the “AB” coated lenses. Since the beam from the LED diverges there was no need to consider the position of where the lens should be located.

Furthermore, if one wanted to use a shorter focal length the cost would also increase. It was chosen to compare with the other "AB" coating with a much longer focal length. What makes this specific lens fall short is the small focal length would greatly decrease the beam diameter. This would be advantageous as it would allow the beam to pass through the two interfaces of the cuvette without spilling outside the edges. However, the amount of collimated light that passes through won't get absorbed by the analyte which in turn greatly impacts the accuracy of the concentration measurements.

After the beam has been collimated and passed through the cuvette it needs to be focused onto the PIN photodiode to output an accurate read-out. The size of the beam once it has been collimated was considered at first, but the next step was to know the effective area on the photodiode to ensure all of the light would be focused into that space. Knowing this information would ensure theoretically the beam waist was not larger than the actual diameter of the lens.

Table 3.27: Quantitative Focusing Lens Comparison

Unmounted Plano-Convex Focusing Lens Comparison Chart						
Part No.	**Total Cost**	**Material**	**AR Coatings**	**Focal length (mm)**	**Optic Diameter (in)**	**In stock?**
LA1560-B	$56.07	N-BK7	B	25	0.5	✔
LA1560-AB	$61.52	N-BK7	AB	25	0.5	✔
LA1540-AB	$60.25	N-BK7	AB	15	0.5	**X**

The **LA1560-B** component in **Table 3.27** was chosen for various reasons. Since all optical components were paid out of pocket there would be no reason to choose the most expensive lens out there. The specific lens does not come already mounted so this drastically reduces the total cost as well. Selecting the best material remained the same as the collimating lens because there is no reason to use a different one for this type of optical design. The reason why the focusing lens needs to be smaller than the collimating lens is due to the specific focal lengths.

Additionally, there is a strong emphasis on maximizing the transmission from a plano-convex lens in the optical design, one must consider the Anti-Reflection coating that enables this to occur. When considering what AR coating would be the best to use for this optical design is the spectral range. Since the 880 nm wavelength of light coming from the LED was already known, both "A" and "AB" already include this in their respective wavelength ranges. Since there is a strong emphasis on maximizing the transmission from a lens in the optical design, one must consider the coating that enables this to occur.

3.7 Particulate Measurement System
3.7.1 Lens Selection

Since the goal of the measurement system is to image particles of a very small size, selecting the correct lenses is very important. Spherical aberration must be reduced while at the same time maintaining a certain f-number so that the system is not limited by diffraction. The first selection must be the correct type of lens. The three main types that could be used are aspheric lenses, singlets, and achromats.

3.7.1.1 Aspheric Lenses

Aspheric lenses are superior in terms of reducing spherical aberration. This would allow the system to have a very accurate focusing spot which would provide high image quality. Aspheric lenses typically have a smaller f number compared to the other two lenses. This would require a very accurate degree of collimation. A small error could cause high levels of aberration within the system. This would produce a very bad image. Aspherical images are also typically extremely expensive. Since they have a complex structure they require precision polishing, leading to a price point double that of achromats and ten times that of singlets.

3.7.1.2 Singlet Lenses

Singlets are a much simpler type of lens consisting of just a single piece of glass. This glass can be molded into bi convex or plano convex for focusing. Since our design can not be considered to be focusing at an infinite distance away for this project it would be better to utilize biconvex lenses. The simplicity of the singlet also allows for more control over the magnification of the system and its number. Singlets in terms of price are the preferred lens. They can be found around thirty dollars, much cheaper than both other lenses. This also allows you to buy more of them and give more degrees of freedom to your system for better control of the parameters of your system as mentioned before. However, with this price they have considerable drawbacks. Since the price of having lenses with your own specified thickness and radii defined is too expensive, we are limited by what is commonly made by lens manufacturers. This means that the optimization of our system the only thing that can really be changed is the distance between our lenses. This makes it very hard to get the specific numerical aperture, diffraction limitation, magnification, and reduction of spherical aberration from our lens system. This leads to a system that takes blurry images with a very large limitation on the size of the smallest identifiable feature.

3.7.1.3 Achromatic Lenses

The last option for our system is achromatic lenses. Normally designed for multiple wavelengths of light, achromats feature two different types of glass being molded together for a very large reduction in chromatic aberration, referred to as the crown and the flint. Since our system is using only one wavelength to illuminate this is mostly unneeded. However, having this crown and flint allows the lens to have better focusing capabilities than just a singlet, better correction of spherical aberration, not to the degree of the aspheric lens though. The cost of these lenses are also a middle point between the other two lenses.

3.7.1.4 Summary
Looking at **Table 3.28**, the achromatic lenses seem to be the best for this project. Featuring decent focusing capabilities as well as not costing an absurd amount the achromats allow the development of an imaging system with the capabilities for small feature focusing while still being highly modular. Giving us the ability to get the f number and magnification we want while still being able to provide a system that is on the limit of being diffraction limited.

Table 3.28: Comparison of Focusing Lens Types

Aspheric Lenses	**Singlet Lenses**	**Achromatic Lenses**
+ + Aberration - - Price	+ + Price	+ Aberration - Price

3.7.2 Achromatic Front Lens Selection

Now that we have decided that Achromatic Lenses are the best for this project it is time to decide which achromatic lens will be best for this project. In the lens design that comes later, it is seen that the imaging system consists of two separate lenses, the back and front lens. The best effective focal length is determined to be around 40 mm with a 12.7 mm diameter for the front lens, 50 mm with a diameter of 25.4 mm for the back lens so the lenses looked into are those focal lengths.

3.7.2.1 Newport PAC025
The Newport PAC025 is a 12.7mm lens with a 38.1mm focal length. It is coated with MgF_2. At 515 nm it has a reflectance of 1.4% making it perfectly capable of letting enough light through. It is made of N-BaK4 and N-SF10 **[85]**.

3.7.2.2 Thorlabs AC127-050-A
The Thorlabs AC127-050-A is a 12.7mm lens with a 50mm focal length coated for the 350-700 nm range. This gives it a reflectance of 0.2%. Material of N-BK7/SF2 **[86]**.

3.7.2.3 Newport PAC024AR.14
The Newport PAC024AR.14 is a 12.7mm lens with a 38.1mm focal length. It features a 0.3% reflectance at 515nm with an antireflective coating made for 430nm to 700nm. Made out of N-BaF10 and N-SF10 **[87]**.

3.7.2.4 Summary
Looking at **Table 3.29**, based on the testing within Zemax, the Newport PAC025 is the best lens for the job. It has the lowest price point out of all of the lenses and since we are not dealing with a low light level the 1.4% reflectance is ok. It also managed to still provide a high degree of focusing with minimum aberration.

Table 3.29: Comparison of Potential Front Lenses

Newport PAC025	Thorlabs AC127-050-A	Newport PAC024AR.14
- Cost + Aberration - Transmission	+ Cost - - Aberration + Transmission	- - Cost + Aberration + Transmission

3.7.3 Achromatic Back Lens Selection

3.7.3.1 Newport PAC040

The Newport PAC040, similar to the lens chosen from the front lenses, features a MgF_2 coating for minimal reflection. It has a focal length of 50.8mm with a diameter of 25.4mm **[88]**.

3.7.3.2 Thorlabs AC254-050-A

The Thorlabs AC254-050-A is a 50mm focal length lens with a 25.4mm. It features the same coating as the thorlabs lens mentioned in the front lens section making it a good pick for low reflectance **[89]**.

3.7.3.3 Thorlabs AC254-050-AB

The Thorlabs AC254-050-AB has the same focal length and diameter as the AC254-050-A, but with a different coating. The AB coating is specialized to extend the anti reflectance to the near infrared range, it also uses a different flint material **[90]**.

3.7.3.4 Summary

Based off of the Zemax simulations and the resulting aberrations the Newport PAC040 is the best lens for the system. Though it was not the cheapest lens, and it does not have the best transmission, its optical performance was the best out of all of the lenses. See **Table 3.30** for more details.

Table 3.30: Comparison of Potential Rear Lenses

Newport PAC040	Thorlabs AC254-050-A	Thorlabs AC254-050-AB
+ Cost + Aberration - Transmission	- Cost - Aberration + Transmission	- - Cost - Aberration + + Transmission

3.7.4 LED Collimation Lens

For the LED part of the design, it is important that the lens is able to provide perfectly collimated light. If the rays are not perfectly collimated our image will be distorted and we will not be able to accurately determine the size of the particles. It is also important that the diameter of the lens is large enough to provide enough light for our system.

Unlike the lenses needed for imaging, a simple singlet lens is enough for this part of the system.

Thorlabs LA1289
The Thorlabs LA1289 is an uncoated singlet with a 12.7mm diameter and a 30mm focal length. It is a planoconvex lens which allows for high accuracy collimation with minimal aberrations [91].

Thorlabs LB1258
The Thorlabs LB1258 is also an uncoated singlet with a 12.7mm diameter and 30mm focal length. This lens is a biconvex lens which through simulations showed higher degrees of aberration compared to the planoconvex [92].

Thorlabs LA1289-AB
The Thorlabs LA1289-A is the same as the first lens but with an anti reflective coating. This allows for a higher degree of transmission however this comes with an increase in price [93].

Summary
Normally the price would make the uncoated the better choice, however, due to availability the coated is a better choice. Also, with the lower degrees of aberration it is clear that a plano convex lens is the right choice. Therefore the Thorlabs LA1289-AB is the best lens for the job. See table 3.31 for more information

Table 3.31: Comparison of LED Collimation Lenses

Thorlabs LA1289	Thorlabs LB1258	Thorlabs LA1289-AB
+ Cost + Aberration - Transmission	+ Cost - Aberration - Transmission	- Cost + Aberration + Transmission

3.7.5 Camera Selection

The imaging system would not be complete without the sensor it is imaging onto. The sensor is very important as the size of the sensor as well as the size of its individual pixels limit what we are able to capture. The speed at which it captures also plays an important role since how much blur is on each particle limits the accuracy of which our system can provide.

3.7.5.1 CAM-IMX296Mono-GS
The CAM-IMX296Mono-GS is an imaging sensor with a sensor size of 5mm by 3.75mm, a pixel size of 3.45µm by 3.45µm, and a resolution of 1440 by 1080 (1.55 megapixels). It outputs in RAW10 allowing for a wide range of contrast. The camera is capable of an exposure time of 30µs allowing for incredibly fast images, meaning even a quick moving particle would not be out of focus. It also features a global shutter. This

means that it is capable of reading the information from every pixel at one time. Combined with its incredibly fast exposure time the camera significantly reduces the possibility for distortion of particles moving at faster speeds. The camera is made by Innomaker but there is a very similar model by Raspberry Pi which means documentation can be pulled from the Raspberry Pi website for the software. This makes the connection of the two systems very easy. However, there is a drawback to the camera. Since our imaging system ideally has 0.7 magnification, the 10 micron particles appear to be 7 microns, meaning that each particle is only captured by two pixels. This satisfies Nyquist but means that the resolution of these particles is very low **[94]**.

3.7.5.2 Arducam OV9281 1MP Monochrome Global Shutter

The Arducam OV9281 1MP Monochrome Global Shutter Camera is an imaging sensor with a sensor size of 3.84mm by 2.4mm, a pixel size of 3µm by 3µm, and a resolution of 1280 by 800 (1 megapixel). The camera also outputs in RAW10. Also similarly to the previous camera it features a global shutter with a quick exposure time. This camera is made by Arducam which has better documentation than Innomaker which could lead to less frustration and an easier time when connecting it to the Raspberry Pi. It also features a smaller pixel size than the previous camera, which gives our 10 micron particles slightly better resolution than the previous camera. However, one major drawback to this camera is its smaller sensor size. This gives a lower field of view, decreasing the amount of particulate that we can image. This decreases the accuracy of our measurement **[95]**.

3.7.5.3 Arducam 5MP OV5647 Camera Module

The Arducam 5MP OV5647 Camera Module is an imaging sensor with a sensor size of 3.7mm by 2.7mm, a pixel size of 1.4 µm by 1.4 µm, and a resolution of 2592 by 1944 (5 megapixels). This camera varies very differently compared to the other two as it is no longer a global shutter and is not monochrome. This introduces disadvantages like a loss of detail due to the bayer filter. Sensors that shoot in color use this filter so that each pixel is for a specific color, all of the neighboring pixels are then averaged together for each pixel in order to create the final image. This leads to a loss of contrast. Also, the rolling shutter can introduce potential problems as well, if any of the particles are moving quickly, they can potentially move across pixels as they are capturing line by line, the particle would then look larger than it actually is since it is covering more pixels. These disadvantages are slightly offset by the fact that the sensor has a much higher resolution and a smaller pixel size. This allows our system to much more accurately capture the particles on the smaller end of our target range, while not decreasing the sensor size to a large degree **[96]**.

3.7.5.4 Summary

Overall the CAM-IMX296Mono-GS seems to be the best choice. It captures in monochrome which is a must for this system to have the highest accuracy. The sensor size of the sensor is also a lot more practical for the creation of our lens design compared to that of the Arducam OV9281 1MP Monochrome Global Shutter Camera. See **Table 3.32** for more detail.

Table 3.32: Comparison of Imaging Camera

CAM-IMX296Mono-GS	Arducam OV9281 1MP Monochrome Global Shutter Camera	Arducam 5MP OV5647 Camera Module
+ + Sensor Size + Global Shutter + Resolution + Monochrome - Documentation - Cost - Pixel Size	+ Documentation + Pixel Size + Global Shutter + Monochrome - Resolution - Cost - - Sensor Size	+ + Cost + + Resolution + + Pixel Size + Sensor Size + Documentation - - Rolling Shutter - - Color

3.8 Debug Panel

The debug panel is a panel consisting of leds that indicate when a system has failed on the machine. The panel is controlled by a switch that when flipped on lights up LEDs indicating which systems are currently functioning. To persevere power the system does not light up the LEDs unless the debug switch is on.

3.8.1 Potential Designs

Since the system does not light up the LEDs unless the debug switch is on and the system is currently working, the LEDs are controlled with an and logic gate. This logic can be achieved with either BJT transistors, Mosfet transistors, or by using the MCU.

3.8.1.1 BJT

One of the solutions available to create an and gate is to connect two NPN BJT transistors in sequence and input the signals into the base of the transistors, as shown in **Figure 3.11**.

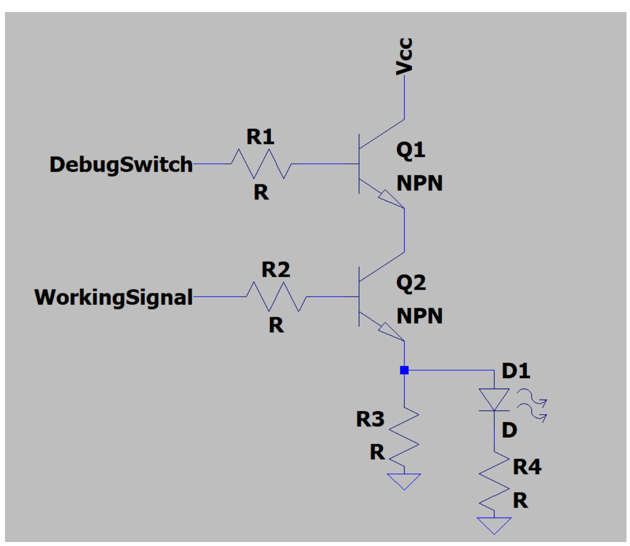


Figure 3.11: BJT And Gate Design

The BJT solution has a component spread of 3 and a total component count per light of 7. This system does not require any pins on the MCU to function as all the logic is handled by the transistors. The power consumed by the system can be controlled by the values used in the resistors as a higher resistance leads to a lower current flow, but also dim the light of the LED. Within this circuit the NPN transistors act as a switch which closes when the logical voltage is applied to the gate terminal. With these two transistors in sequence the current is halted unless both gates are high, which makes the output act as an AND gate.

3.8.1.2 Mosfet

Another potential solution for achieving an and gate behavior is to use mosfet transistors to form a nand gate and a not gate. To form a nand gate we can connect two pmos transistors in parallel then connect them in sequence with two nmos transistors **[40]**. A not gate can be formed with a pmos transistor and a nmos transistor in sequence **[40]**. Combining these circuits we can form the circuit shown in **Figure 3.12**.

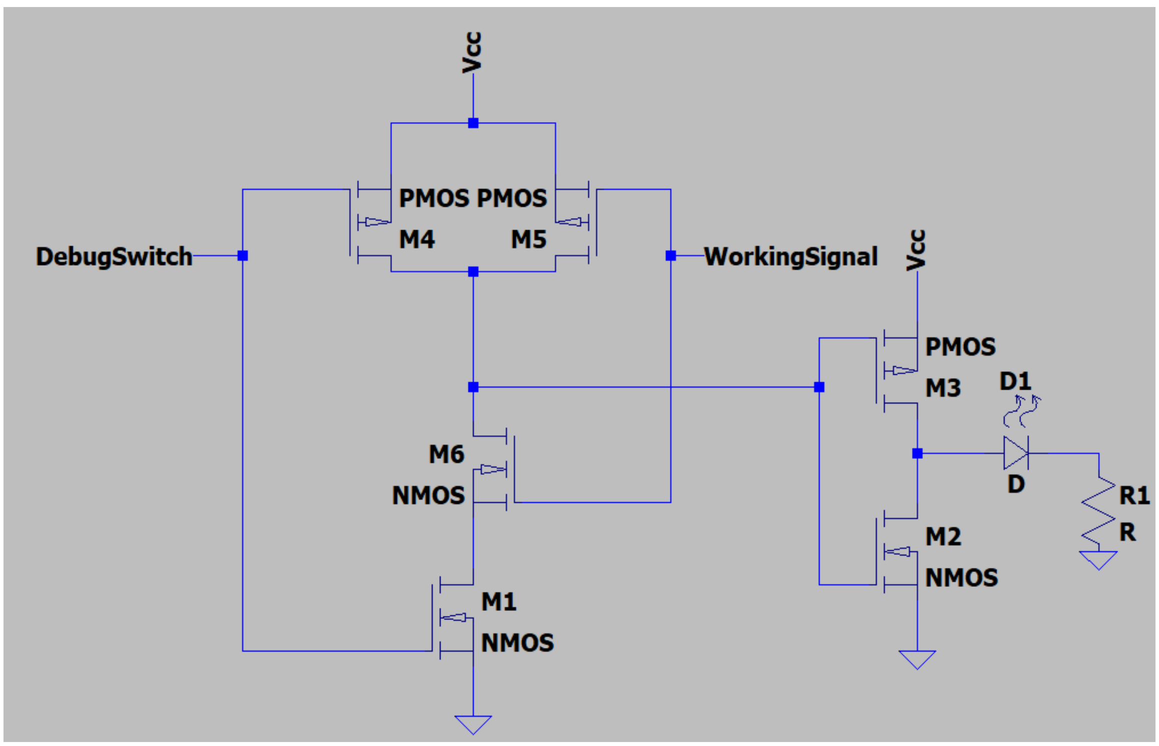


Figure 3.12: Mosfet And Gate Design

This design has a component spread of 4 and a total component cost per light of 8. Likewise to the BJT design this circuit requires no computing component as all the logic is handled by the transistors. Mosfets also tend to be significantly smaller than BJT transistors. The design is slightly more complicated than a BJT design as two logic gates are required to achieve the optimal gate. This design also requires more complicated parts as the BJT design is mostly resistors while this design is mostly mosfet transistors. The power consumed by the system can also be controlled by the values of the resistors.

Within this circuit we can imagine that the nmos transistors act as switches that close when the gate is high, and the pmos transistors act as a switch that closes when the gate is low. We can then split this circuit into 2 components, the left side being a NAND gate and the right side being a NOT gate. When high is applied both of our NMOS on the left are open, which causes the output to route to ground making it around 0V. This then goes to the input of the second gate, which causes the PMOS to close and the NMOS to open, making the voltage across the LED and resistor around Vcc.

3.8.1.3 MCU

The final solution to achieve an and gate behavior is to use the MCU. If we connect the debug switch and the working signal to the MCU, we can then test to see if they are both high and if so we turn on an LED connected to the MCU. An example of this is shown in **Figure 3.13**.

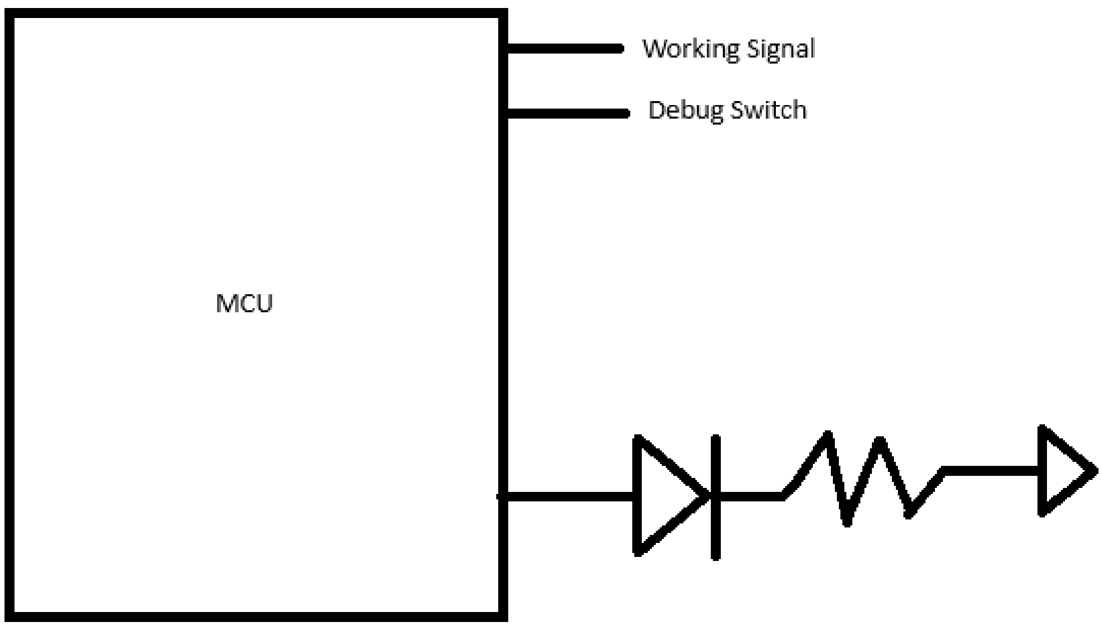


Figure 3.13: MCU And Gate Configuration

This design is the simplest in component spread and count as the design only requires 2 components. This design however, requires computation to check that both signals are on. This design also takes up the most amount of pins on the MCU as one is needed to check the debug switch, and there needs to be 2 to 1 more pins used on the MCU for every LED. This design is also highly dependent on the MCU, which can fail due to other systems failing. This can cause confusion on which system is actually failing as for example if the MCU fails all the lights will be off, which can also indicate a regulator or power failure. We also obtain better control over the LEDs brightness as we can use pulse width modulation to control the brightness.

3.8.2 Summary

Given the potential designs for the debug panel LED circuit, I believe the best option available to us would be to use the BJT design. As seen in **Table 3.33** the MCU design, while having the least amount of components, does take up the most amount of pins on the MCU. We wish to avoid taking up pins that can be used for more computationally heavy tasks. The design also requires us to monitor for a change, which increases the average power consumption. The MCU design also requires the MCU to be functioning, which means when the MCU fails, the debug panel will show a power failure. The mosfet design is also un ideal as it takes up an additional component and has a higher component spread. Since the BJT design requires the least amount of pins and has the second lowest component spread and count, the BJT design is the optimal design for this section.

Table 3.33: Debug LED Circuit Comparison

Design	MCU Pins	Component Spread	Components
BJT	0-1	3	7
MOSFET	0-1	4	8
MCU	1-2	2	2

3.9 Temperature Sensor Selection

In order to get information about the temperature of the pool a sensor must be used. When selecting a sensor it is important to get one that is waterproof, to ensure it works in a wet environment, and one that easily connects to our MCU.

PiShop Waterproof DS18B20
This thermistor uses a 3 to 5 volt supply and provides a digital reading with 9 bit to 12 bit resolution. The device can measure temperatures from -10 to +85 degrees celsius with an accuracy of plus or minus 0.5 degrees celsius. This easily falls into the average temperatures of a pool while providing accurate measurements which is more useful to the consumer. It is also able to communicate using only one pin which is a nice advantage. The packaging it comes in is already waterproof which makes it easier to implement into the design **[120]**.

LM35DZ from Digikey
This thermistor uses a 4 to 30 volt supply and provides an analog reading. This is a slight disadvantage since the signal degrades more as it travels to the MCU. Also, the thermistor requires an analog to digital pin to send its temperature measurements to. The range of measurements is from -55 to 150 degrees celsius with an accuracy of 0.75 degrees celsius. Compared to the DS18B20, this is a much larger safe temperature range, but it is counteracted with a lesser accuracy. Since PoolWatch does not require such drastic temperature ranges this is a negative. The device also does not come with waterproofing so it has to be done manually, the thermistor is cheaper though so we would be able to make more mistakes with it **[121]**.

RTD PT100
This sensor is a resistance temperature detector, it works differently than the thermistors in the way that the resistance from the device must be measured to determine its temperature reading. This forces us to use a temperature to digital converter as a middle man between our MCU and sensor. However, this device provides other positives that help correct this disadvantage. The temperature accuracy from this device is 0.01 degrees celsius with a temperature range of -50 to 550 degrees celsius. This is much larger than either thermistor. The cost of this sensor is around 10 dollars more than the other two, along with the need for a whole other part; this is by far the most expensive option **[122]**.

Summary
Overall the PiShop Waterproof DS18B20 seems to be the best sensor for the job, with its middle of the line accuracy and cost it provides a nice balance between the other two. It also easily connects with our mcu.

Table 3.34: Comparison of Temperature Sensors

Waterproof DS18B20	LM35DZ	RTD PT100
++ Connectivity + Accuracy - Cost	+ Connectivity - Accuracy + + Cost	- - Connectivity + + Accuracy - - Cost

3.9 Full Stack Development

To create a website that supports the operations required to satisfy this project, we used a full stack deployment. To store user information, device settings and test results we needed a database. We also needed a backend web server to handle the requests of both the device and users. Likewise a frontend website is required to allow the user to make requests of the device and to view the results of the collected tests. There are several common stack options that are available to us and were reviewed in this subchapter.

3.9.1 LAMP

A LAMP stack is a full web deployment that takes advantage of Linux, Apache, MySQL and PHP or Python **[10]**. The stack is a bit antiquated as more modern web servers architectures provide more usability. The advantage of using a LAMP stack is that the stack is relatively simple and has plenty of documentation surrounding the stack as it has been around for quite some time. The stack requires a Linux system, which means that without any form of containerization the webserver made in this stack would be limited to running on Linux machines. If we want to provide easy deployment on every machine we would need to create a full on virtual machine to support the stack. Another disadvantage is the rigidity of MySQ, which has issues with handling unstructured data **[10]**. Apache web servers also often run into problems with handling many requests. Another challenge with working with LAMP stacks is that the stack uses two different languages. To run the front end of the stack we would be developing in Javascript, but the backend would be programmed in either PHP or Python.

3.9.2 MERN

A MERN stack is a stack that is made up of Node JS and Express for the backend, React for the frontend and MongoDB for the database **[11]**. The stack provides several advantages over a LAMP, the first of which being consistent in its programming language. While a LAMP stack requires at least two different programming languages to function, MERN stacks are programmed entirely in Javascript. There also exist numerous packages available for Javascript development, which we can take advantage of for this project. MERN stacks allow for simple control over routes on the website, which allows us to easily separate the backend and frontend of our site. A noticeable disadvantage of a MERN stack is that the create react app tool has been deprecated **[12]**.

3.9.3 MENN

Another available stack we can use is a MENN stack, which is the same as a MERN stack, but uses Next.js instead of React. Next provides several advantages over React, the most notable is simplified routing **[13]**. Routing in MENN stacks follows the routing of the folder system that handles the Next app. Another advantage of Next is that you can decide whether the page should be rendered server side or client side. This allows us to provide the end user with a partially loaded site while we render some of the more complicated resources of the page. A disadvantage of MENN stacks is that they are slightly more complicated to program than the average MERN stack.

3.9.4 Summary

Given the available options to create a full stack deployment, I believe that we would be best suited for developing a MENN stack. As seen in **Table 3.34** LAMP stacks are antiquated and lack a lot of the features of modern stacks. LAMP stacks also rely on the Linux OS instead of the programming language, which cause complications when we want to containerize the app. MERN stack, while simple and efficient, are relying on deprecated tools to build them. Since MENN stack uses up to date tools and provides better control in rendering options, a MENN stack would be best for meeting the requirements of this project.

Table 3.35: Stack Comparison

Web Architecture	LAMP	MERN	MENN
In date	Out of fashion	Deprecated tools	Yes
Simple Routing	Yes	no	Yes
Consistent	no	Yes	Yes

Language			
Deployability	Only on Linux	Any computer	Any computer
Server Side Rendering	no	no	yes

3.10 Containerization

When developing a web app, the ability to deploy the app anywhere is essential to creating a professional tool. This is often achieved by creating a resource such as an image or virtual machine that comes with all the necessary dependencies for the program to function as intended. For PoolWatch, we wish for the webserver to be easily installable on any common web environment to allow users to maintain their own web server, as well as making our web server easily expandable. A containerized web server also be more resistant to potential attacks as the programs only have access to items present in the program's container.

3.10.1 Docker

Docker is a common tool used to create container images. These images are used to spin up containers, which are essentially light weight virtual machines. Containers are isolated from the rest of the system like the common virtual machine, but they rely on the host's operating system and thus require no hypervisor **[7]**. All of the dependencies required by the program are backed into the image, which also relies on a base image to build. The base image can be any preexisting image and there exists base images for all common programming languages.

In order to build a Docker image we need to create a dockerfile, which contains the configuration of the image **[117]**. Within the dockerfile we define what base image we are using and what files we want to add to the image. We also define what command the image should run once the image is made into a container. These images are stored locally, but most web hosting services require them to be uploaded to a Docker registry. Docker provides an image for hosting our own registry, and the official Docker website allows users to host one image repository for free. Since we have more than enough space and processing power to handle a registry, we hosted our own.

The main advantage of using Docker is that it is the industry standard. There is a lot more content we can reference when using Docker as opposed to the other options available to us. Docker is also lightweight compared to complete virtualization. Another advantage of Docker is that the dockerfile acts as documentation for everything on the image. The final advantage I can see for Docker is that creating an image is much faster than creating a virtual machine image. Disadvantages for Docker would be the requirement of a base image to build a new image. We do not know what is on the official base images docker provides, and we might need to delve into their documentation if we run into any significant errors. Another disadvantage is the storage of these images. If we want to use a web hosting platform, we need to host our images on a registry, which requires some form of security to prevent others from taking the registry over.

3.10.2 Podman

Podman is another available software for creating container images. Podman works in a similar way to docker, but there are several key differences. Docker uses a server-client relationship for running pods with the Docker daemon, while podman operates on its own **[8]**. Docker only supports running a container as the root user, while podman can run containers as separate users **[8]**. Docker ignores firewall settings, while podman adheres to them **[8]**.

Creating Podman images follows the exact same path as Docker images. A dockerfile needs to be created with the configurations of the image written in that file. Similarly a base image is required to build the new image, and to take advantage of most web hosting solutions, that image needs to be uploaded to a registry. Podman and Docker images are cross compatible so we can use Docker images as the base image and upload Podman images onto Docker registries.

The main advantages that Podman has over Docker is the increased security Podman provides. Podman allows us to block root access to a container as the containers can run as a separate user. Podman is also slightly more lightweight software as it does not rely on any daemon to run. Podman is also native on SELinux OS, such as RHEL, CentOS, and Rocky **[8]**. Podman not bypassing the firewall prevents us from accidentally exposing a port we should leave close. Podman naturally shares the advantages that Docker has such as creating lightweight images and having a large library of base images. Another advantage of Podman is that the CLI features a couple of extra arguments that make debugging and pulling images easier. The disadvantages of Podman are similar to Docker as we need a registry for our images and we need to rely on base images. Another disadvantage is that podman is not as commonly used as Docker, so most documentation we find on the subject is for Docker. Another problem with Podman is that it does not run natively on Windows, which can be problematic for some of our team members.

3.10.3 Virtual Machine

One of the solutions for containerizing the webserver would be to create a virtual machine with all of the required code preinstalled on the machine. There are several ways to go about doing this. The most professional way of achieving this would be to design a configuration as code solution for booting up a virtual machine and installing all the dependencies on the device. Another solution for this would be to use a hypervisor to start a virtual machine and hand install all the dependencies on the device. The outcome of this solution would be a virtual machine image typically in a QCOW2 format.

3.10.3.1 Configuration as Code

There are 3 steps in creating a configuration as code solution for generating QCOW2 images: initialization, configuration, cleanup. In initialization, the program would take an ISO of our image and implant a startup script inside of the ISO, then start a virtual machine using the ISO. Depending on the running PC's processing power and whether or not the ISO is being downloaded or already present on the machine, this step can take from around half an hour to an hour to complete per run. In the next step configuration, the program would install all the necessary packages on the virtual machine and then shut the machine down. Configuration is highly dependent on what is being installed, but if a full update is taking place on the machine, then configuration would likely be at least 30 minutes. In the final step cleanup, the program would compress the produced QCOW2 image and other miscellaneous cleanup tasks. Configuration is the shortest step, depending on what we want for cleanup, it could take 10 minutes.

There are several tools we would need to create a configuration as code solution. The first tool is some form of hypervisor to host the machine while we boot the machine up and configure the machine. The most apt option for a hypervisor would be a kernel based virtual machine as that software would make starting the device much easier than another solution. In my experience QEMU is the best solution as it produces QCOW2 images by default, which are easily translatable to other formats and are supported on common web deployments such as Harvester **[9]**. Another tool we would need is a system deployment tool. This tool would be used to start up the virtual machine and configure the machine with all of our dependencies. The industry standard for this is Ansible, which was originally designed for configuring preexisting machines all at once. However Ansible can be used to run commands on a host machine, which means Ansible can be used to start up a virtual machine. Once the machine has been started up, then Ansible would be used to connect to and configure the machine then shut the machine down. Going back to the host machine, Ansible can once again be used to take care of the cleanup step for the same reasons as mentioned in initialization.

There are several advantages in choosing a configuration as code solution for creating virtual machine images. The most notable one is that every component on the virtual machine is documented in the code. When using this method we can create images that are all the exact same. Configuration as code, however, is much more difficult to implement than any of the other methods. When designing an image generator you need to install everything on the QCOW2 image by command line inputs. This makes configuring and downloading dependencies harder as there would be no helpful gui or guide to walk you through the installation.

3.10.3.2 Hand Install

There are two general steps for creating a virtual machine by hand installing. The first step is to download an ISO and boot up the virtual machine using a hypervisor. In this step you would choose a suitable OS distribution and then follow the booting wizard and install all of the required packages from the installer. The next step would be configuring the virtual machine with the required dependencies and code. During this step you would download the software needed to run the code and configure any environment variables, such as proxy settings, that the code needs to run.

The only piece of software that is needed to create a virtual machine QCOW2 image by hand is a hyper visor. The most appropriate one to use would be QEMU as the hypervisor stores the virtual machine images as QCOW2 images. QEMU also comes with a gui that can be used to create and start the boot of the virtual machine.

The advantage of hand installing a virtual image is that the process is much easier to establish than a configuration as code solution. By creating the image by hand we bypass the requirement of creating and maintaining a code base for creating images. The hand installing method also runs into the detriment of often creating unique images. Without a code base backing this method, there is no documentation of everything installed on the image so when a new image is created they often are unique, which can lead to confusion surrounding the requirements of the software. There is also the problem of relying on human input, which can be unreliable and can lead to having to recreate the image.

3.10.3.3 Hybrid

Another way about creating virtual machine images would be to combine the configuration as code solution and the hand install solution. By using the methods pointed out in the configuration as code solution, we can create a base image that features all of the required software and configurations of several images. We then install the individually required software and code for each software on the code base. For example we can create an image that has proxy settings configured and git installed using the configuration as code solution and then we modify that image by hand to install the code base and the dependencies by hand.

Since this method would combine both hand install and configuration, we would need to use all the software that is used in them. We would need to use Ansible for downloading the ISO, installing the virtual machine, configuring the boot, and the initial configuration of the virtual machine. We would need to use a hypervisor for hosting the virtual machine and to modify the machine after the initial configuration.

The advantage of this method is that we can lower the risk of human error when creating an image. By configuring a generic virtual machine image using configuration as code, we can create a base image with all of the repetitive configuration already done, which is what tends to be missed when done manually. We also ensure that all of the images were booted up the same way. We are also able to save a significant amount of time by using this method as we are not required to create a code base for every piece of software we need. The disadvantages of the hybrid method is that similarly to the hand install method there would still be confusion surrounding the dependencies required by every individual package. Since the individual requirements would be hand installed there would be no place in the image creation where they are documented, which can lead to problems in troubleshooting as software tends to deprecate with time and improper version control can lead to headaches.

3.10.3.4 Advantages

The main advantage of using a virtual machine is that it allows for an OS outside of the system running the image to use. If we went with a docker or podman image we would be forced into using the OS of the system that created the image, but with a virtual machine the options available to us expand. The other advantage is that we do not rely on any base images to create the virtual machine. With docker or podman you are required to provide a base image that you install dependencies on and implant your code into. With the virtual machine we know exactly what is in the image as everything was created by us. Another advantage is that the virtual machine images we create are compatible with docker and podman images. KubeVirt provides a base image for creating virtual machine images. Using that image we can implant our virtual machine image and then create a docker or podman image out of it. This allows us to host the virtual machine image on common web hosting software such as kubernetes.

3.10.3.5 Disadvantages

The main disadvantage of a virtual machine solution is the image size. Since we need to include everything needed to run an entire computer for every image we end up with a rather large image compared to docker or podman. These images tend to be anywhere from 4 Gigabytes to 11 Gigabytes depending on the OS we want to use. Another disadvantage is that most web hosting software is geared towards podman or docker images, which means any networking we need to do with these images becomes significantly harder. These images also take much longer to create than docker or podman images. If we are including downloading the ISO in the creation time then it can take up to an hour and a half per image or even longer if we include compression and docker image translation times. The virtual machine method is also significantly more complicated than the docker or podman solution. Another problem is the lack of documentation surrounding this solution. Ever since docker and podman came out, the use of virtual machines has fallen to the wayside and likewise has been their documentation. Since no one uses them any more, the modern problems we might run into if we choose this method are less likely to have been run into before, and therefore might lack any obvious solutions.

3.10.4 Summary

Given the available paths towards containerization I feel that using Podman would be the best solution for this project. A full virtual machine solution would be overkill for our purposes and lacks the quality of life aspects of Docker and Podman. As seen in **Table 3.35,** a virtual machine solution would also be much more time consuming. While Docker is very comparable to Podman, Docker lacks the security advantages that Podman provides. While Docker does have more documentation than Podman, they both ultimately function in the same way and for our purposes we are using Podman very conventionally. All in all there is no real reason for us to abandon the security advantages that Podman provides for us for the documentation advantages that Docker has.

Table 3.36: Containerization Option Comparison

Technology	Size	Speed	Different OS	Ease of use	Security	Kubernetes
Podman	smallest	fast	no	good	good	supported
Docker	small	fast	no	best	average	supported
Virtual Machine	largest	slow	yes	worst	good	Supported with addons

3.11 Web Deployment

In order to properly communicate between our device and our website, we need to deploy our software onto a web hosting service. This service needs to be able to handle containerized images and have enough processing power to host an image registry, a web server, a website, and a database. This web host also needs to be able to route traffic between these separate services.

3.11.1 Kubernetes

One of the available tools for hosting container images is Kubernetes, which is the industry standard for housing container images. Kubernetes uses yaml files for deploying and managing containerized apps **[14]**. The software provides capabilities for deploying, networking, secret management, and exposing of applications.

3.11.1.1 Kubernetes Features

There are several kinds of configurations for Kubernetes clusters, and there tends to be multiple programs running on clusters at one time. To organize everything running within the cluster, Kubernetes provides two methods of organizing and directing resources. The first method of organizing is to use a namespace **[15]**. All of the defined resources exist within a namespace, which acts as a grouping for resources and their project. This allows the user to define resources with the same name as long as they are in a separated namespace. For example if we had several programs that wanted to use a database, we could section them into different namespaces and create a database deployment in each namespace for them to take advantage of. While namespaces are useful for separating projects, we also need to be able to identify the deployments, which is often done with labels **[16]**. In Kubernetes you often define deployments, which create several other definitions with random names. These deployments need to be recognizable to other deployments, and with the use of labels we can define a key value pair for our deployments to search for. For example if we use a deployment to create several pods, we can label them with our project name and use a service selecting that pair to expose all those pods.

In order to deploy our source code to a cluster there are two common configurations used. The most basic available one is a pod, which is a simple deployment of a containerized image **[17]**. These pod definitions are often wrapped up into a deployment definition, which can be used to deploy multiple pods at once **[18]**. These definitions allow us to deploy our code and configure some of the networking aspects around the code by applying labels to the pod and housing them in a namespace. All pods deployed within the cluster are assigned an IP address usable only within the cluster, but inbound communication from outside the cluster and providing a domain name to the pod requires special configurations.

Kubernetes by default does not expose pods to the outside world or to other intracluster pods. In order to expose these pods, Kubernetes provides service definitions **[19]**. There are several service definitions, but for our project the ones we needed to be using are ClusterIP, and NodePort. ClusterIP allows us to expose a port of a pod within the cluster. It also provides an intracluster DNS entry for that service. This would be used to provide a domain name to a database running within the cluster and expose the database to other programs in the cluster. By using a ClusterIP service for this, we can avoid exposing our database to outside threats, as only intracluster programs would be able to access the database. NodePort services on the other hand are used to expose a port on the cluster to the outside world. NodePort is essentially a way to port forward traffic from outside the cluster to a pod. There is an important limitation with NodePort in that the only ports you can expose are in the range 30000-32767 **[19]**. While NodePort is useful for non-webserver traffic, Kubernetes provides a different route for web server traffic known as an ingress **[20]**. Ingress definitions are reverse proxies that intercept traffic coming into the cluster on ports 80 and 443 and direct them to their corresponding pod within the cluster. The ingress knows where to send the traffic by reading the hostname found in the HTTP request, which allows us to run several websites with different hostnames on the same cluster. We can use an ingress to deploy our image repository and website on the same cluster and direct traffic by giving them individual hostnames.

When deploying programs in different environments, we often need to adjust some of the hard coded variables. With containerized images this is often done by modifying environment variables. Kubernetes provides two distinct resources for configuring environment variables. The first resource available are ConfigMaps, which are simple key value pairs that define the environment variables name and its value **[21]**. ConfigMaps are useful for defining non sensitive values such as port and URLs for services the pod requires, but when handling sensitive data Secrets are used **[22]**. Secrets function the same as ConfigMaps, but they encrypt the data to better secure information such as passwords. Secrets are also used when providing credentials to private image repositories.

Since Kubernetes deploys source code as images, the moment that the image is removed or the cluster is restarted, all the data stored in the image is deleted. This is very problematic when working with databases and image repositories as a blackout can lead to complete loss of user data. Kubernetes provides a solution to this problem with persistent volumes, which are defined file paths on the computer running the cluster, which are shared by pods **[23]**. Persistent volumes (PV) are managed by first creating a PV resource, which defines a file path on the computer and how much available space the

PV can use. This PV can then be claimed by a persistent volume claim (PVC), which is a request for a certain amount of space the cluster wants to use. The PVC is then attached to a pod with a file path that the pod wishes to use. When this is all set up any files that the pod puts into the mounted path are on the path defined by the PV, which means that if the pod is restarted for any reason the data persists.

3.11.1.2 RKE2

There exists a deployment of Kubernetes known as RKE2, which is designed to be in full compliance with the United States government security standards **[24]**. This cluster would provide better security options for our webserver and comes preloaded with helpful tools, which we would want to use in our deployment. One of these tools is Nginx which is an ingress we can use for routing traffic based on the traffic's hostname.

3.11.1.3 Helm

Helm is a graduated project from the Cloud Native Computing Foundation, which allows for managing Kubernetes projects **[25]**. This software allows for the easy deployment, upgrade, and removal of all the kubernetes files essential to a project. The software is structured around building a Helm chart, which contains several yaml files defining the deployment of our project. When we wish to deploy the chart we simply need to provide the chart a name, which can be used to view information about the chart after deployment. Helm also provides basic logic and variable management for our deployment. We can define variables in the values.yaml file and have them be replicated out to the Kubernetes files. We can use helm to create loops if we want to deploy the same deployments with slight modifications, or use them in logic expression to prevent the deployment of certain configurations. If we make any changes to the Kubernetes files, helm provides a simple update method for applying these changes within the cluster.

3.11.1.4 Virtual Machine Deployment

There is an addon known as Kubevirt, which provides Kubernetes clusters with the ability to handle virtual machine images **[26]**. If we need to handle a virtual machine image for our code, then we do not need to install any hypervisor as we can just upgrade our Kubernetes cluster to handle the virtual machine.

3.11.1.5 Multinode

Kubernetes is designed with the capabilities of connecting multiple computers to form one cluster. We can create a master node, which can have several agent nodes connected to it to bolster our cluster's processing power. This capability is important as we can leverage it to increase the scalability of our webserver, which is unnecessary for this project, but important for a realistic deployment of this app.

3.11.2 Docker Compose

Docker Compose is a tool for spinning up multiple containers at the same time and configuring them **[27]**. While Kubernetes relies on its own container handler, Docker Compose uses the Docker daemon, which comes preinstalled with any Docker installation. Similarly to Kubernetes, Docker Compose uses yaml files to configure the deployment. Docker Compose also provides tools for monitoring deployed resources.

3.11.2.1 Docker Compose Features

Docker Compose obtains a project name based off the base name of the projects directory **[28]**. While project names do not provide the same level of separation that namespaces in Kubernetes do, they are much easier configured as they are set for you. They can also be manually set by using the project-directory option when running the Docker Compose command **[28]**. By having a way of separating resources by project, we are able to avoid name errors, as we can use the same name for a resource multiple times.

To deploy images Docker Compose provides services resources **[29]**. While services in Kubernetes are for networking, in Docker Compose they define the pods being deployed onto the Docker daemon. In these services you are also able to define which ports are being port forwarded on the machine. Another big difference with Docker Compose services is that they can also build images and use them, while in Kubernetes deployments are incapable of building images.

By default Docker containers are reachable in the Docker daemon by their name, but Docker also provides links, which allow the user to give the container an extra name **[30]**. By using links, we can easily traffic data intradaemon and using the port forwarding options of the services we are able to expose our containers to the outside world. Configuring an ingress would be slightly more difficult as we would need to either manage an ingress image to use intradaemon or expose multiple ports on our cloud machine that route to these pods.

For providing configurations on a Docker daemon container, you can provide environment variables in the service of the container **[31]**. The environment variables are key value pairs and work the exact same way as in kubernetes. Secrets on the other hand are handled differently as they appear as a file mounted /run/secret and can be provided as either a file reference or an environment variable on the computer running the daemon **[32]**. This can be rather problematic as while kubernetes handles the encryption and decryption of the variable, Docker Compose does not. If we want to encrypt our variables the image reading them must be aware of the encryption and be capable of decrypting the secrets if the program wants to use the information. This could be very problematic when using images outside of what we create as they might not have that capability and we would need to fix that ourselves. When pulling images from private repositories, Docker provides a login command we would run first.

3.11.2.2 Docker Daemon Advantage

Since Docker Compose uses the Docker daemon, Docker Compose has access to all of the images created on that machine. This means we would not need to create a private registry to house our code. By removing that requirement we ensure the security of our code and prevent an avenue for attackers to steal our resources. We also save significant time as we do not need to create a registry.

3.11.3 Web Providers

In order to house our code deployment we needed to take advantage of one of the many cloud providers that provide virtual machines with an ip address accessible anywhere. For our purposes we do not need that strong of a machine to handle our software. If we use an RKE2 cluster, then at a minimum we need 2 CPUs and 4 GBs of ram **[33]**.

3.11.3.1 Digital Ocean

Digital Ocean is one of the many cloud providers available to us. There are many available virtual machine configurations on the site, but for this project the cheapest available option comes to $24.00 a month **[34]**. This option has 2 CPUs, 4 GBs of ram, 80 GiB of storage, and 4 GiB transfer. Digital Ocean also provides a Kubernetes option, but it would cost an extra $4.00 per month for our requirements.

3.11.3.2 Google Cloud

The cheapest virtual machine that meets our requirement that Google Cloud costs $48.91 a month **[35]**. This machine would have 2 CPUs, 8 GBs of ram, configurable storage, and 4 GiB transfer. The machine has about twice the amount of ram that the Digital Ocean machine has, but costs significantly more.

3.11.3.3 AWS

AWS can provide for us a virtual machine that meets our requirement for $37.96 a month **[36]**. This machine would have 2 CPUs, 8 GBs of ram, and 5 GiB transfer. While this machine is slightly more capable than the Digital Ocean option it still costs more, but provides better capabilities than the Google Cloud option for cheaper.

3.11.3.4 Microsoft Azure

For our requirements Azure provides a virtual machine for $65.92 a month **[37]**. This option has 2 CPUs, 4 GBs of ram and 20 GiB of storage. Despite providing less than all the other options available this solution is the most expensive.

3.11.4 Summary

With the available options for the web deployment of our code, I believe the best path forward would be to create a Digital Ocean virtual machine and run a kubernetes cluster on that machine. As seen in **Table 3.36** AWS and Google Cloud both provide a more robust machine then Digital Ocean, our project does not require that much processing power so the cheapest machine that meets the computation requirements of RKE2 is ideal. While we can avoid creating a registry with Docker Compose, Kubernetes provides much more useful tools and options then Docker Compose. As seen in **Table 3.37** Kubernetes is also scalable with other nodes, while Docker Compose can only run on one machine. Kubernetes also is much more of the standard than Docker Compose. Though unlikely if we want to host virtual machines Kubernetes provides capabilities for that.

Table 3.37: Web Host Provider Comparison

Web Host	Digital Ocean	Google Cloud	AWS	Azure
Cost	$24	$48.91	$37.96	$65.92
CPUs	2	2	2	2
Ram	4	8	8	4

Table 3.38: Web App Manager Comparison

Manger	Docker Compose	Kubernetes
Scalable	no	yes
Network Tools	yes	yes
Persistence Tools	yes	yes
Security	present	better
Image Handling	Uses Docker Daemon	Requires a separate running registry

3.12 Device Server Communications

Our device communicates to a webserver hosted on a Kubernetes cluster. Since we are communicating with a web server, it stands to reason that we would need a simple HTTP server running on the device capable of creating requests and receiving responses. Since the Kubernetes cluster is also running multiple different exposed software we also need to bypass on an ingress.

3.12.1 HTTP Messages

In order to build a simple HTTP server we need a rudimentary understanding of HTTP requests. In simple terms an HTTP message is a tcp connection that starts with a request being given to a server and then the server sending its response, then the connection closing. There are 4 HTTP request types, which are GET, PUT, and HEAD. GET is typically used for requesting resources. PUT is used for when you are sending data or a file to the web server. HEAD is used to just request the heading information of the HTTP request. Response and request messages start with their version of HTTP being used and the type/response code of the message. For a typical successful message a response message has a 200 code, but unsuccessful requests have a unique code depending on the error. For example if a resource requested does not exist, a 404 code is given. The messages are split up into two parts: the head and the body. The head is used to describe the data being sent over. The head typically contains information about the body such as the content type and length. The body contains what we wish to send or receive. Inside of the body is where you typically find the HTML data or the json data being sent/recieved.

3.12.2 Ingress

In order to separate the several exposed services our web host is providing, an ingress was utilized. This is a reverse proxy that reads inbound traffic and routes the traffic to its service based on the hostname. This allows us to reuse the ip address of our node as all services are routed based on the domain name we provide for it. We can also manage certificates through the ingress.

3.12.2.1 Nginx

Nginx is an HTTP web server that handles the flow of HTTP requests. Nginx is the industry standard when it comes to routing web traffic **[38]**. There exists an official Nginx Docker image, and Nginx comes preinstalled on an RKE2 cluster. The web server supports load balancing, reverse proxy, tls and ssl. The web server can also handle HTTP/2 and HTTP/3.

3.12.2.2 Istio

Istio is a service mesh infrastructure designed to provide zero-trust security networking **[39]**. Istio provides solutions for tls encryption, load balancing, policy management and access control. Istio is centered around security and can provide observability to the traffic entering your cluster.

3.12.2.3 NodePort

Another way the device can communicate with the webserver is to expose a special port on the system using a NodePort service. This would allow us to completely bypass the ingress and connect onto the webserver with a dedicated port. This can cause a couple of problems, such as managing multiple devices at once, as this does not handle scaling well. This solution also provides no oversight to who is connected on that port, and any ssl security we wish to apply needs to be on the application and not the ingress.

3.12.2.4 Summary

Given that we are planning to use an RKE2 cluster to deploy our app, then it stands to reason that we would use a Nginx ingress. RKE2 comes preloaded with Nginx so we save time on installing the ingress. Also since the ingress is predeployed we avoid any issues with human error that can occur if one of us installed the ingress. While NodePort is also a potential solution, it opens us to potential attacks as we are exposing unnecessary ports.

3.13 Battery

The system is powered by a rechargeable battery pack with a nominal voltage of 14.4 V. This voltage was selected as the main power rail because it provides sufficient headroom above the system's highest power requirement of 12 V, ensuring stable operation under load. This simplifies the power architecture since lower voltages can be derived through the use of either buck or linear regulators, making it the most practical and efficient starting point. The main reason a rechargeable battery pack was chosen was to increase installation flexibility so we are not constrained by outlet location, with the added benefit of enhancing poolside safety by eliminating the tripping hazards associated with power cords.

Several key factors were considered in the battery selection process, including energy density, cycle life, charging time, self-discharge rate, and cost. Charging time can be estimated by dividing the battery's capacity by the charging current **[42]**. These criteria were used to compare and evaluate the two common rechargeable battery types: Lithium-ion and Nickel-Metal Hydride.

3.13.1 Lithium-ion

Lithium-ion rechargeable batteries were considered due to their higher energy density compared to most other rechargeable batteries, low self-discharge rates (typically 1–2% per month **[43]**), and fast charging capability. Li-ion batteries are one of the most widely used options on the market, offering charge cycles ranging from several hundred to a few thousand. Their primary drawback is the potential fire risk if exposed to water, so a waterproof or water-resistant casing would be critical since our device is located poolside.

3.13.2 Nickel-Metal Hydride

Nickel-Metal Hydride (NiMH) batteries were considered despite generally ranking lower than Li-ion batteries across most performance metrics. The cycle life of a NiMH battery is less than that of Li-ion batteries, charge times are slower, and they are less cold tolerant compared to Li-ion batteries. NiMH batteries also exhibit a self-discharge of about 10-15% a month **[43]**. While NiMH batteries are often perceived as more cost-effective, their shorter cycle life and lower capacity can lead to more frequent replacements, resulting in higher overall costs over the lifetime of the system. The primary reason for considering NiMH batteries is their recyclable and safety advantage, they do not catch on fire if exposed to water or more robust handling and are able to be recycled since they do not contain highly reactive materials.

3.13.3 Battery Selection

Table 3.38 shows a comparison of the two battery types in consideration. The Li-ion battery technology was selected due to its higher energy density, allowing more energy to be stored in a smaller and lighter package compared to NiMH batteries. The low self-discharge rate ensures longer charge retention and the faster charging capability means less time spent waiting for the battery to recharge. These factors together led to the selection of Li-ion batteries as the system's power source.

NiMH batteries may be considered in scenarios with moderate energy demands and if cost is a primary concern, though the price difference is relatively minor compared to the tradeoffs in energy performance and self-discharge rate. Although Li-ion batteries present a greater risk of short-circuiting or fire if exposed to water, these safety concerns can be mitigated by using a waterproof enclosure, making the benefits of a Li-ion battery pack worth the tradeoffs.

Table 3.39: Rechargeable Battery Type Comparison

Features	Li-Ion	NiMH
Energy Density	High	Moderate
Cycle Life	High	Moderate
Charging Time	Fast	Slow
Safety Risk	Higher	More Tolerant
Self-Discharge Rate	1–2% per month	10–15% per month
Cost	Higher	Lower

To reduce system complexity and increase safety, pre-assembled battery packs were selected instead of building custom packs. This decision helps prevent wiring or charging errors and ensures reliable integration with other system components.

To reduce system complexity and increase safety, pre-assembled battery packs were selected instead of building custom packs. This decision helps prevent wiring or charging errors and ensures reliable integration with other system components.

Table 3.9 shows the three 14.4V Li-ion battery packs that were evaluated. After reviewing their features as well as the system requirements, the 14.4 V rechargeable battery pack from Liion Wholesale (part number 4S1PMJ1) was selected as the final choice.

This battery pack was chosen because it offers the highest continuous discharge current at the lowest price, with a capacity and watt-hour rating that fall in the middle range of the three options. The battery pack also includes additional features, such as a PCB with safeguards for overcharge, over-discharge, mishandling, and excessive temperatures. The battery is also IEC 62133 and UL 1642 certified, providing the best overall value while meeting both power requirements and safety standards. **[44]**.

Table 3.40: Battery Pack Comparison

Features	Liion Wholesale	GlobTek	Sparkole
Capacity	3500 mAh	5000 mAh	2600 mAh
Watt-Hour Rating (Wh)	50.4 Wh	72 Wh	37.44 Wh
Max Continuous Discharge Current	10A	5A	N/A
Size	75 × 38 × 38 mm	90 × 75 × 31 mm	71 × 37.5 × 36.5 mm
Connector Type	Wire Leads	Wire Leads	JST-VH connector
Certification	IEC 62133, UL 1642	UL2054, IEC62133-2, UN38.3	CE RoHs UN38.3 MSDS
Cost	$36.99	$46	$18.99

3.14 DC Regulators

Our system includes multiple components with varying power requirements. Some components, such as the MCU, require precise operating voltages to avoid damage. Other components, such as the solenoid valves and water pump, are able to operate at a wider voltage range. To ensure each device receives the appropriate power, DC-DC converters are used to regulate and distribute voltage as needed.

There are two main types of DC-DC converters: linear regulators and switching regulators. One of the most important differences between them is efficiency. Linear regulators typically operate with much lower efficiency than switching regulators. An ideal voltage regulator has an output voltage efficiency of 100%, with some switching regulators being able to reach 97% **[46]** while it is only 50% for linear regulators **[47]**.

Each type of regulator has its own use cases, since higher efficiency often comes at the cost of added complexity or design constraints. Another important metric when choosing a DC regulator is to look at the quiescent current. Low quiescent current (Iq) allows the regulator to maintain its regulating functions while drawing only a small amount of current when there is no significant load. This is important in battery-powered applications that spend most of the time in a low-power state. If the Iq is too high, a substantial portion of the battery's energy may be wasted by just waiting. Therefore, a low Iq ($\leq 100\ \mu A$) **[48]** is critical to achieving the desired battery runtime in applications where active operation is periodic.

Choosing the appropriate voltage converters for the system's various power rails involves weighing the trade-offs and determining whether the disadvantages are acceptable for our specific application.

3.14.1 Linear Regulators

Since linear regulators operate in the linear region, they can only step-down voltage, meaning the output must always be less than the input voltage in order to provide a steady output. The main advantage of linear regulators is their simplicity; they are easy to use, with most having built-in compensation that ensures stability without the need for extra components. Other key benefits include low noise (since they do not switch on and off), a space-saving design due to fewer components, and a vast, inexpensive selection available. Since linear regulators can only reduce voltage, their use is more limited. Another disadvantage occurs when the difference between the input and output voltage is large, resulting in a significant amount of power being lost as heat. This results in poor efficiency and requires the use of heat dissipation methods in the design.

Despite these limitations, linear regulator simplicity makes them a practical choice for voltage drops, such as stepping down from 5V to 3.3V, if a quick and reliable solution is needed when budget and PCB board space are limited **[49]**. LDOs(Low Dropout), are a subset of linear regulators designed to operate with a much smaller voltage difference between the input and output compared to traditional linear regulators. LDOs are frequently placed after switching regulators as post-regulators to clean up ripples and switching noise, especially when accurate analog signal conversion is required by components such as ADCs that are used in our optical systems.

3.14.2 Switching Regulators

Switching regulators operate by rapidly turning an active component, like a MOSFET, on and off to perform voltage conversion. Switching regulators have a higher power conversion efficiency than linear regulators but at the cost of more switching noise and greater design complexity, requiring more external components like capacitors and inductors. Switching regulators support a wider range of power supply configurations, allowing multiple output voltages of different polarities to be generated from a single input voltage to meet demands. There are three major switching regulator topologies: Boost, Buck, and Buck-Boost. Boost regulators increase the input voltage to a higher output voltage. Buck regulators lower the input voltage to a lower output voltage. Buck-boost regulators combine features of both buck and boost converters, allowing them to produce an output voltage that can be either higher or lower than the input or, in some types, inverted.

Table 3.39 below summarizes the two main types of voltage regulators discussed and their main features in order to make informed decisions about the needs of our system.

Table 3.41: Regulator Comparison Summary

Features	Linear Regulators	Switching Regulators
Efficiency	Low	High
Switching Noise	None	High
Heat Dissipation	High	Low
Component Count	Low	High

Our system uses a 14.4V battery as the main supply. Both switching and linear regulators are used in our power system design in order to provide the required 12V, 5V, and 3.3V rails. Switching regulators were selected for their higher efficiency and improved thermal performance, which are important for extending battery life. An LDO is included to improve ADC reading accuracy by helping filter noise and reduce voltage ripple, which is beneficial for our more noise-sensitive components.

3.14.3 Voltage Regulator Selection

3.14.3.1 Buck Regulators

Buck regulators are used to step down the 14.4V power rail to supply lower power components operating at 3.3V or 5V. There are two options to consider when selecting buck regulators for the system: using ready-made buck power modules, which simplify implementation, or designing a custom buck regulator circuit using an integrated IC, for greater flexibility and lower cost.

To support all our power requirements, we consider buck regulator modules and ICs with a max input voltage (Vin) of at least 14.4 V, an output voltage (Vout) range of at least 3.3–12 V, and a max output current (Iout) of 3 A. We evaluate four buck regulator options in this section: two ready-made modules and two standalone ICs for use in a custom regulator design.

The TI WEBENCH® Design Center was used to identify potential buck regulator ICs that meet the system's power supply requirements. WEBENCH streamlines the design process by providing schematics, a bill of materials (BOM), and performance simulations. WEBENCH also allows component substitutions with equivalent parts when components are out of stock or too costly. This approach enabled the rapid development of an initial design of a high-efficiency buck regulator, requiring only testing to confirm functionality.

The ready-made buck regulators were selected from commonly used and widely considered reliable modules to ensure the availability of enough reviews and support for troubleshooting. Although these modules usually have lower efficiency than custom designs created using WEBENCH, they are a good substitute when a quick solution is needed.

The DFR0379 DC-DC Buck Converter with Digital Display from DFROBOT was selected due to the popularity and reliability of the monolithic LM2596 buck converter IC from TI, which is widely used for its simplicity and dependable performance. It has a wide input voltage range (up to 40V), adjustable output voltage range (such as stepping down to 5.5V), efficiency of 90%, low quiescent current (around 80 μA), switching frequency of 150 kHz, and built in protections like overcurrent and thermal shutdown makes it a reliable go-to choice **[51]**.

The DFR1015 from DFROBOT was selected for being an easy to acquire buck converter power module that is able to output 4 different fixed voltages that might be required for the system. The buck converter module is able to output up to 5A with a heat sink added, which is the highest of any of the converters on the list, with safety features such as overload and over temperature protections **[52]**.

The TPS563200 regulator was selected not only because it meets the required specifications, but also because the regulator was chosen for its simplicity of implementation, low standby (quiescent) current, and minimal external component requirements. It also offers a high output current, adding flexibility to accommodate the final current requirements of all subsystems **[53]**.

The TPSM862135 was chosen primarily for its straightforward design and ease of integration. As a general-purpose buck converter module, it is especially well-suited for projects where development time is limited. While the quiescent current is higher than that of some alternative ICs, this is an acceptable tradeoff given its simplicity and ability to help keep the project on schedule. Additionally, the availability of a fixed 3.3V output version provides flexibility if a dedicated supply for the MCU is needed later **[54]**.

Table 3.42: Buck Regulator Comparison

Features	DFR0379	DFR1015	TPS563200	TPS62135

Input Voltage Range	3V-40V	7.5V~30V	4.5V-17V	3V – 17V
Output Voltage Range	1.25V-37V	3.3V/5V/9V/12V	0.76V-7V	0.8V – 12V
Max Output Current	3A	3A, 5A	3A	3A
Efficiency	88%	85%	91.3%	89.6%
Quiescent-current(I_Q)	80 µA	150 µA	95µA	18µA
Switching Frequency	150 kHz	500kHz	650 kHz	2.56 MHz
Cost	**$4.90**	**$5.30**	**$0.90 (BOM)**	**$1.41 (BOM)**

3.14.3.2 LDOs

When choosing the optimal LDO for our application, we need to consider the Power Supply Rejection Ratio (PSRR), a critical metric used to measure how well a circuit is able to reject ripples. High PSRR is vital to ensure that fluctuations in the input voltage do not affect the output voltage stability. Another important consideration is the dropout voltage, which is the minimum difference that must be maintained between the input and output voltage for the regulator to function properly. Low dropout voltage is necessary to maximize the regulator efficiency.

Table 3.40 lists two linear regulators to consider, the TLV759P and TLV761 from Texas Instruments. Both LDOs were selected for their affordability, low dropout voltage, low noise, high PSRR, and small footprints but differ mainly in target use cases. The TLV759P was chosen due to its low quiescent current, making it well-suited for powering microcontrollers and other low-power loads in battery-powered systems. The TLV761, on the other hand, was chosen for its effectiveness in post-regulation filtering with a higher PSRR. Both regulators contain an internal soft-start, which reduces excessive inrush current during system startup, and other safety features such as thermal shutdown and foldback current limit for thermal dissipation during a short-circuit event. Both LDOs also exceed the minimum requirements, making them suitable not only for the current design but also for potential future system adjustments.

The thermal performance of both LDOs was evaluated to determine whether their junction temperatures are within the maximum operating limits, in order to make sure that the ICs are not damaged under load. This analysis was particularly important when considering the trade-off of using a linear regulator for the final 5 V to 3.3 V voltage drop, since linear regulators have a lower efficiency and can dissipate a lot of heat compared to switching regulators.

LDO efficiency for this voltage drop was determined using **Equation 3.4**:

$$\text{LDO Efficiency} = \frac{Vo}{Vi} \text{ x } 100\% \text{ } (\textbf{Equation 3.4})$$

For Vo = 3.3 V and Vi = 5.0 V, the efficiency is:

$$LDO_{eff} = \frac{3.3}{5.0} \times 100\% = 66\%$$

A key consideration for the power system is the MCU's current consumption, particularly during power-intensive tasks such as Wi-Fi communication. An MCU such as the ESP32 can draw up to 250 mA during heavy use **[50]**, which directly affects energy consumption and battery life. In order for the junction temperature to be within acceptable limits, the actual dissipation P_D, must be less than the maximum allowable dissipation $P_{D(max)}$. The thermal equations are given below:

$$P_D = (V_{IN} - V_{OUT}) \times (I_{OUT}) \text{ } (\textbf{Equation 3.5}\text{: LDO Power Dissipation})$$

$$P_{D(max)} = \frac{T_{J(max)} - T_A}{R_{\theta JA}} \text{ } (\textbf{Equation 3.6}\text{: Max Power Dissipation})$$

For both LDOs we are going to be using input and output values with some headroom, which gives us Vin = 5.0 V + 5% = 5.25 V, Vout = 3.3 V ± 2% ≈ 3.3 V. The I_{out} = 300 mA and $T_{j(max)}$ = 125°C for both LDOs as well.

For TLV759P, the $R_{\theta JA}$ = 80.3 °C/W while for TLV761 (SOTY package) , the $R_{\theta JA}$ = 95.4 °C/W. Solving for $P_{D(max)}$ for TLV759P we get:

$$P_{D(max)} = \frac{125 - 25}{95.4} = 1.05 \text{ W } (\textbf{Equation 3.7}\text{: Max Power Dissipation for TLV759P})$$

Solving for TLV761 we get:

$$P_{D(max)} = \frac{125 - 25}{80.3} = 1.24 \text{ W } (\textbf{Equation 3.8}\text{: Max Power Dissipation for TLV761})$$

The actual dissipation for this design is found as:

$$P_D = (5.25 - 3.3) \times 0.300 = 0.585 \text{ W } (\textbf{Equation 3.9}\text{:Power Dissipation for 300mA Load})$$

Since $P_{D(max)} > P_D$ for both regulators, the junction temperature remains within acceptable limits. An LDO regulator was ultimately chosen due to its simplicity and low noise, which is important for noise-sensitive circuits. The thermal performance and efficiency of 66% were deemed acceptable tradeoffs for the intended load current and easy integration into the system design. While both the TLV761 and TLV759P offered similar performance, the TLV759P was ultimately selected because it has a higher $P_{D(max)}$ and lower $R_{\theta JA}$, indicating better thermal handling capabilities.

Table 3.43: Linear Regulator Comparison **[92,93]**

Features	TLV759P	TLV761
Input Voltage	1.5 V to 6.0 V	2.5V to 18V
Output Voltage	0.55 V to 5.5 V	0.8V to 13V
Output Current	1A	1A
Dropout Voltage (Vdrop)	225 mV(max) at 1A (3.3V)	225 mV(max) at 1A (3.3V)
Quiescent Current (Iq)	25µA	60µA
PSRR	50dB at 1kHz 45dB at 100kHz 30dB at 1MHz	60dB at 1kHz 40dB at 1MHz
Cost	$0.138	$0.120

Chapter 4 Standards and Design Constraints

4.1 ADC Standard

IEEE 1241 2023, titled *IEEE Standard for Terminology and Test Methods for Analog to Digital Converters*, provides common terminology and test methods for evaluating the performance of ADCs. The standard is mainly meant for ADCs that convert signals in evenly spaced steps, but some parts can still be useful for those that don't follow that pattern. This standard was considered for this project because ADCs are used to measure chlorine and phosphoric acid levels in pool water, and the ESP32's built-in ADC is used for this purpose. The standard defines key ADC concepts such as sampling rate, resolution, and linearity, which are needed for getting accurate digital readings. The standard also explains how to set up ADC tests to achieve repeatable results, including techniques for keeping signals steady and minimizing electrical noise. In addition, the standard also outlines common sources of error such as offset and gain errors and provides guidance on how to detect and avoid them. By applying the guidelines and recommended best practices in this standard, the project achieves more accurate and reliable ADC readings, which are critical for obtaining reliable test results **[91]**.

4.2 Standard Chlorine Testing DPD Method

This system uses N,N-diethyl-p-phenylenediamine (DPD) to detect chlorine with various concentrations at an absorbance corresponding to a particular wavelength. This experiment must be done safely and correctly to ensure accurate results. Without a Standard Operating Procedure (SOP), the results that are received from the system ultimately fail its intended purpose. Some other examples of what constitutes good quality results are Traceability of Measurement, Internal Quality Control Activities, Calibration, Verification, and Maintenance of Instrumentation **[85]**.

For instance, the measurements collected are using certified materials that correspond to an international standard. This way there are biases or user error when analyzing the data collected from the sub-system. Another has a list of test elements that coincide with a data set that has already been published by a well-established scientific research facility.

Chlorine standards are typically referred to as a primary or secondary standard. Primary standards are typically used to calibrate an instrument or check calibration **[85]**. Where secondary standards can only check to see if the proper units are being used for calibration. In best practice, at least four vials should be used and do not contain any leftover residue on the interior and exterior surface. To demonstrate an accurate concentration of chlorine a micro-cuvette is used to collect very small controlled amounts. According to the Environmental Protection Agency (EPA) 334.0 the concentration of Chlorine must be within $\pm 15\%$ of expected values to pass **[85]**. The following can be used to determine the experimental values to pass the calibration check:

$$\frac{(Experimental\ Value - Expected\ Value)}{Expected\ Value} \cdot 100\%$$ (**Equation 4.1**: Calibration check)

Based on the variables listed in **Eqn. 3** the experimental and expected values are derived from their corresponding numerical assessments, which are decided beforehand. To fall outside the $\pm 15\%$ range would mean the calibration was a failure, and would need to be performed again until the values fall into the specified range.

Additionally, the Initial Demonstration of Capability (IDOC) shall be performed so the individual can produce precise and accurate analysis. This needs to be completed using a chlorine standard solution. The amount of time that is allowed before the sample's accuracy diminishes is around 15 minutes **[85]**. Once the allotted time is up the specimen containing the reagent and water must be properly disposed of, otherwise the results would not follow the necessary secondary standard.

According to SM 4500-CL G 2.a.2 the wavelength range that must be used to measure an absorbance peak with an optical path length >1 cm must be from 490 nm - 530 nm **[85]**. Anything above or below those limits are not considered as an accurate measurement and do not follow the primary and secondary standards. Some potential problems that could lead to inaccurate measurements are the following: dirty optics, reagent shelf life, and sample vial is dirty **[85]**. If there are any other problems beside the ones listed it may come from the equipment provided by the manufacturer and needs to be returned immediately.

4.3 Standard Moly. Blue Colorimetric Method

[86] Interestingly, there are very few related standards related to the molybdenum blue colorimetric method. This method can be automated if the Diffusive gradients in thin films (DFT) elution is accounted for in the experiment. From what was found after doing an extensive web search shall be briefly discussed. The necessary reagents that are used are ammonium molybdate, potassium antimonyl tartrate, and ascorbic acid. To ensure proper calibration, the standard solution must have the same concentration of salt as what is extracted using a solvent. The proper amount of time to wait is 15-20 minutes before the color develops and is ready to be measured for this specific analytical procedure. To avoid any contamination, the phosphate can be measured in the eluent portion if high purity reagents neutralize the ions.
From the measured concentration of what is absorbed by the analyte (c_e), the mass of the analyte (M) on the binding layer (V^{bi}) in the solution can be shown as:

$$M = \frac{c_e(V^{bi}+V_e)}{f_e}$$ (**Equation 4.2**: Mass of the Analyte)

[86] From what's being shown in **Eqn. 4.2** are the necessary parameters that need to be considered when the DGT standard is used with water that is flowing or subjected to some type of turbulence. In this instance, the elution factor (f_e) is unity because the ferrihydrite binding gel used with the ascorbic acid was extracted in the exercise.

4.4 WiFi Communications Standards

To facilitate communication between the device and the server, the device is taking advantage of Wifi, which is standardized under the IEEE standard 802.11 **[113]**. The standard defines the bandwidth, speed, and range of WiFi components, and defines several different options depending on the type used **[114]**. The selected MCU for PoolWatch supports the standards 802.11b, 802.11g, 802.11n **[115]**.

4.4.1 802.11b

The 802.11b standard requires that the WiFi signal is operated at 2.4 GHz, with a channel width of 20 MHz, which can provide a maximum Data Rate of 11 Mbps **[114]**. This standard has no additional components to increase its range, but since the standard's signal operates at a smaller frequency then the other standards its wavelength is larger allowing the signal to propagate further. 802.11 has a relatively small channel compared to the other standards, which limits the amount of data that can pass through it. Compared to the other standards, it also has the smallest data rate.

4.4.2 802.11g

The 802.11g standard is an upgrade to the 802.11b standard and provides 54 Mbps, while operating on the same frequency band and width **[114]**. The standard likewise has no methods to expand its range, but still operates within 2.4 GHz allowing for a better range compared to the 5 GHz signal other WiFi standards use. This standard has massively improved upon the data rate without affecting any of the benefits of 802.11b.

4.4.3 802.11n

The 802.11n standard drastically improved the data transfer, bringing it up to 600 Mbps and now the signal operates in both 2.4 GHz and 5GHz frequency bands with a channel width of 20 and 40 MHz **[114]**. The standard also provides MIMO for a single user, which expands its overall range. The increased channel size and frequency band allowed this standard to provide nearly 12 times the data transfer of 802.11g and with the new MIMO technology it was able to expand further than previous standards.

4.5 HTTP Standard

To facilitate the transfer of information between our device and the user, we are taking advantage of a webserver, which communicates using the Hyper Text Transfer Protocol (HTTP). The protocol defines stateless communication between clients and server to enable flexible communication between hypertext systems **[70]**. The messages are broken into a header section, which tends to hold metadata about the request or response, and the body section, which holds the actual data. The protocol is based around the use of methods to make requests of the server and response codes, which indicate the status of the request. Figures **4.1** and **4.2** show the generic structure of HTTP messages.

```
METHOD /route HTTP/version
Header1: value
Header2: value
...

Content
```

Figure 4.1: HTTP Request

```
HTTP/version Status-code Status-text
Header1: value
Header2: value

content
```

Figure 4.2: HTTP Response

4.5.1 Connection Protocol and Data Transfer

The HTTP uses a TCP connection between a client and server. A client in this scenario is an application requesting some resource or service from a server, which is an application handling responses to requests in this scenario. HTTP is stateless, which means that all messages can be understood in isolation **[70]**. The typical route of an HTTP message is that the client connects to the server, then sends a request message, which then the server responds and finally the connection is closed between them. This process is illustrated in **Figure 4.3**.

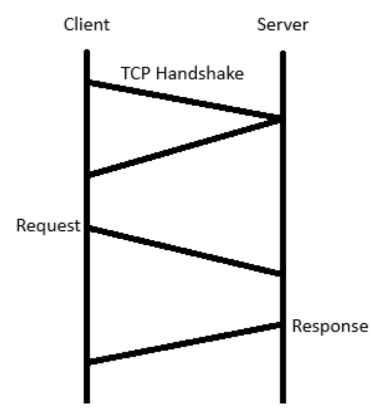


Figure 4.3: HTTP Route

There are several headers that further help with distinguishing the route of the message and type of data. HTTP provides the host header, which is the hostname of the resource requested **[70]**. This header allows for the server to route the message to the proper application if the server hosts multiple applications with different hostnames. HTTP also defines the header Content-Length, which is the whole number of octets of data **[70]**. In most situations the Content-Length can be understood as the amount of characters in the response. This information is helpful when estimating load time of server resources. Most importantly HTTP defines the Content-Type header, which is the kind of data being sent over **[70]**. This header is often used by applications to determine how to handle data, for example a browser would know that a text/html Content-Type needs to be rendered as an HTML file.

4.5.1.1 text/html

As the name suggests one of the most common types of content sent between servers and clients is HTML. The default Content-Type for this resource is text/html **[70]**. This notifies the client to render the body of the message using the html schema of the application. This is most commonly used when providing a web page to a browser.

4.5.1.2 application/json

Another common Content-Type is application/json. This Content-Type is associated with the transfer of json files, which are dictionary based information files. When communicating information from a form or some other form of data collection a client will often store this information as a json file, which is sent to the server. Likewise if a client needs information from a database, the server sends the data in a json file.

4.5.1.3 images

Most websites take advantage of images, and they are transferred with the Content-Type image/<image file type>. There are a lot of image formats used and so to delineate between them to allow the website to render the data the Content-Type header is used.

4.5.2 Methods

HTTP defines multiple methods available for the client to request from the server. The communication between client and server is expected to fall into one of the methods available and the server is expected to be able to handle those requests.

4.5.2.1 GET

The first method available to a request is the GET method, which is used for requesting resources from a server to be sent to the client **[70]**. This request typically receives a 200 response code for a successful completion. The method is used when pulling information from web servers such as a web page, image, or database entry.

4.5.2.2 HEAD

Another common method available is the HEAD method, which functions similarly to the GET method, but just pulls the headers of the requested resource **[70]**. This method should send the same headers as the GET method on the same route unless the header requires the content to be generated first. Likewise to GET the method expects a 200 status code. This method is often used when determining the size of data being requested or whether an application would be able to handle the type of data.

4.5.2.3 POST

The POST method is used for requesting a web server to process content provided to it in the message **[70]**. This method is typically used when providing user information, such as form information or credentials, to a server. Depending on the nature of the request the method can expect a wide range of 2xx status codes.

4.5.2.4 PUT

The PUT method is used when requesting that a webserver create or replace a resource based on the content of the request **[70]**. Similarly to the HEAD and GET methods, this method expects a 200 status code. This method is often used for upload files or database entries to a webserver.

4.5.2.5 DELETE
The DELETE method is used for requesting that a server delete a targeted resource **[70]**. This method typically expects a 202, 204, or 200 status code. As the name suggests this method is used for deleting resources such as files or database entries from a web server.

4.5.2.6 CONNECT
The CONNECT method is used to establish a tunnel to a server and then forward data bidirectionally **[70]**. This method is often used in a proxy server as this method does not actually handle received data.

4.5.2.7 OPTIONS
The OPTIONS method is for requesting information about communication options available **[70]**. This method is used to query what is available on the server to help determine how to develop an API.

4.5.2.8 TRACE
The TACE method requests an application-level loop-back of the message **[70]**. This method expects a 200 response code.

4.5.3 Response Codes

HTTP response messages come with response codes, which indicate the status of the message. These codes are used to indicate if more resources are needed, if the request succeeded, or if it failed.

4.5.3.1 1xx Codes
Codes in the 100s are informational codes and they indicate an interim response for communicating connection **[70]**. Some common codes used in this section are 100 and 101. 100 indicates that the server has received the response and has not decided what to do with it, but will send a final response. 101 indicates a request to upgrade to a different protocol.

4.5.3.2 2xx Codes
Codes in the 200s are success codes. They indicate that the request was successful and some common codes are 200, 201, 202, 203, 204, 205, and 206. 200 is the generic success code and it indicates that the request was received and accepted. 201 is the created code and it indicates that a resource was created. 202 is the accepted code and it indicates that a request has been accepted but not yet processed. 203 is the non-authoritative information code and it indicates that a request was accepted but modified by a proxy. 204 is the no content code and indicates that the server accepted the request, but has nothing to send back. 205 is the reset content request, which indicates that the request was accepted and the client should refresh the content. 206 is the partial content code, which indicates that the server is fulfilling a part of the request.

4.5.3.3 3xx Codes

Codes in the 300s are redirection codes and they indicate that further action is required by the user **[70]**. Some common codes are 300, 301, 302, 303, and 304. 300 is the multiple choices code and it indicates that the resource needs to be specified. 301 is the moved permanent code and it indicates that the resource has been moved to a new URL. 302 is the found code, which indicates that the resource resides under a temporary URL. The 303 code is the see other code and it indicates that the user is being redirected. 304 is the not modified code and it indicates that the conditional GET or HEAD failed the condition.

4.5.3.4 4xx Codes

Codes in the 400s are client error codes and they are related to an error on the clients part **[70]**. Some common codes are 400, 401, 403, 404, 405, and 408. 400 is a bad request, which is a generic failure. 401 is an unauthorized error, which means the client does not have the required permission to complete their request. 403 is the forbidden error, which means the server understood the request, but refused to fulfill it. 404 is the not found error, which means the server could not find the requested resource. 405 is the method not allowed error, which means the client is requesting a method the server does not support. 408 is the request timeout error, which means the server did not receive the full request.

4.5.3.5 5xx Codes

Codes in the 500s are server errors, which means the server ran into a problem when handling the request. Some common codes are 500, 501, 502, 503, 504, and 505. 500 is the internal server error, which is a generic server error. 501 is the not implemented error, which means the requested path has not been implemented. 502 is the bad gateway error, which means that the server received an invalid response. 503 is the service unavailable error, which means the server is currently unable to handle that request. 504 is the gateway timeout error, which means the server timedout. 505 is the HTTP version not supported error, which means that the server can not handle the HTTP version of the request.

4.6 Battery Standard

The ANSI/NEMC C18- Safety Standards for Primary, Secondary, and Lithium Batteries is one of the important standards to highlight for the project. This standard published by The American National Standards Institute outlines the safety requirements for portable primary, secondary, and lithium batteries. Specifically, it applies to Nickel-Cadmium, Nickel-Metal Hydride, and Lithium-ion batteries. The standard outlines performance and mechanical tests for each battery type, along with safety tests and design requirements to ensure reliable operation. Battery standardization is critical, as it promotes consistency in terminology, construction, and safety practices across different manufacturers. One of the main ways this standardization is achieved is through rigorous and wide-ranging safety tests. Clearly defined key terms are essential, as they help clarify the meaning of test results and help manufacturers interpret specific outcomes during safety evaluations.

For example, the standard defines mechanical failure of a battery case or cell container as the leakage of liquid or gas, but not the ejection of solid material **[88].** This definition is used as a reference when the term “rupture” appears in safety testing, providing a consistent interpretation. Without such standardized definitions, different manufacturers might interpret terms like “rupture” differently, potentially leading to inconsistent testing practices and safety assessments. This standard also details the tests that need to be conducted to make sure the battery is safe for both normal use and for reasonable misuse situations. The normal use situation tests include those for vibration, mechanical shock, thermal shock, and altitude tests **[88].** The misuse situations include tests for short circuits, forced discharge, impact and crush **[88].** This is especially important, as manufacturers have to anticipate how batteries could be unintentionally misused and conduct tests to identify those risks and implement needed safeguards to prevent common hazards. As Lithium-ion batteries are going to be used in our system, it is important to handle them correctly as they can catch on fire under thermal stress and lead to battery explosions, thermal burns, and death.

The standard also addresses battery dimensions, standard charging conditions, polarity, and the safe handling of portable lithium-ion batteries under normal use to prevent hazardous conditions **[88].** The design guidelines are followed by larger battery makers such as Duracell **[110]** and Energizer **[111]**, as this standard is one of the main ones used in their datasheets for battery evaluations. The standard also outlines requirements for warning labels and packaging, including precautions for fire hazards and ingestion risks **[88].** This includes the use of “warning” or “caution” labels and poison prevention packaging to clearly indicate the dangers of ingestion and the potential health effects.

Another important standard to highlight is the IEEE 1679.1-2025 standard called IEEE Guide for the Characterization and Evaluation of Lithium-Based Batteries in Stationary Applications. This standard provides guidance for users on the application of lithium-based batteries in stationary settings, specifically focusing on secondary (rechargeable) batteries such as lithium-ion. This standard is intended to be used alongside IEEE Std 1679-2020, the standard which outlines recommended practices for a broader assessment of energy storage technologies.

According to this IEEE standard, as lithium-ion batteries have become increasingly important in stationary energy storage and standby power systems, there is a growing need for clear, standardized information regarding their performance characteristics and safe operating conditions. This standard supports end users by providing a consistent framework for evaluating lithium battery technologies in terms of performance, safety, and reliability, guiding users in selecting and integrating battery technologies effectively **[89]**.

These standards are important since batteries are a key component of our system and because the system is positioned near water, adhering to these safety guidelines is essential for ensuring safe and reliable operation. Lithium batteries are so vital now in portable electronics that their development was recognized with the 2019 Nobel Prize in Chemistry **[90]**, highlighting the importance of compliance with established safety practices.

4.7 Quartz Cuvette Standards

[112] This system uses a UV quartz cuvette to detect chlorine and phosphate with various concentrations at an absorbance peak that corresponds to a particular wavelength. When these small capsules are manufactured there must be a way to ensure the maximum amount of transmitted light can pass through. According to Standard 80 the cuvette is created by adhesive bonding. This means shards or pieces of quartz are stuck together with an optical adhesive. The resultant method would decrease the cost and provide the 10mm path length that is needed to derive the Beer-Lambert Law.

As for what chemicals that are considered safe for usage were neutral pH solutions and small concentration of solvents at room temperature. If the optical adhesives left over are exposed to strong acids, the quartz pieces start to loosen and eventually break off entirely. Precision of using Standard 80 as the catalyst is low compared to other standards, but is sufficient enough for use in spectrophotometry.

[112] Additionally, the purity of the cuvette is extremely important in collecting the spectral transmission, thermal resistance, and precision. Sintered 80 follows a different method of manufacturing compared to Standard 80. Instead of the optical adhesive holding the quartz pieces together, the molecular bonds fuse together at elevated temperatures. This allows the minimum UV wavelength to be around 190-200 nm. Since the current system is going to be placed outside in a high temperature and humid environment there needed to be a temperature limit the cuvette could withstand without any degradation. For Sintered 80 cuvettes have a high chemical resistance to the majority of corrosive acids or organic solvents.

Lastly, precision must be considered when taking into account the uncertainty of the final concentration values. So, with a very low pathlength tolerance and fine parallel glass interfaces this makes Sintered 80 a commonly used as an advantageous metric in colorimetric or spectroscopic systems. The overall cost of Sintered 80 is much higher compared to Standard 80, and this is due to being a guaranteed long lasting product and immune to a vast amount of physical or chemical damage.

Another type of UV quartz manufacturing method that is commonly practiced for high transmission is called Sintered 83 **[112]**. This approach does not include any adhesives to bond the quartz pieces together, but heats up the material via fusion or optical contact and diffusion bonding. Some of the nuance to the process of how the quartz pieces are bonded together are highly resistant against high temperature and very corrosive chemicals. This be useful in determining what manufacturing method the UV quartz cuvette should be considered for this particular system.

4.8 Imaging Standard

ISO 13322-1 defines the requirements of an imaging system that are used to image particulates as well as proper methods of characterizing the imaged particles. This standard is important for us to consider as it recommends what is needed in particle imaging systems. For instance the needed number of pixels per particle as well as the requirements of the illumination. It also brings up the importance of calibration. Without keeping these different design constraints in mind our system will be a poor imager. For characterizing the particles, it mentions the need of several samples, the different methods used for determining particle size, as well as the standard deviation. Without using the set standards for determining size, our measurements could be a lot more arbitrary and useless in practicality. Overall this standard is important to keep in mind as it leads us to creating an imaging system that is able to image particles with reliability and accuracy **[123]**.

4.9 Testing and Presentation Constraint

In our final presentation, the primary constraint was access to a pool, as it was not feasible to have reviewers walk to the nearest one due to limited accessibility. To address this, we proposed using a large fish tank as a substitute, which would allow for easier setup and demonstration. This constraint also impacted our testing, particularly for subsystems like the main water pump, which would require a dedicated water setup area if a pool is unavailable. Using a tank instead of a real pool would also limit full functional capability testing, as the environment of a pool differs significantly from that of a water tank. As a result, testing some of the optical subsystems would require careful evaluation to ensure conditions replicate real-world scenarios as closely as possible.

4.10 Economic Constraint

Our project is fully self-funded, so keeping costs low is essential while still meeting the required functionalities. The system includes three optical subsystems, along with various other components such as valves and pumps, in addition to server hosting which already contribute significantly to the overall cost. To manage the budget effectively, we adopted a balanced approach, selecting more affordable components whenever possible, provided they still met our performance requirements and remained fully operational within the system.

For example, rather than selecting a high-end water pump, we opted for a more cost-effective alternative. This decision was based on the fact that our demonstration would take place in a small tank rather than a full-sized pool, reducing the need for a high-powered pump. Given that the fluid dispensing system included one pump, four valves, and the necessary tubing, we aimed to find components priced around $15 each to stay within budget.

There were limits to how much we could feasibly reduce costs, so knowing that the optical systems would be the most expensive components, we chose to prioritize funding in that area. The main testing functionality of the system depends on the optical subsystems, resulting in necessary but unavoidable costs, such as the inclusion of a camera and an additional microcontroller to support it. To further support long-term reliability, we prioritized components that were readily available and easily replaceable in case of a component breakdown, ensuring that none were obsolete at the time of selection. This focus on accessibility and maintainability helps ensure long-term dependability of the system.

4.11 WiFi Constraints

Table 4.1 summarizes the capabilities of 802.11b, 802.11g, and 802.11n

Table 4.1: WiFi Capabilities **[114] [116]**

WiFi Standard	802.11b	802.11g	802.11n
Data Rate	11 Mbps	54 Mbps	600 Mbps
Maximum Range	100 feet	100 feet	160 feet

The distance that the device can be away from a router can not exceed 160 feet at most and more likely can not be over 100 feet away. This implies that the device can only work within close proximity to the house of any user making above ground pools often placed in the middle of a yard nonviable for this product as it stands. The standards support up to 600 Mbps transfer, while the MCU can handle only 150 Mbps at most **[115]**. This implies that the device is limited in the amount of data it can transfer realistically, however this is not a problem as the device communicates in messages no bigger than 200 bytes.

4.12 Power Constraint

Our system is battery-powered, which places strict power constraints on the selection of components, especially those with high power consumption. This required careful consideration and balancing of power requirements to determine which components were feasible for inclusion into the system. Our team established a maximum power consumption target of 24 W, and a power tree diagram was created to estimate worst-case power usage based on selected components and projected values for remaining ones.

This estimation allowed us to assess whether the overall system would stay within the defined power budget and identify which components consumed the most power. With this insight, we could make informed adjustments to reduce power draw where possible. Since low power consumption was a key design requirement, it was important to ensure the system would not draw excessive power, especially since the system is a portable device, where high power consumption would not be practical. We also accounted for some power headroom in case future components or unforeseen additions increase the system's overall power demands.Staying within power limits is critical to ensure the system's reliability and real-world functionality.

4.13 Time Constraint

The current implementation of the system is based on multiple time constraints. First being the amount of time the battery can last outside near a residential pool. So, the correct selection must be considered based on the individual components that absorb the power being delivered by the MCU. As for the concentration analyzers a calibration curve drastically reduced the amount of time allotted before collecting a single data set from. However, the photodiode and LED should be set to a designated clock signal to ensure the web server doesn't reset without knowing the concentration has been successfully accounted for. For the particulate imaging system, this also needs a designated clock signal so the amount of time it takes to collect an image for one sample won't be too early or late.

4.14 Environmental Constraint

To ensure that our product minimizes its environmental impact we made sure to adhere to the following guidelines. Since we are using chemicals in our phosphate and chlorine measurement, we follow proper disposal guidelines according to EHS. Also, we made sure to pick long lasting components and ensure that they are well protected. The more that a component fails or breaks in our design, the more waste we create. Also, we made sure that the components are easily replaceable. If they are not, that leads to more cases of our entire product being thrown out. If consumers are instead able to replace the single component that fails, less goes to waste.

4.15 Social Constraint

With our product being aimed at the general consumer market, we need to make sure that it is easy to use. If our product is overly complicated, the average user is dissuaded from using it. To make this happen, the website that connects to our device has a simple design, easy to understand. Also the data that appears on our website follow that same vein, we provide the results in common American units. Our device also has a debug panel. This allows users to easily see when something is wrong with the system.

4.16 Health and Safety Constraint

With the potentially hazardous reagents we are using in our measurements we need to make sure we limit the possible harm that can befall our customers. We do this by making sure contact with the reagents is limited. With our automated pump system, all the customer needs to do is initially load the reagent into the device. From there, our pump network feeds the reagent into the cuvettes to undergo tests. The electrical components we use can also potentially cause harm. With our device being made to go right next to water we need to make sure to lower the chance of electrocution. We do this by moving the electrical systems as far away as possible from the pool, locating them in the back of the device. Our system is also fully enclosed to limit the possibilities of someone reaching in and shocking themselves.

4.17 Manufacturability Constraint

Since our product aims to be mass manufactured, we ensure that it is made with currently available methods and machines. We need to ensure that we are not requiring components that need costly custom machinery, they are instead able to be made with what is already existing. This also follows the idea of all of our components being grounded by physical constraints. No components needed experimental manufacturing methods.

Chapter 5 Comparison of ChatGPT

5.1 ChatGPT Use in Web Development

ChatGPT provides a lot of support for code development, which naturally means that the app supports web app development and deployment. Using ChatGPT can vastly speed up the process of creating a full stack app and can help with the research into web app creation. This section reviews ChatGPT's capabilities in handling requests surrounding total web app development.

5.1.1 Kubernetes

For this project, the web app is hosted on a Kubernetes cluster. To review the capabilities of chatGPT when designing cluster configurations, we look into how the system handles requests from simple to complex.

5.1.1.1 Simple Requests

To verify the basic knowledge of ChatGPT, I asked the app several simple questions surrounding Kubernetes and its peripherals **[Chat 1]**. The resulting outcomes were entirely accurate, but there was some ambiguity in the answers surrounding services. In the prompt, ChatGPT states that a service is a stable IP and DNS name for a pod. This is a true statement, but the statement can be interpreted as a service creating a DNS entry accessible from outside the cluster. A service's DNS entry only exists within the cluster and so someone new to Kubernetes may misconstrue the answer and try accessing the pod from outside the cluster using the service.

5.1.1.2 Medium Requests

To test ChatGPT's developing capabilities I asked the app to develop several configurations that would be needed for some of the aspects of the project **[Chat 2]**. The produced results in this test were all accurate, but there were parts in the results, which were bad practices or somewhat incomplete. For example when asked to create a docker registry deployment, ChatGPT also went ahead and created a NodePort service to support the app, however the service does not define which port to use. This means that the cluster will choose a random port between 30000 to 32767, but the prompt continues by saying the registry could be accessed on port 32000, which is more than likely to not be the case. ChatGPT also acknowledged that the user would want a persistent volume claim to be attached to the registry deployment, but makes no mention of a persistent volume, which is required to support the persistent volume claim.

5.1.1.3 Complex Requests

To test the ability of ChatGPT to handle abstract complex tasks I provided a picture of our design to the app and asked the app to develop all the resources needed to support that infrastructure **[Chat 3]**. The overall provided result could theoretically work, but is vulnerable to data loss. The outcome ChatGPT provided ignored any need for persistent volumes, which means that if the cluster were to restart or the database pods were to restart or become corrupted, all of the information stored within them would be lost. ChatGPT did however notice that the main app needs to have the production database's url instantiated as an environment variable, which was something I was expecting ChatGPT to miss.

5.1.1.4 Conclusions

Overall, ChatGPT appears to be able to handle most requests pertaining to Kubernetes development. I would not recommend that someone new to Kubernetes use ChatGPT though as it tends to forget about certain features such as persistent volumes and ChatGPT can be ambiguous at some points, which can cause someone new to Kubernetes confusion. For the most part a proper understanding of Kubernetes would be much more valuable than using ChatGPT to try and handle Kubernetes.

5.2 ChatGPT Use as Knowledge Aid

The team consists of two Photonics students and two Computer Engineering (CpE) students. Since the CpE students have limited background in photonics, many of the concepts related to the photonics portion of the project were unfamiliar to them. To better understand the design choices made by the Photonics students, without requiring them to explain every detail, ChatGPT was used as a basic reference to answer simple questions and help the CpE students grasp the core ideas.

Prompt 1: "How are chemicals tested for in water?"

- Colorimetric (Color-Based) Tests
- Ion-Selective Electrodes
- Titration
- Spectrophotometry
- Laboratory Instrumentation
- Test Strips

The answers provided in the first prompt in Chat 3 by ChatGPT were very broad and generalized, which was expected given that the first question was also intentionally broad. The goal of this initial question was to see whether ChatGPT could identify commonly used tests in chemical analysis. This prompt served as a starting point to identify some of the most common answers and acted as a guide for further prompting if the initial response was not helpful. The answers provided by ChatGPT helped clarify what needed to be done to narrow down the search criteria by listing some of the components the Photonics students would be using, which served as clues for further investigation.

Prompt 2: “Elaborate for tests using lasers and lenses”

- Absorption Spectroscopy
- Laser-Induced Fluorescence (LIF)
- Turbidity or Scattering Detection
- Raman Spectroscopy

The answers provided by prompt 2 in Chat 3 ChatGPT aligned with discussions held between group members and helped identify the names of tests that could be conducted. Reading the short explanations also helped solidify the understanding of how lasers can be used to measure the concentration of chemicals in water. The explanations provide a clearer idea of the requirements for designing such a system, including which protocols like ADC that would be needed. When ChatGPT was asked specifically about the name of the test being conducted for specific chemicals like Chlorine or Fluoride, ChatGPT responded with LED absorption spectroscopy. This term was then shared with a Photonics team member, who clarified that the use of *spectroscopy* was inaccurate in this context, as that usage typically involves multiple wavelengths, whereas the current design uses only a single, specific wavelength.

From this prompt, ChatGPT provided follow-up information, including examples of the types of systems the technology could be implemented in, such as pool systems. From there, more specific and targeted questions could be asked based on the proposed setup chosen. The follow-up prompts suggested by ChatGPT served as starting points for potential research directions and highlighted areas of investigation that could be explored further.

Overall, ChatGPT was a valuable tool for understanding parts of the system that some team members were unfamiliar with, especially where foundational knowledge was limited due to minimal overlap in courses taken about a specific field. Using ChatGPT as a starting tool to understand basic concepts as to how a system worked was a great way. Since meetings were limited due to time constraints, ChatGPT served as a helpful resource for quickly understanding core concepts. This knowledge could then be verified with team members who had more expertise in the subject. This allowed different sections of the project to progress more efficiently, as team members could be asked more targeted and relevant questions, minimizing the need for basic explanations.

5.3 ChatGPT Pump/Valve Design Comparison

Before going into what sort of response received from ChatGPT, the acronym “GPT” stands for Generative Pre-trained Transformer. When someone generates a dataset the algorithm will be trained in a way so that any question introduced will respond similarly to how a human would in mid-conversation. Since ChatGPT uses a deep learning model the amount of training sets would be comparable to the amount a user inputs into the prompt. Typically, a sequential process occurs when an initial prompt or question will potentially generate multiple answers. While the user can either accept one answer or none, this will essentially “reward” the model to be used again in the next epoch iteration.

From the initial response received from ChatGPT after the following question,”How should I design a pump and valve system for pool water?” and the response provided was:

- Are you filtering, dosing, sampling, measuring (e.g., optical detection), or moving water between levels?
- Desired flow rate?
- Total Dynamic Head (TDH)
- Water quality

It did not seem to really understand the very broad question which makes sense, but also provided me with more questions to ask myself rather than immediately trying to answer the initial question that was asked. When considering the key pump parameters there are many different factors to take into consideration. For example, the type of pump such as a centrifugal or peristaltic/diaphragm was something the group had already considered **[Chat 5]**. This reinforces our initial questions and the brainstorming process was successful for this part of the system.

Once the valves were evaluated by ChatGPT there were different valves provided with a use case associated with them. For each option the LLM ensured that the listed valves were appropriate to use with chlorinated water which was not something I did not specify in the initial prompt, but assumed the presence of chlorine already in the pool water.

The option to choose a solenoid valve was provided, which again confirmed the selected valve in this design would be a valuable component to be used. For each use case the small summary did not seem to be useful since there was no mention of how each individual valve worked in detail. An extensive search was conducted instead using a web browser to obtain more knowledge on the individual valves.

After the valves, the layout of the plumbing was simplified and provided in a diagram with arrows pointing to the connections that would go between individual components. It chose components for the user and attempted to put them together, but some of the components added had a flow split and solenoid valve connected to the sensor chamber **[Chat 5]**. Furthermore, the diagram provided two different valves. One would function to provide anything that was in the sampling line, but the second would prevent any backflow from occurring. This was useful to learn the bigger picture of what the Pump/valve system could potentially evolve into.

Something else that ChatGPT provided were some critical considerations to think about when designing this particular system. For instance, safety was a concern since liquid and electricity does not act as an insulator when both come into contact with each other. This was considered a high tier risk **[Chat 5].** To mitigate the risk the bot recommended using waterproof enclosures for all electronics**.**

Also, some useful tips the bot left behind made me aware of potential problems that may occur if the selected material of the tubing would not be compatible with chlorine in the initial design. However, this was proven to not be an issue after a chart provided a definitive answer that there would be no issue using polyethylene as the polymer is highly resistant to chlorine at low concentrations **[87]**. The chemicals that were marked with "A" essentially meant there was no reason for chlorine water to corrode with polyethylene.

Chapter 6 Hardware Design

6.1 Solenoid Valves and Pump Design

Figure 6.1 shows how to drive a 12 V relay using 3.3 V logic from the ESP32 microcontroller. The 12 V relay contains a coil that requires more current than a GPIO pin can provide, meaning the ESP32 GPIO pin cannot directly drive the relay coil because it is limited to a small output current. To address this, the relay module includes an optocoupler and a transistor, which serve as a bridge between the low-power 3.3 V control signal and the higher current required by the 12 V relay coil. This design allows the ESP32 to control the relay safely without overloading its GPIO pins. The relay module currently in use is the 12 V 2-Channel Relay from HiLetgo. When configured for high-trigger mode using the jumper, the relay activates whenever the input voltage is greater than or equal to 1.5 V DC.

The same approach is to be used for all connected loads, such as the water pump and the three solenoid valves. Each relay channel controls only one load, meaning one pump and three solenoid valves require a total of four channels. The current setup uses two 2-channel relay modules, with an additional 1-channel relay module available for design flexibility. All relay coils are powered by the same 12 V supply, so the power source must be capable of delivering the total required current without causing a voltage drop or shutting down. They are powered through a 12 V regulator rated for a maximum of 3 A. Since each relay channel is expected to draw about 50–60 mA when active, the available current is more than sufficient for our use.

The COM terminals of each relay connect to the positive supply used by the load. If the load is connected to the NC (Normally Closed) terminal, it is powered when the relay is inactive and turns off when the relay is energized. If the load is connected to the NO (Normally Open) terminal, it remains unpowered when the relay is inactive and only receives power when the relay is energized.

The relay modules also include flyback diodes across each relay coil to suppress voltage spikes produced when the coil is switched off. In addition, flyback diodes are also placed across each inductive load, such as pumps and solenoid valves, to protect the circuit from back EMF.

The figure below illustrates the internal schematic of the 12 V relay module and how the inductive loads are connected to both the ESP32 and the relay. GPIO34, GPIO36, and

GPIO39 are assigned to control the solenoid valves, while GPIO4 is used to control the water pump. A second figure shows the relay connections to the DC voltage source and the common ground.

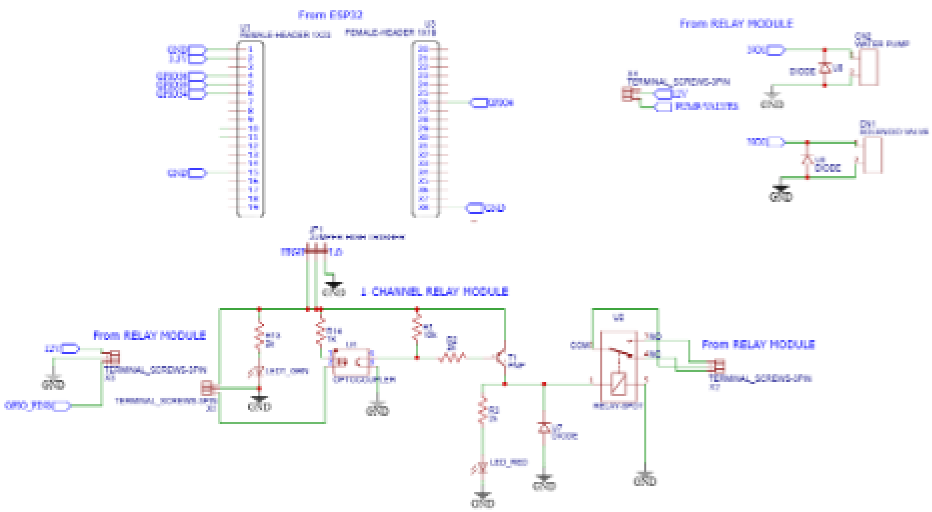


Fig. 6.1:. 12V Relay Module Connections

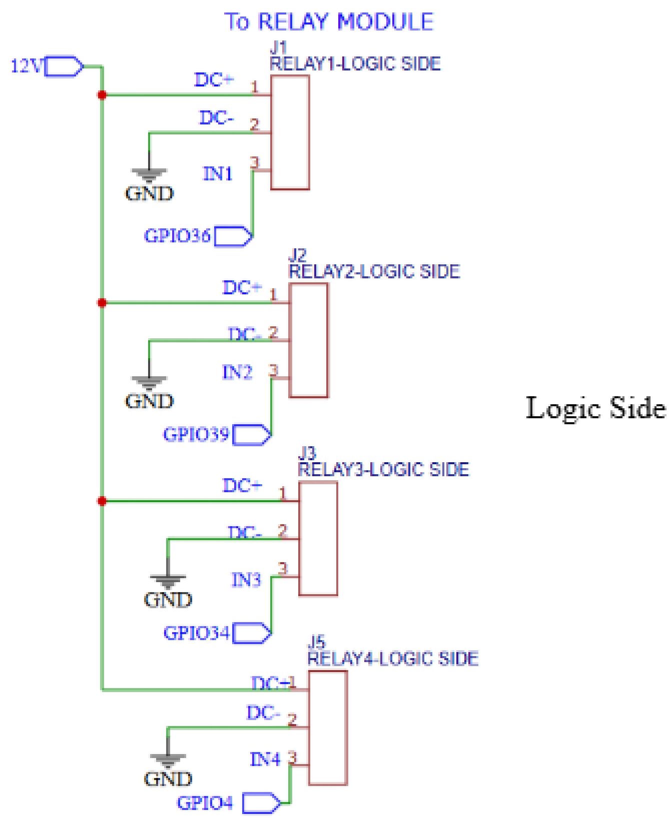


Fig. 6.2:. 12V Relay Logic Side Connection

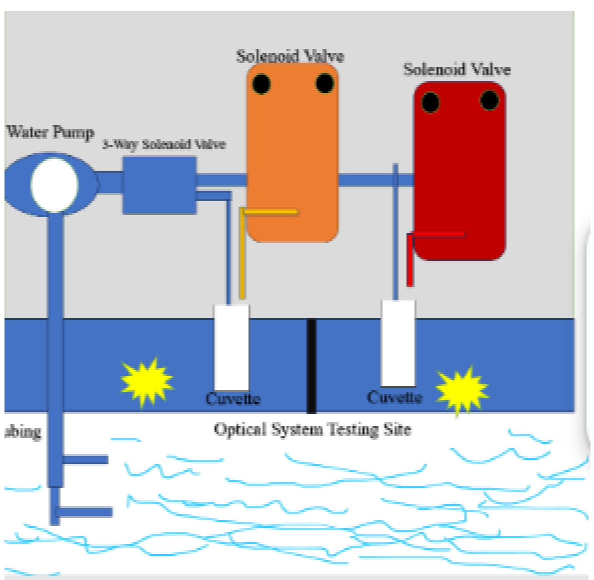


Fig. 6.3: Dosing System and Pump Layout Design

This is the initial setup design for the pool pump and dosing system. The peristaltic pump draws water up from the pool at one end and then connects to a three-way solenoid valve at the valve's inlet. The solenoid valve is also connected to more tubing on the two outlets, which has tubing that leads to the cuvettes. The two solenoid valves for the reagent dispensing are gravity fed and released into small connectors that lead into the cuvettes as well.The current design uses a wall-mounted setup for the pump and valves, but it may be replaced with brackets or redesigned to be installed horizontally rather than vertically.

6.2 Detection Systems Hardware Design

6.2.1 Chlorine Concentration Circuit Design

The diode that should be used for the final design to ensure the chlorine detection system was a 515 nm LED with a 180Ω resistor. The power supply would need to provide 3.3 V which means 15mA of current was driven from the resistor to the LED in the circuit.

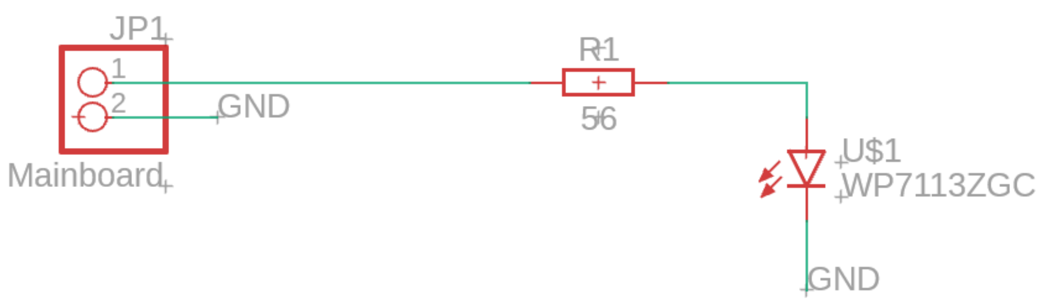


Figure 6.4: Basic 515 nm LED Design

The individual components in **Figure 6.4** show the resistor drives enough current to allow the LED to operate in mode, so the charge carrier lifetime shortens as the holes and electrons recombine and emit more energy than the actual bandgap energy of the semiconductor. For the current design only a DC voltage is required with a small resistance value. At the specified forward voltage and resistance drive a 20mA current to the diode.

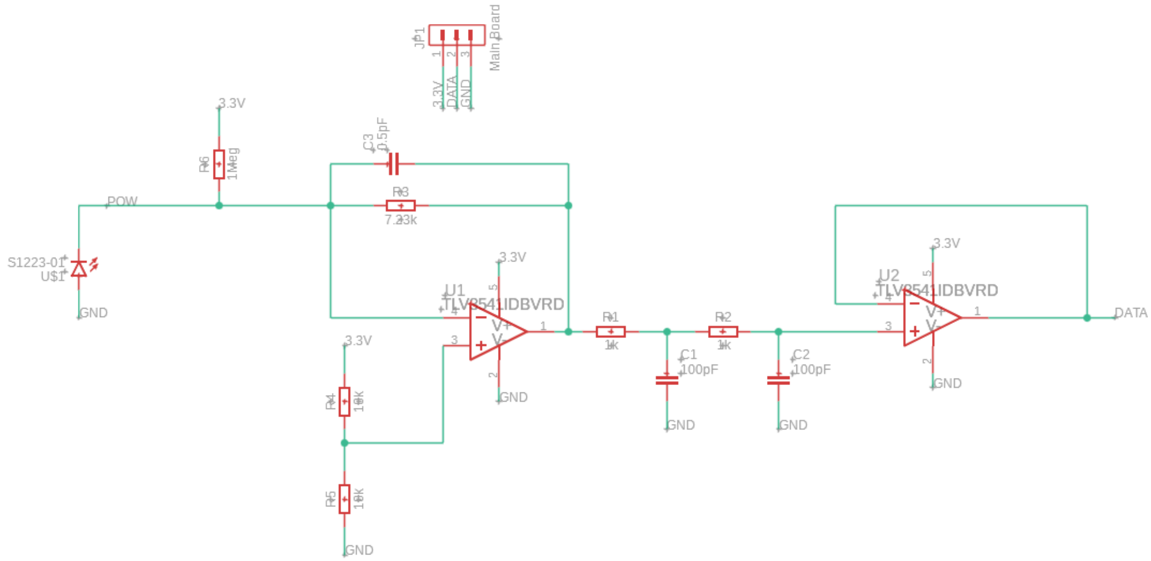


Figure 6.5: Basic PIN Photodiode design for Chlorine Detection System

For the initial design shown in **Figure 6.5** a 4N25 optocoupler was used to represent the photodiode for the chlorine detection system. By applying a reverse biased voltage of 3.3V, this increases the width of the depletion layer in the semiconductor material and decreases the response time. The photodiode acts in a photoconductive mode which is the desired to convert the quantum efficiency to a measurable DC current. Then a 2nd order low pass Sallen-Key low-pass filter is used to reduce the amount of shot and thermal noise coming from the photodiode and feedback resistor. The output pin D27 converts the sinusoidal signal to a discrete value which coincides with the calibration concentration curve and outputs a numerical value in parts per million.

6.2.2 Phosphate Concentration Circuit Design

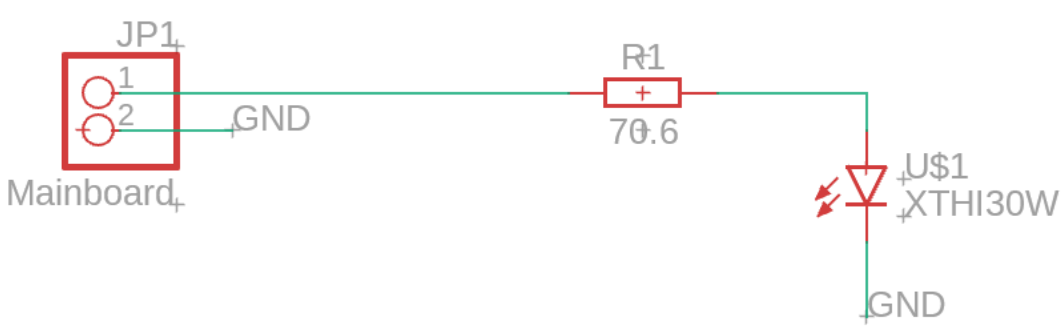


Figure 6.6: Basic 880 nm LED Design

The individual components in **Figure 6.6** show the resistor drives enough current to allow the LED to operate in a particular mode, so the charge carrier lifetime shortens as the holes and electrons recombine and emit more energy than the actual bandgap energy of the semiconductor. For the current design only a DC voltage is required with a small resistance value. At the specified forward voltage and resistance drive a 20mA current to the diode

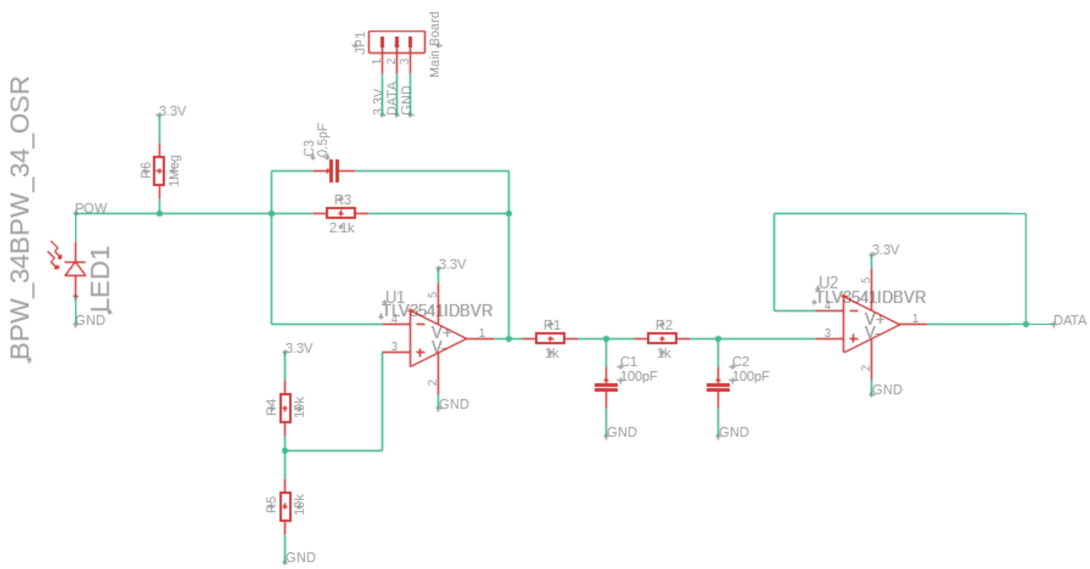


Figure 6.7: Basic PIN Photodiode Design for Phosphate Detection System

Correspondingly, the initial design shown in **Figure 6.7** a 4N25 optocoupler was used to represent the photodiode for the phosphate detection system. By applying a reverse biased voltage of 3.3V, this also increases the width of the depletion layer in the semiconductor material and decreases the response time. The photodiode also acts in a photoconductive mode which is the desired to convert the quantum efficiency to a measurable DC current. Then a 2nd order low pass Sallen-Key low-pass filter is also used to reduce the amount of shot and thermal noise coming from the photodiode and feedback resistor. The output pin D14 converts the sinusoidal signal to a discrete value which coincides with the calibration concentration curve and outputs a numerical value in parts per million.

6.3 Particle Measurement System Hardware Design

6.3.1 Lens Design for Imager

For the lens design to properly image the small particles the first thing needed to be determined was the size of the particles being imaged. Below 63 microns is typically organic matter, 63-250 microns is typically fine sand, and greater than 250 microns is coarser sands **[109]**. To allow for accurate measurements of particles on the lower end the target size is 10 microns for the smallest discernible feature. Now that we know what size we are imaging it is important to determine the numerical aperture so that we can do the ray tracing to get the basic lens design. The field of view of our images plays a big role in this. Ideally, we get the maximum amount possible constrained by our cuvette to get the most accurate reading possible, but it is also important to consider the size of our imaging sensor. The cuvette can provide 10 mm of imaging area while the sensor is only 5 mm at max. The size of the individual pixels of the sensor is also important to consider. Since we want to accurately detect particles that are of the smallest feature we must follow the Nyquist-Shannon sampling theorem, where our detection rate must be half the size of the smallest feature to accurately detect them. It is important to note that the smallest feature size is the particle after it goes through the imaging system, using **Equation 6.1** you can determine what this size is. The last thing that constrains our numerical aperture is the need to have a small enough airy disk that we are able to capture the 10 micron particles.

$$I_h = m * O_h \qquad \text{(\textbf{Equation 6.1}: Image Height)}$$

This can then be all put together. Since we want the largest field of view possible and our sensor is smaller than our object, optimally we decrease our magnification as much as possible down to 0.5x, this comes from our sensor being half the size of our object. The minimum magnification is constrained by the minimum size that can be accurately measured by our pixels. Two times our pixel size would be roughly 7 micrometers, this can then be turned into magnification using equation 6.1 rearranged. This gives us a lower bound of 0.7x. We can then determine the maximum f number using **Equation 6.2**. This gives us a maximum image side working f number of 5.5. Using **Equation 6.3** this can be turned into the image side numerical aperture.

$$\frac{d}{2} = 1.22 * \lambda * F \qquad \text{(\textbf{Equation 6.2}: Diffraction Limited Spot Size)}$$

$$NA = \frac{1}{2*F} \qquad \text{(\textbf{Equation 6.3}: Numerical Aperture from f/\#)}$$

Now that we know what we want from our optical system we can now determine how we are going to achieve this. We are using a two lens system to help limit the spherical aberration without spending too much on lenses. With the image side numerical aperture we can determine the corresponding angle with **Equation 6.4**, we will call this u_3. From this we can determine the angle on the object side of the image, u_1, with the magnification, this is **Equation 6.5**. These two angles relate to the total amount of angle change that must be achieved by the system. This can be divided up to each lens, through some testing I found that this is best done with ⅔ of the angle change being done by the first lens and the rest being done by the second. With the total amount of change needing to be done known we can determine the angle coming out of the first lens with **Equation 6.6**. Since the cuvettes width is 10mm it makes sense to choose a lens whose diameter is close to that width, a common diameter is 12.7mm, so this was used. Half of this is the height that the ray reaches at the location of the first lens. With the angle and height the distance can be determined as well as the power of the first lens with **Equation 6.7**. The inverse of this is the effective focal length which was determined to be 43 mm with the lens being 48.6mm from the object. This can then be related to commonly found focal lengths in commercial lenses, 38.1 mm. With this new focal length the angle change required by the second lens can then be redetermined. The placement of the second lens can also be considered. Since the second lens is responsible for roughly ⅓ the angle change of the first lens it can be placed at a distance ⅓ of the way to where the light would converge without the second lens being there. This is calculated with **Equation 6.8**. With this the power of the second lens can be determined with the following **Equation 6.9**. Then the effective focal length can be seen to be roughly 50.38mm, with a distance from the first lens of 110mm. Since this lens is further away from the original object it needs to have a larger diameter to capture the off angle light. Relating this to a common manufactured lens we get a focal length of 50.8mm. We are now ready to import the lenses to Zemax and optimize the system.

$$\alpha = sin^{-1}(NA) \qquad \text{(Equation 6.4: Angular Aperture)}$$

$$u_1 = \frac{u_3}{m} \qquad \text{(Equation 6.5: Marginal Ray Angle in Object Space)}$$

$$u' = u_{total} * \frac{2}{3} + u_1 \qquad \text{(Equation 6.6: Marginal Ray Angle Between Lenses)}$$

$$k_1 = -(u' - u_1)/h \qquad \text{(Equation 6.7: Power of Lens 1)}$$

$$y = (0 - h)/(u' * 3) \qquad \text{(Equation 6.8: Distance Between Lenses)}$$

$$k_2 =- (u_3 - u')/(x * tan(u_1) + u' * y) \qquad \text{(Equation 6.9: Power of 2nd Lens)}$$

In Zemax we can import the chosen Newport achromatic lenses with the focal lengths found in the ray tracing approximation. Note that the two lenses are faced with the crowns towards each other and the flints facing the object and image respectively. Now letting Zemax change the distance between the two lenses we are able to let it optimize to minimize the spherical aberration, get closer to the desired magnification, and the desired numerical aperture. This optimization left us with a final magnification of 1.1x and an image side numerical aperture of 0.122. It is also limited geometrically at 8 µm.

The following figure shows how the imaging system looks.

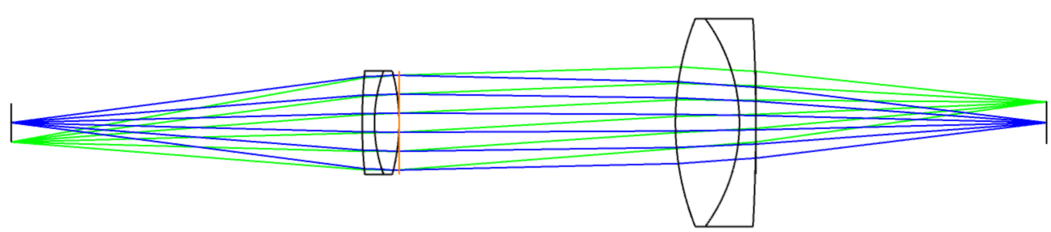
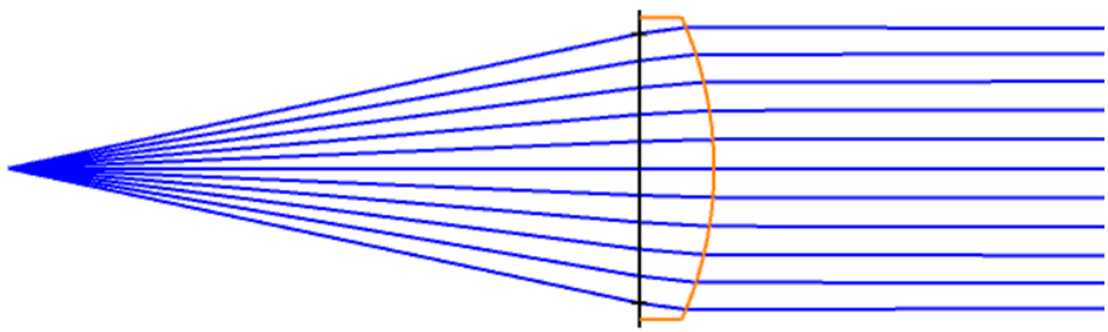


Figure 6.8: Particulate Imaging System Simulation

6.3.2 Lens Design for LED Collimation

In order to perfectly collimate the green led light we first had to determine the angle at which the light left the LED. Experimentally this was found to be 12 degrees. Since we want the light to match the diameter of the rest of our system we can then determine the distance from the LED the lens should be placed for the light to have a diameter of 12.7 mm. This comes out to be 30mm, this number is also the focal length of the lens that should be used. For the infrared LED the angle the light left the LED was determined to be 14.9 degrees. Then following the same guidelines as the green LED the distance the lens needed to be placed at was found to be 23.8 mm. This is not a common focal length, because of this the focal length that was decided to be used was 20 mm. This shorter focal length would capture more light and, with verification from Zemax, would still provide a good enough collimation over the distance of our system. See figure 6.4 for the LED collimation simulations.



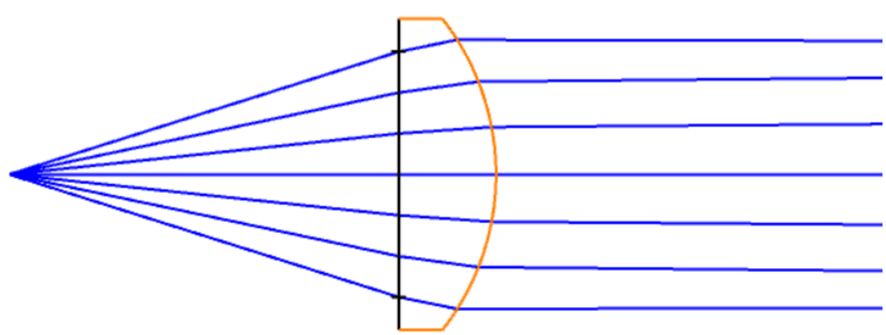


Figure 6.9: Green LED Collimation Simulation and Infrared Collimation Simulation

6.4 Temperature Sensor Design

To connect the temperature sensor to the MCU the following circuit is used **Figure 6.10**.

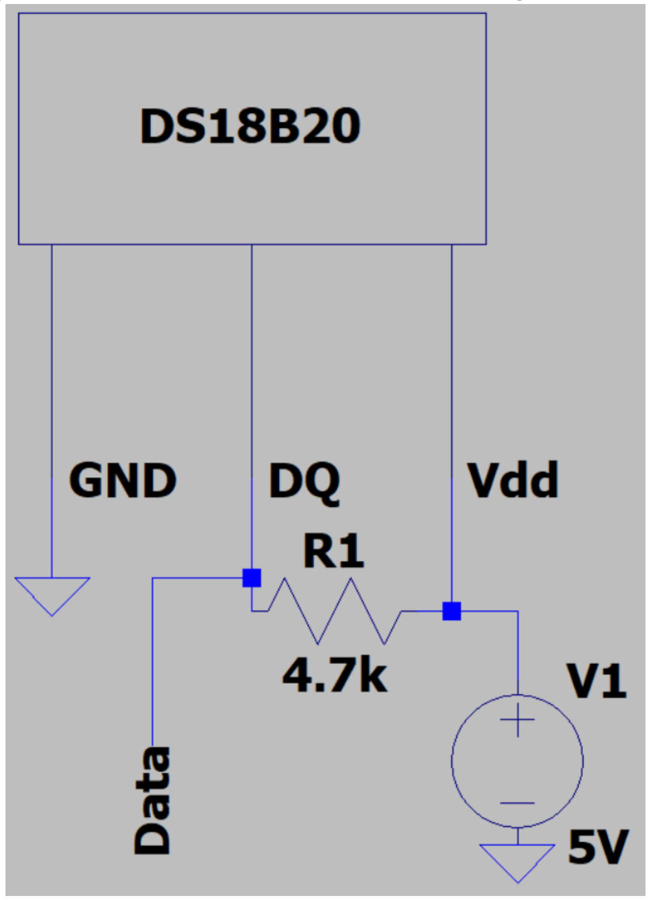


Figure 6.10: Schematic of Temperature Sensor

The temperature sensor requires a 5 volt supply at 1 mA. This is used as a supply as well as part of a pull up system using a 4.7kΩ resistor. The data goes to pin 18 on the MCU, providing its measured temperature.

6.5 Debug Panel

The debug panel is a series of LEDs that indicate when a switch is flipped which systems are working. The panel needs to have an LED for each system. We also want to minimize the amount of power consumed by this system to ensure sustainability of our power supply.

6.5.1 Individual LED Design

Pictured in **Figure 6.5** is a generic model of an individual panel LED. In the model we receive two signals, the debug switch and the Working signal. The debug switch, Vcc, and Working signal would all be the logical voltage of our MCU, which is 3.3V. When either the Debug Switch or the Working Signal are low there is no current flowing through our system, and therefore we are not consuming any power. This implies that for our power analysis we need only worry about when both signals are high.

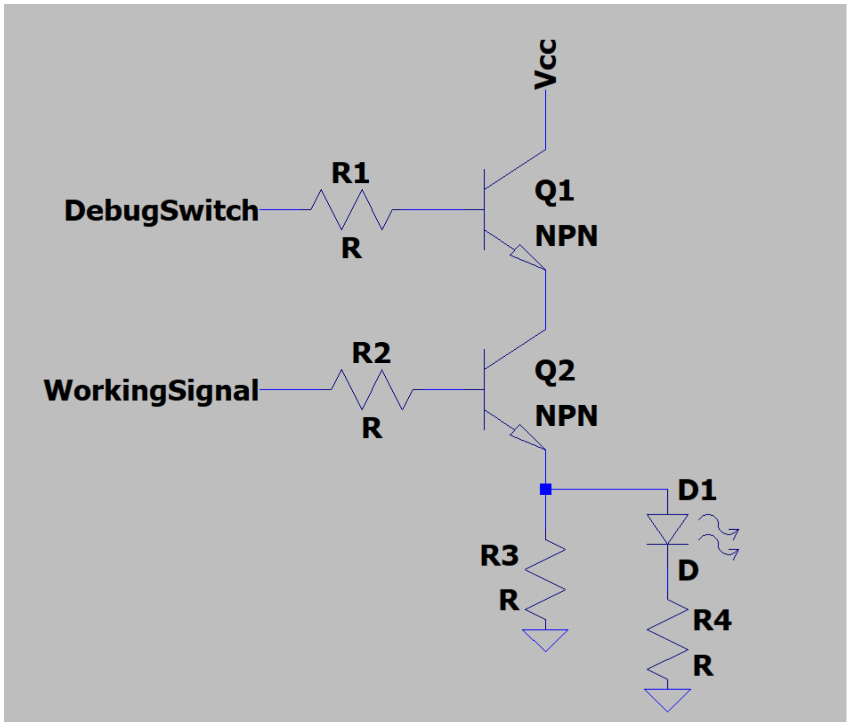


Figure 6.11: Individual LED Design

We can then simplify our circuit into **Figure 6.12** by assuming that the collectant current and emitter current are the same and that the LED has a drop of .7 V across it when on.

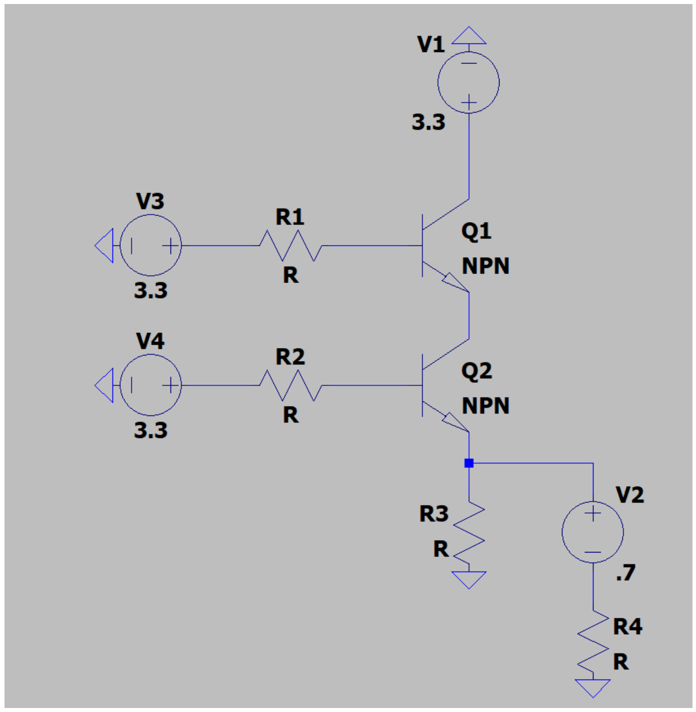


Figure 6.12: Simplified Individual LED Design

For this project we are using BFU520WX transistors, which have a current gain between 60 and 200, typically around 95 **[41]**. Solving for the power consumption of the individual elements of the circuit when both inputs are high, we get the following equations:

$$P_{R1} = \frac{34.81}{R(\beta+3)^2}$$ (**Equation 6.10**: Power of Resistor 1)

$$P_{R2} = \frac{34.81}{R(\beta+3)^2}$$ (**Equation 6.11**: Power of Resistor 2)

$$P_{R3} = \frac{\left(\frac{5.9(\beta+1)}{\beta+3}+.7\right)^2}{4R}$$ (**Equation 6.12**: Power of Resistor 3)

$$P_{R4} = \frac{\left(\frac{5.9(\beta+1)}{\beta+3}-.7\right)^2}{4R}$$ (**Equation 6.13**: Power of Resistor 4)

$$P_{V2} = .7\left(\frac{\frac{5.9(\beta+1)}{\beta+3}-.7}{2R}\right)$$ (**Equation 6.14**: Power of Diode)

$$P_{Q1} = \frac{(.7 - \frac{5.9}{\beta+3})(\frac{5.9(\beta+1)}{\beta+3})}{R}$$ (**Equation 6.15**: Power of NPN 1)

$$P_{Q2} = 0$$ (**Equation 6.16**: Power of NPN 2)

Solving for the total power of the system in relation to the resistance for the current gain we get the following equations:

$$P_{60} = \frac{21.8}{R}$$ (**Equation 6.17**: Total Power at a Current Gain of 60)

$$P_{95} = \frac{22.431}{R}$$ (**Equation 6.18**: Total Power at a Current Gain of 95)

$$P_{200} = \frac{23.031}{R}$$ (**Equation 6.19**: Total Power at a Current Gain of 200)

In order to make sure that the LED is on we must also ensure that there is at least a voltage of .7 V across it. We can model that voltage as shown in **Equation 6.20**.

$$V_{LED} = \frac{2.95(\beta+1)}{\beta+3} + .35$$ (**Equation 6.20** Voltage Across LED)

These equations indicate that no matter the current gain of the circuit as the resistance increases the power consumed by the circuit decreases. Ultimately With variations in resistance tolerances I believe we should use 10 kΩ for our resistance, which would result in our final circuit design shown in **Figure 6.13**.

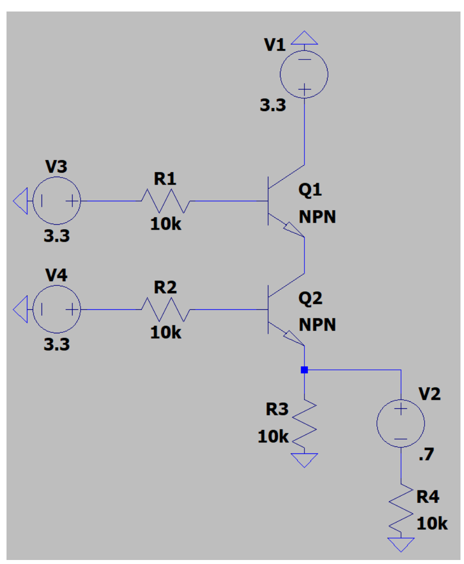


Figure 6.13: Final LED Circuit Design

6.5.2 Total Panel Design

Pictured in **Figure 6.14** is the total Schematic for the Debug Panel

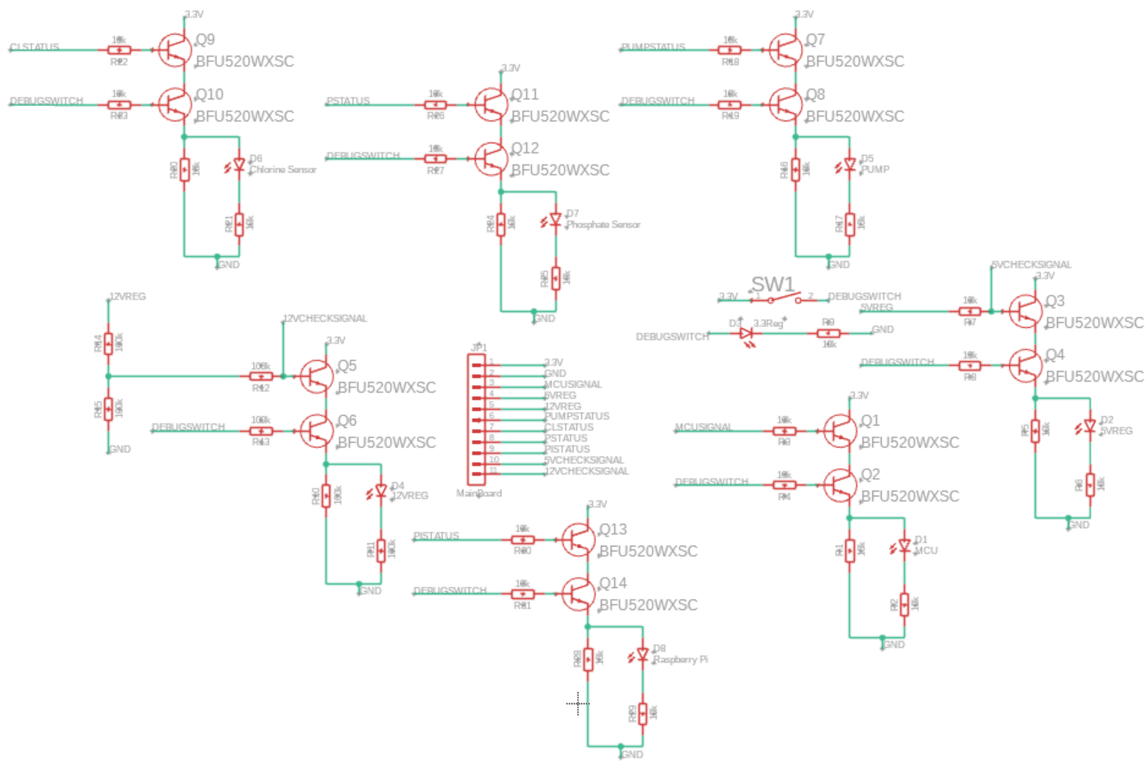


Figure 6.14: Total Debug Panel Schematic

6.6 Raspberry Pi Connections

To control the camera a Raspberry Pi is used. It then must be connected to the MCU to be controlled. **Figure 6.15** shows how the Raspberry Pi is connected.

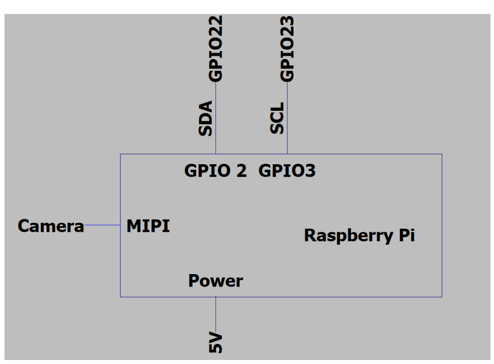


Figure 6.15: Raspberry Pi Pinout

The Raspberry Pi requires a 5 volts at 3 amps voltage supply. It uses a MIPI connection to control and receive signals from the camera. It is controlled by the MCU using I2C, from the Raspberry Pi's second and third GPIO pins to GPIO 22 and 23 on the MCU.

6.7 MCU Board

To host our MCU and connections to peripherals the following schematic shown in **Figure 6.16** is used. This board also provides programming capabilities to the MCU, and provide ADC to measure battery level, as well as a reset button and off and on switch.

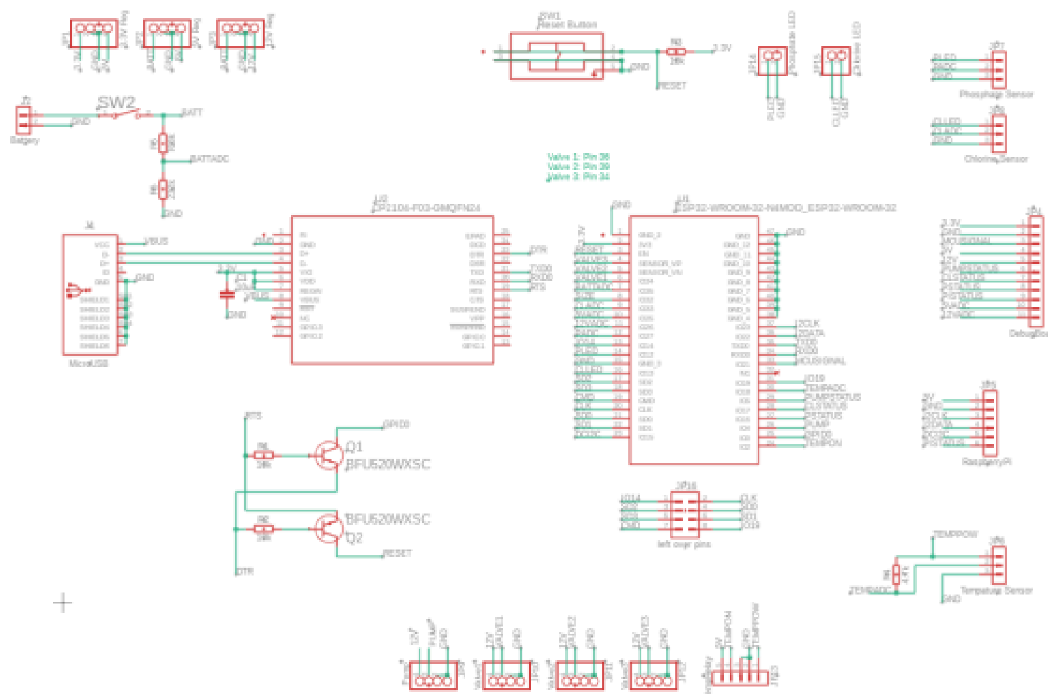


Figure 6.16: Main Board Schematic

We were advised to modify this main board to use a different UART to USB bridge, then what we had chosen, and we also added extra grounds to our main board, which became the design shown in **Figure 6.17**.

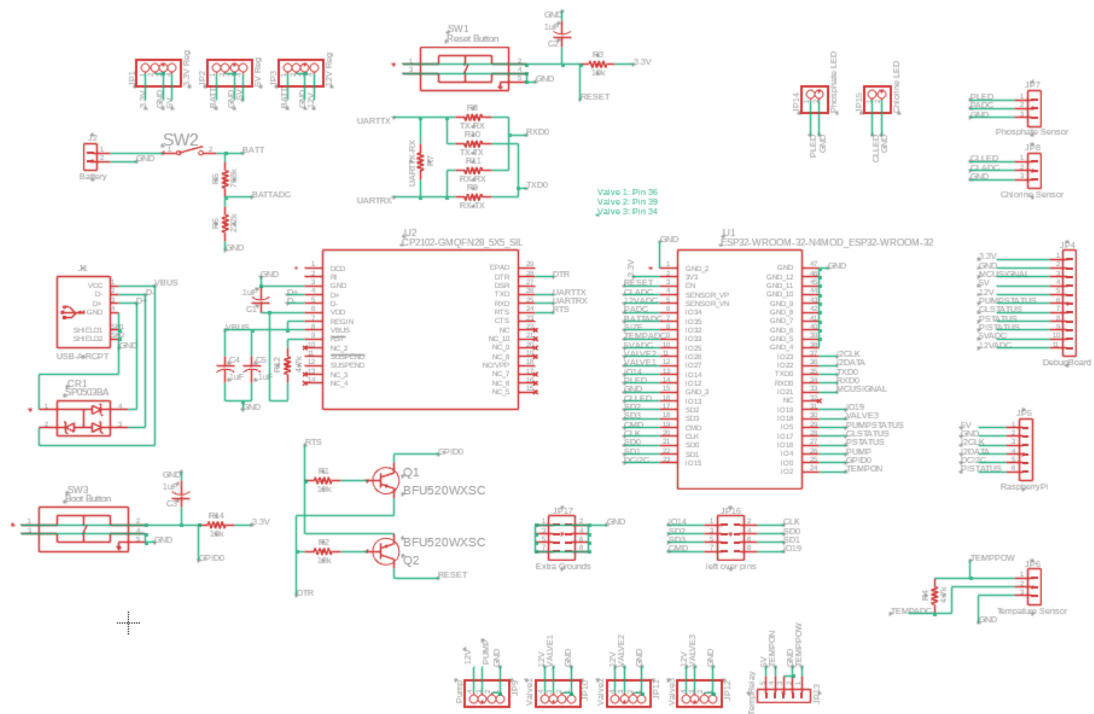


Figure 6.17: Final Main Board

6.8 Power Subsystem

The power delivery system utilizes two switching regulators to reliably provide 12V, 5V, and 3.3V rails for all subsystems. The initial design and component selection for these regulators were carried out using TI's WEBENCH Power Designer, along with the corresponding bill of materials (BOM). The LDO was selected based on Texas Instruments datasheets and resources available through their Power Management website. After filtering available options to meet system-specific requirements, a suitable LDO was chosen. The final schematic, which uses both the switching regulators and the LDO, was developed using Autodesk Eagle.

Switching regulators were selected primarily due to the anticipated heavy use of Wi-Fi, which can generate significant heat and place high power demands on the ESP32. The original plan was to use the onboard LDO in the ESP32 CP2102 module by supplying 5 V through the VIN pin (rated for 5 V input), which would then regulate down to the required 3.3 V for the ESP32 chip. However, as the design for the system evolved and additional components were introduced, it became clear that the onboard LDO might not offer sufficient efficiency or thermal performance. To prevent overheating and ensure reliable operation, the onboard LDO was bypassed, and a clean, regulated 3.3 V supply was used to power the ESP32 directly.

The 12V rail from the rechargeable battery is currently unregulated, as the valves and pump are capable of operating reliably without a tightly regulated supply. This approach conserves PCB space and may also help reduce the pump's speed, which is desirable given that it operates without a motor driver. The photodiode in the optical system has not yet been fully tested yet, so its exact voltage and current requirements still need to be determined, as another power rail would design implementation more difficult if it needs a much stronger power rail than 12V. As a result, a buck-boost regulator has not been implemented at this stage just yet. Once photodiode requirements are confirmed and during prototype testing, the system's performance under varying load conditions has been evaluated. A buck-boost regulator or other components may then be added if necessary to ensure voltage stability across all subsystems.

The three regulator circuits shown in the next section are not soldered directly onto the main PCB. Instead, they are assembled on separate breakout boards to facilitate easier troubleshooting. These breakout boards are connected to the main PCB using male pin headers, which are soldered in place after the regulators have been fully tested.

6.8.1 Regulator Schematics

The schematic for the 12V switching regulator using the TPS621361 IC is shown on the top in **Fig. 6.18** below. The 14V battery is bucked down to a regulated 12V in order to supply the photodiodes, water pump, and solenoid valves that require 12V. The regulator's total BOM was $1.51 and an efficiency of 94.5%.

The schematic for the 5V switching regulator using the TPS563200 IC is shown on the top in **Fig. 6.19** below. The capacitors and resistors used were those suggested by WeBench and the efficiency using the design suggested came out to be about 93.5% efficient. The total BOM cost was $0.95 which was well within our budget.

The TPS563200 is used as a second switching regulator to generate a dedicated 5 V rail, which is then be stepped down to 3.3 V using an LDO. Using an LDO to drop directly from 12 V to 3.3 V would result in excessive heat generation, and while using a second switching regulator to directly produce 3.3 V was initially considered, it may not provide a sufficiently clean output for sensitive components. Although it would have been possible to tap 3.3 V from the existing 5 V rail, that rail is powering several peripherals, and the combined current draw could compromise voltage stability. Since the MCU, particularly its ADC module, requires a clean and stable supply, it was determined that creating a separate 5 V rail for the LDO was a more reliable solution. The final reason for using the TPS563200 was its low BOM cost and the ability to reuse components, as having duplicates of the same regulator simplified the design. This cuts down on troubleshooting later when the regulator boards are built.

Fig. 6.20 shows the LDO regulator with the pin attachments for the PCB board.

Fig 6.21 shows the two 5V regulators branching off of a 14V battery, then one of the 5V regulators has a 3.3V regulator branching off of it as an output. The 3.3V output from the regulator is connected to the 3.3V pin of the ESP32 CP2102 development board to supply the voltage it requires. The pins for attaching to the PCB are located on the left side and are soldered in place during assembly. Additionally, a 12V output branches off from the battery, where a buck-boost regulator may be added later if certain peripherals, such as the photodiode for the optical system, are confirmed to operate reliably at 12V. This needs to be verified before finalizing the regulator. Once confirmed, the buck-boost regulator is added in that area, similar to the placement of the 3.3V and 5V regulators.

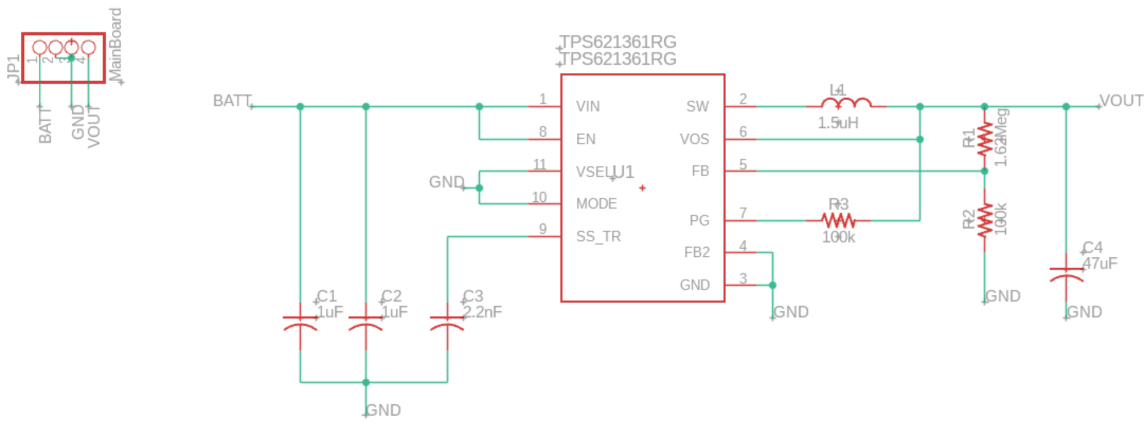


Fig 6.18: 12V Regulator Schematic

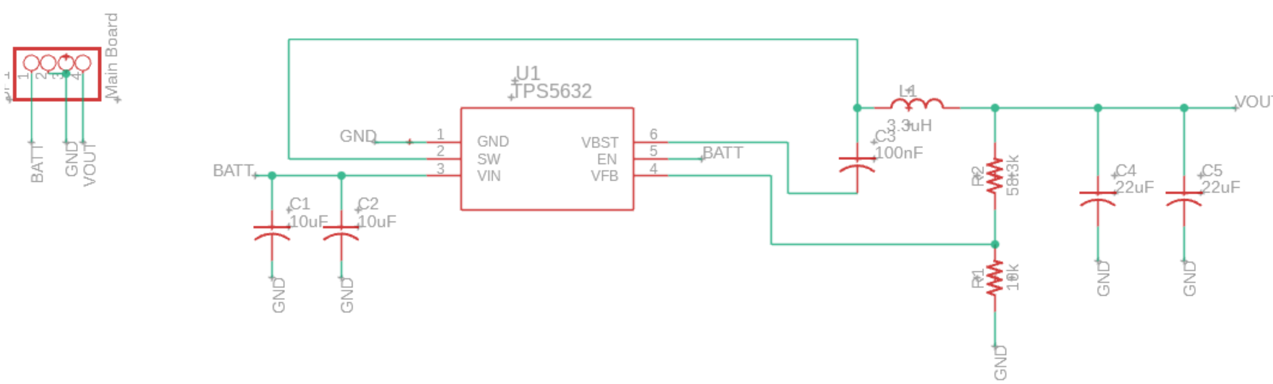


Figure 6.19: 5V Regulator Schematic

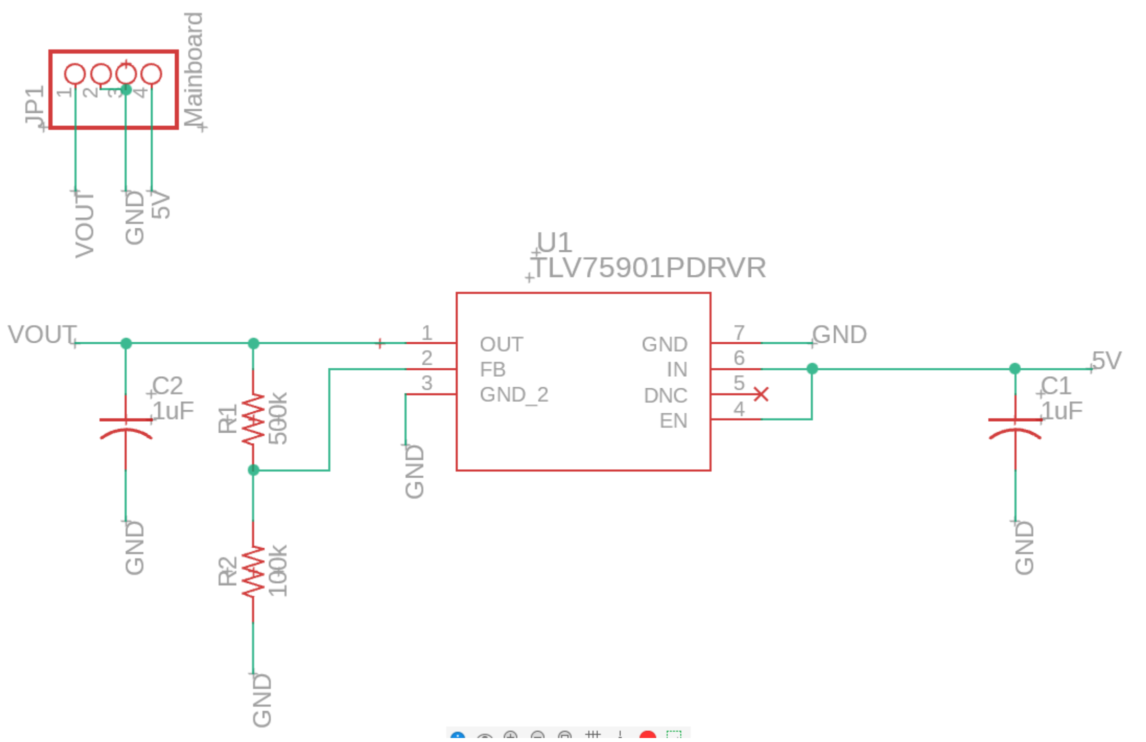


Figure 6.20: 3.3V Regulator Schematic

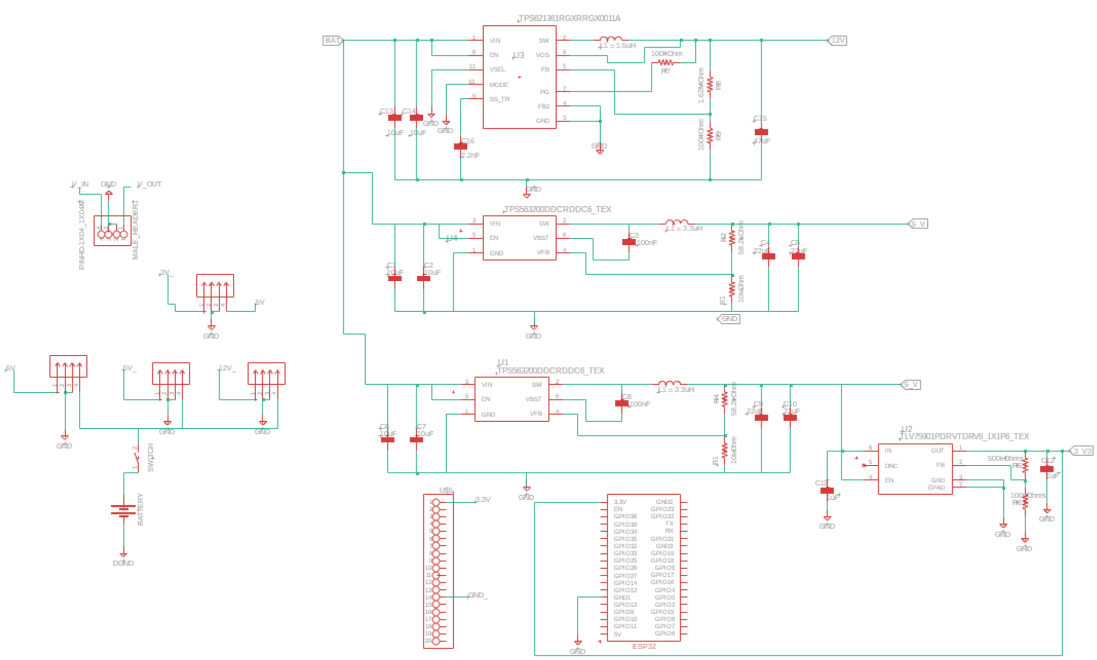


Figure 6.21: Regulators to MCU Connection

After manufacturing those regulators mentioned above, we realized that the 3.3V regulator and 5V regulator would not suffice to power our system so we changed them to the designs shown in **Figure 6.22** and **Figure 6.23**.

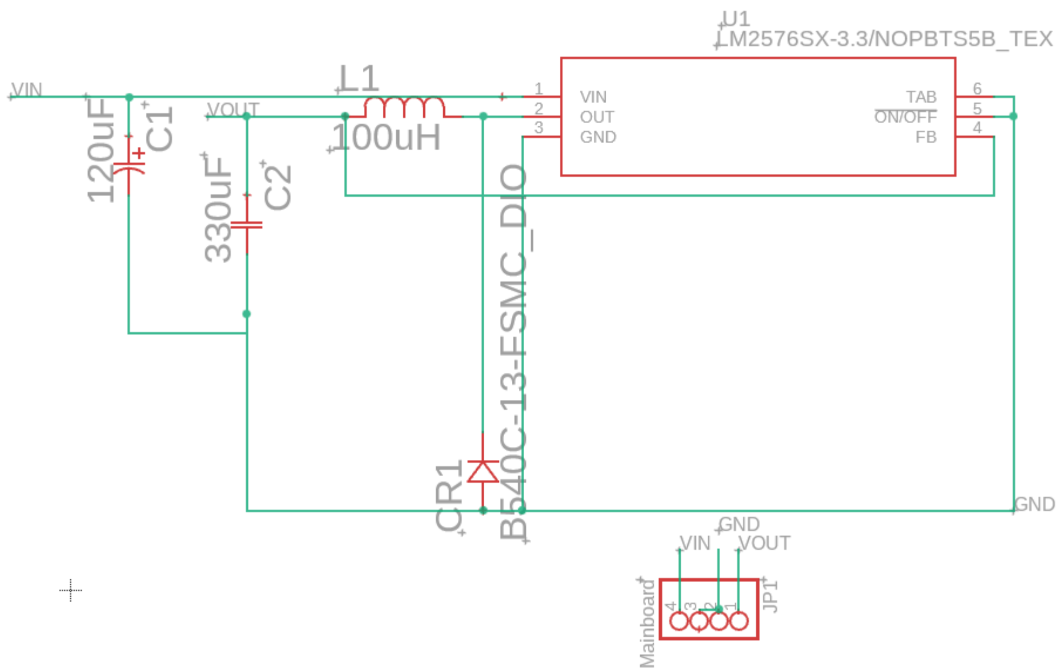


Figure 6.22: Final 3.3V Regulator

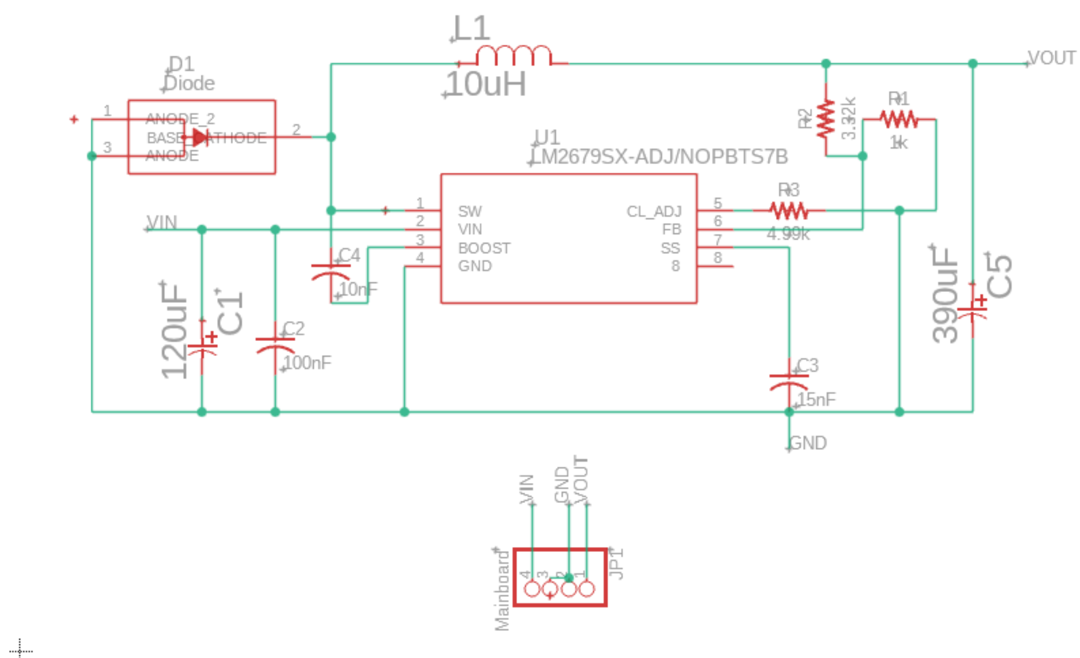


Figure 6.23: Final 5V Regulator

6.8.2 Power Requirements

Table 6.1, provides a running estimate of the overall power requirements and distribution for the system. While these values are initial estimates and may change based on the final design, the table offers a useful snapshot for planning power usage based on selected components and research-based estimates so far. An engineering requirement was to design a power system with a target consumption of under 20 W. Most components, particularly the pump and valves, operate in short bursts, so the average power consumption is expected to remain within acceptable limits during normal operation.

Table 6.1: Power Requirements

	Voltage (V)	Current(mA)	Power
MCU	3.3V	250 mA	0.825W
Raspberry 4	5.2 V	250mA/400mA(startup)	2.08W
TLV3541IDBVR Op-Amp(x4)	+/-5 V	5mA	0.05W
150080BS75000 LEDs(x8)	3.3V	0.2mA	0.00066 W
WP7113ZGC/XT HI30W LEDs	3.3V	20mA	0.66W
Water Pump	12V	223mA	2.676W
Solenoid Valves (x2)	12V	500mA	6.60W
Brushless Fan (x2)	5V	80mA	0.40W
Relay Module(x2)	12V	40mA	0.48W
Relay Module	5.2V	5mA	0.026W
Temperature Sensor	5.2V	20mA	0. 104W
Total			~13.3W

6.9 PoolWatch Overall Schematic

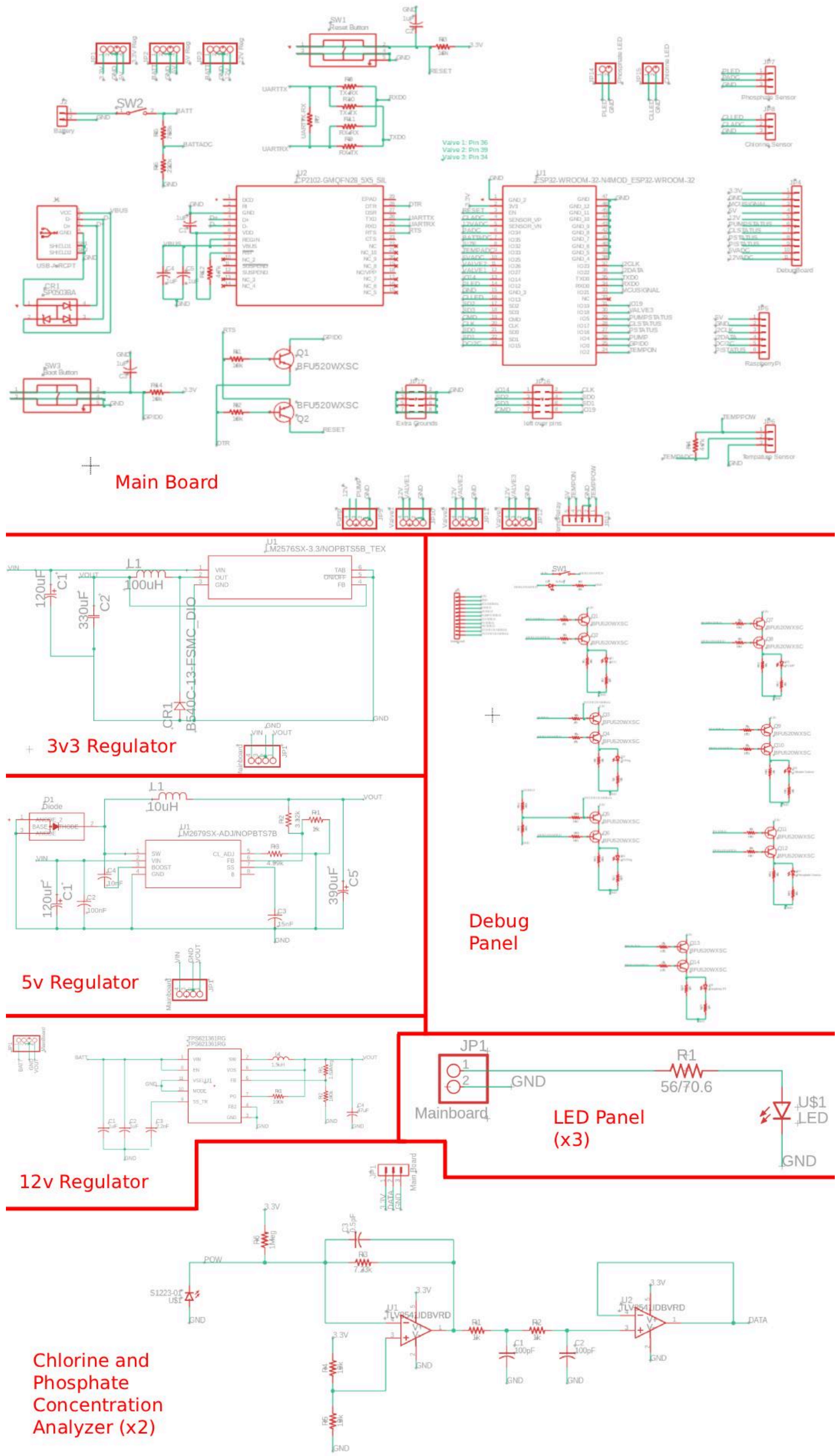


Figure 6.24: Total Schematic

Chapter 7 Software Design

7.1 Pump, Valves, and Optical Subsystems Control

This section describes the software logic used to manage the system's operations. The ESP32 microcontroller acts as the central controller and communicates wirelessly via Wi-Fi to the cloud and connected app. The MCU activates specific channels on a 12V four-channel relay module to start and stop the water pump and two solenoid valves as needed. Timing sequences and logic conditions are used by the MCU to coordinate water intake, reagent dispensing, and when the optical subsystems are activated for analysis. In addition, the MCU ensures that power is efficiently managed by enabling system components, like the two optical systems, only when required and disabling them during idle periods. **Figure 7.1** below shows the flowchart of the overall system described.

For programming the microcontroller, we used the Arduino IDE. This platform was chosen for its extensive library support, which makes rapid prototyping more accessible and efficient. The code is written in the C programming language and implements the main control logic. This allows the system to execute scheduled tests and also run individual tests on demand, either through the app using WiFi or manually via the control box.

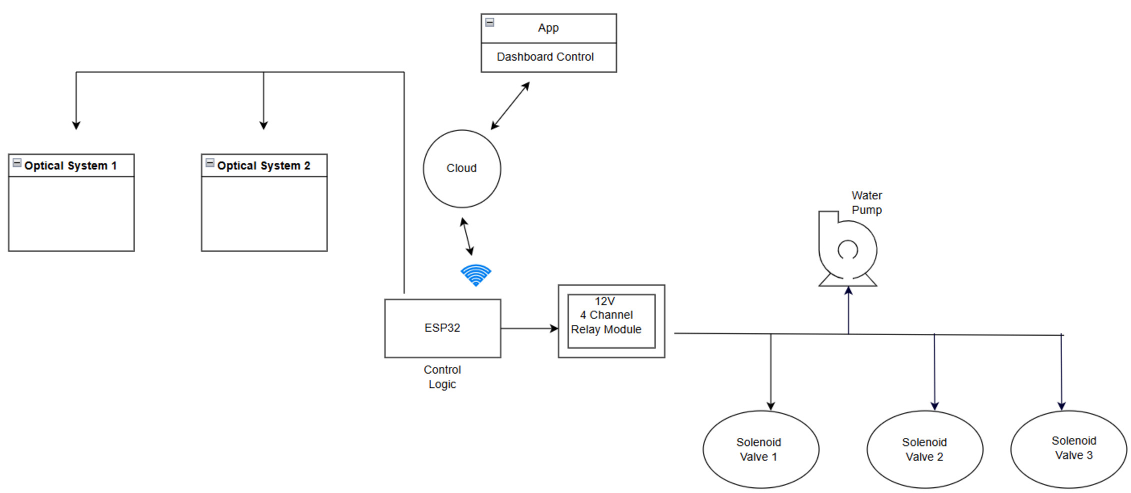


Figure 7.1: ESP32-Based Control System Flowchart

7.1.1 Fluid Handling Subsystem

The water pump and solenoid valves are connected to a multi-channel relay module, with one relay channel assigned to each component. Since the system includes four components, a four-channel relay module is used to accommodate all the necessary connections. The relays control the on/off states of the valves and pump based on control signals sent by the MCU. Upon system startup, the MCU activates the relay to power the water pump. Once turned on, the pump draws water from the pool to initiate the testing process. After sufficient water is collected, determined by a predefined flow rate and time setting, the MCU turns off the pump and activates the solenoid valves via relay to dispense the required reagents. The reagents are dispensed briefly from the solenoid valves, which are then quickly shut off. The mixed fluids continue to flow through the piping system until they are deposited into the cuvettes. At this stage, the MCU engages the optical systems to begin the analysis process.

7.1.2 Optical Subsystem

This data is transmitted wirelessly to a cloud server via Wi-Fi, where it is stored and made accessible through a connected mobile application using an IoT interface. This setup enables real-time remote monitoring of the system, with results available on the app within minutes of data collection. The control of the optical systems does not use relays but via the configured GPIO pins.

7.2 Raspberry Pi Design

Since the demand of processing power as well as storage is high for image processing, a Raspberry Pi is used to provide the connection between the ESP32 and the camera used in the particulate imaging system. The ESP32 provides the control signals involving when to conduct the measurement to the Raspberry Pi in I2C format and the Raspberry Pi provides the control signals to the camera involving the camera settings using MIPI CSI-2 format. Once the picture is taken, the Raspberry Pi pulls out the key information needed from the image, involving light blocked due to particles as well as the size distribution of the particles. This information is then sent back to the ESP32 to be used by the Chlorine Measurement system as well as the user on the website. See **Figure 7.2** for the software flowchart.

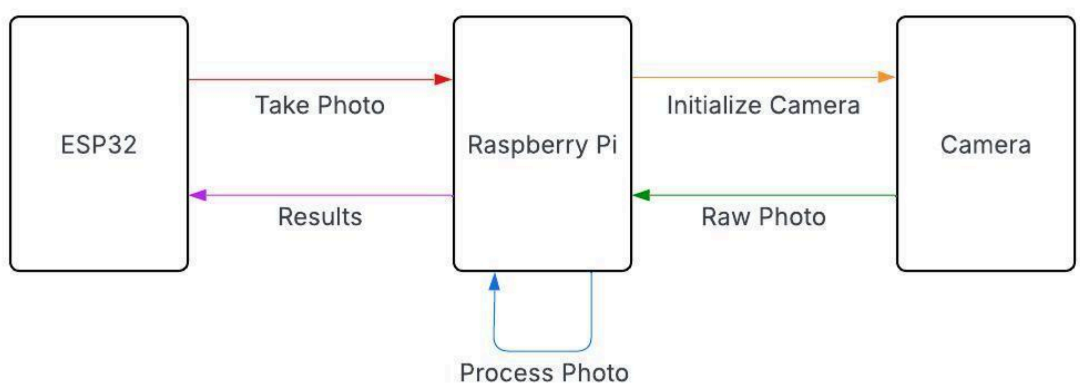


Figure 7.2: Software Flowchart for Raspberry Pi Camera System

7.3 Web Traffic design

The machine hosting the webserver also hosts several other applications to support the development and deployment of the website. **Figure 7.3** shows how the applications are structured on the host and how to access them.

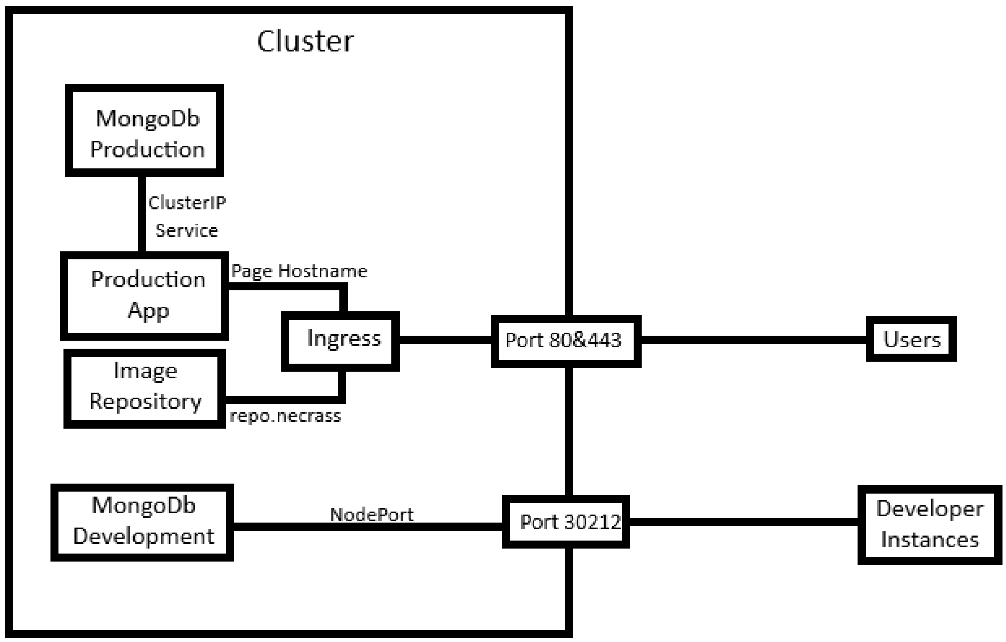


Figure 7.3: Web Interface

With the design shown in **Figure 7.3** the production database is hidden within the cluster, which means that the only way to connect to the database is from an app running within the cluster. The image repository is hidden behind the unique top level domain name necrass, which is not an existing top level domain. This means that any reference to that hostname is needed to be manually configured on the computer trying to access it, which allows the hostname to act as a password as the only people able to access the repo would be the ones that know the hostname. A separate database is provided for testing purposes. This database does not have any persistence and is completely vulnerable to attack, however the database can and will be wiped if any attacks happen on it.

7.4 Web Server Design

7.4.1 Backend Design

The backend of our server is responsible for device-server communications and website-server communications.

7.4.1.1 Account Creation/Login

To enable individual accounts on our website, the backend supports logging into and the creation of new accounts. Accounts are stored on a Mongo database, with the structure shown in **Table 7.1**

Table 7.1: Account Data Structure

Variable	Description
Id	Unique Id associated with account.
Username	Username of account.
Password	Password of account.
Devices	A list of device serial numbers associated with the account

The flow of account login and creation till the user reaches the website is shown in **Figure 7.4**.

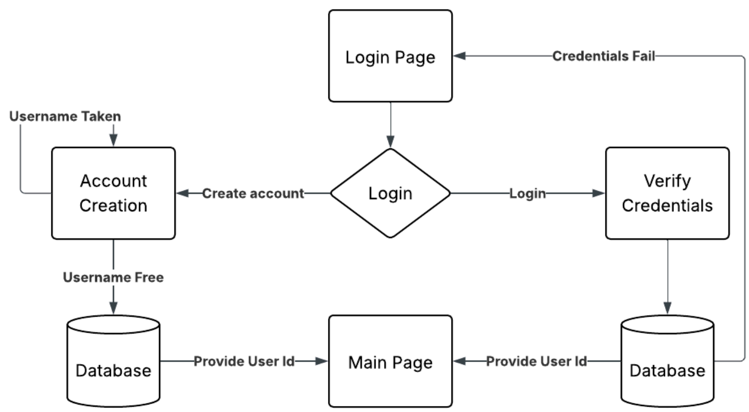


Figure 7.4: Account Creation Flowchart

7.4.1.2 Device Adding/Modification

Once within the main page the user is able to add and configure the devices associated with their account. The data structure of a device is shown in **Table 7.2**.

Table 7.2: Device Data Structure

Variable	Description
Serial Number	The serial number of the device
Battery Charge	Percent of battery remaining
Reports	A list of reports gathered by the device
Connected	A boolean to show if the device is connected
Pump Status	A boolean showing that the pump is functioning
5V Regulator Status	A boolean showing that the 5V regulator is functioning.

12V Regulator Status	A boolean showing that the 12V regulator is functioning.
Sample Rate	How often the system samples the pool.
Need Update	Boolean signifying an update is required of the device.
Test Chlorine	Boolean requesting chlorine test
Test Phosphate	Boolean requesting phosphate test
Test Temperature	Boolean requesting temperature test
Test Particulate	Boolean requesting particulate test
Update Servers	Servers to notify about pool changes

As we developed further the device data structure did not provide enough support for all the features we needed, so it was updated to **Table 7.3**.

Table 7.3: Final Device Data Structure

Variable	Description
Serial Number	The serial number of the device
Battery Charge	Percent of battery remaining
Reports	A list of reports gathered by the device
Connected	A boolean to show if the device is connected
Pump Status	A boolean showing that the pump is functioning
5V Regulator Status	A boolean showing that the 5V regulator is functioning.
12V Regulator Status	A boolean showing that the 12V regulator is functioning.
Sample Rate	How often the system samples the pool.
Need Update	Boolean signifying an update is required of the device.
Test Chlorine	Boolean requesting chlorine test
Test Phosphate	Boolean requesting phosphate test
Test Temperature	Boolean requesting temperature test

Test Particulate	Boolean requesting particulate test
Fill Water	Boolean requesting cuvettes be filled
Update Servers	Servers to notify about pool changes
Last Update	Last time the device was updated
Last Sample	Last time the device was sampled

Within the device data structure there is a list of reports. These reports are formatted in the structure shown in **Table 7.4**. The Device does not monitor the 3.3V regulator's status as the MCU fails if that regulator fails.

Table 7.4: Report Data Structure

Variable	Description
Timestamp	Timestamp of the report.
Chlorine Concentration	Concentration of chlorine within the pool at time of the report.
Phosphoric Acid Concentration	Concentration of phosphoric acid within the pool at time of the report.
Temperature	Temperature of pool at time of report.
Particulates Amount	Amount of particulates in pool
Particulate Size	Size of particulates in pool

Pictured in **Figure 7.5** is the flowchart of adding and modifying devices on an account.

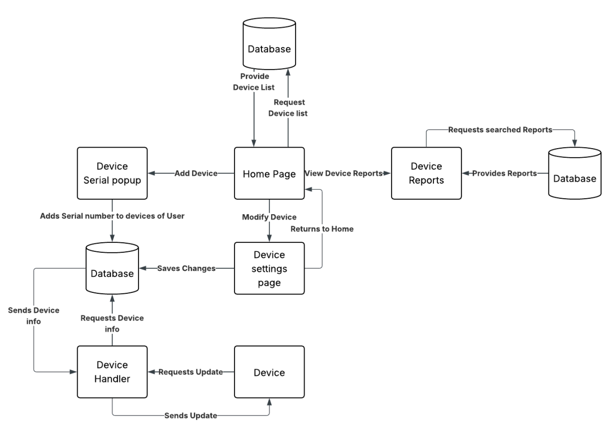


Figure 7.5: Device Management Flowchart

7.4.1.3 Report Uploading

The device communicates with HTTP json POST messages, which can be broken down into two different sections: the header and the body. The header section of our json HTTP message follows the format shown in **Table 7.5**.

Table 7.5: Device-Server Header Format

Variable	Description
Host	Hostname of the webserver
Content-Type	application/json
Content-Length	Length of the body message

The format of the body would be in json and it contains the entries shown in **Table 7.6**.

Table 7.6: Device-Server Report Body Format

Variable	Description
Chlorine Concentration	Concentration of chlorine within the pool at time of the report.
Phosphoric Acid Concentration	Concentration of phosphoric acid within the pool at time of the report.
Temperature	Temperature of pool at time of report.
Particulates Amount	Amount of particulates in pool
Particulate Size	Size of particulates in pool
Serial Number	The serial number of the device

The device does not attach a timestamp to the message as that is done when the data is added to the webserver. The flowchart shown in **Figure 7.6** illustrates the handling of reports.

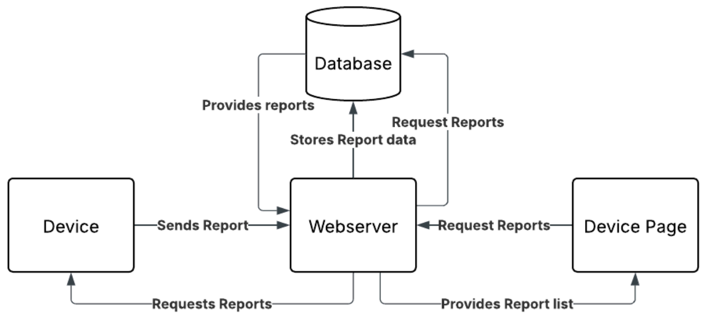


Figure 7.6: Report Webserver Flowchart

7.4.1.4 Device Statistic Handling

To provide information relating to the metadata of the device, such as its battery charge, the device communicates with the server using a HTTP json POST request. The header of this request is the same format as **Table 7.5**, but the body follows the format of **Table 7.7**.

Table 7.7: Device-Server Statistic Body Format

Variable	Description
Battery Charge	Percent of battery remaining
Pump Status	A boolean showing that the pump is functioning
5V Regulator Status	A boolean showing that the 5V regulator is functioning.
12V Regulator Status	A boolean showing that the 12V regulator is functioning.
Serial Number	The serial number of the device

The flowchart of the device metadata would follow the same path as shown in **Figure 7.7**, but instead of transferring reports it transfers statistics of the device.

7.4.1.4 Server Notifications

To allow for notifications of other servers when targets hit the data structure shown in **Table 7.8** is used.

Table 7.8: Notifier Structure

Variable	Description
Type	Type of Notification either email or plain text
Server	The location of the recipient of the notification
When	A string containing logic indicating when to notify

This occurs whenever a report is added to the server, and follows the flowchart shown in **Figure 7.7**.

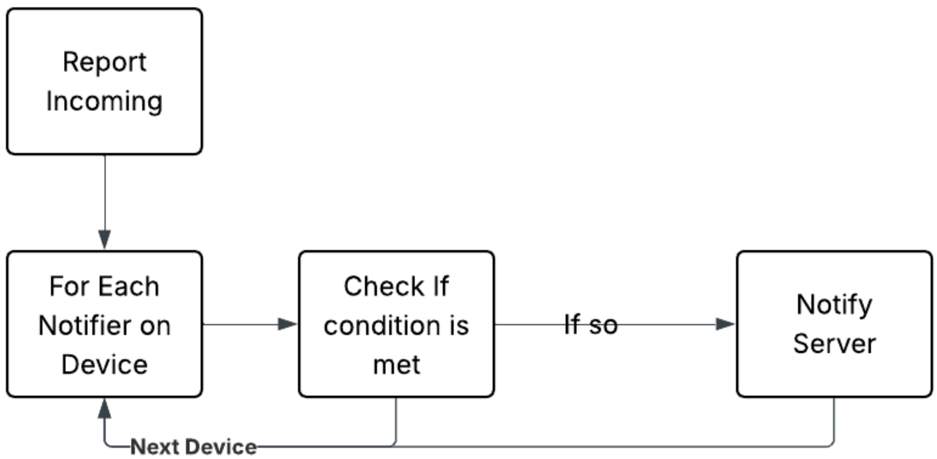


Figure 7.7: Notifying Flowchart

7.4.2 Frontend Design

The website design follows the flowchart shown in **Figure 7.8**.

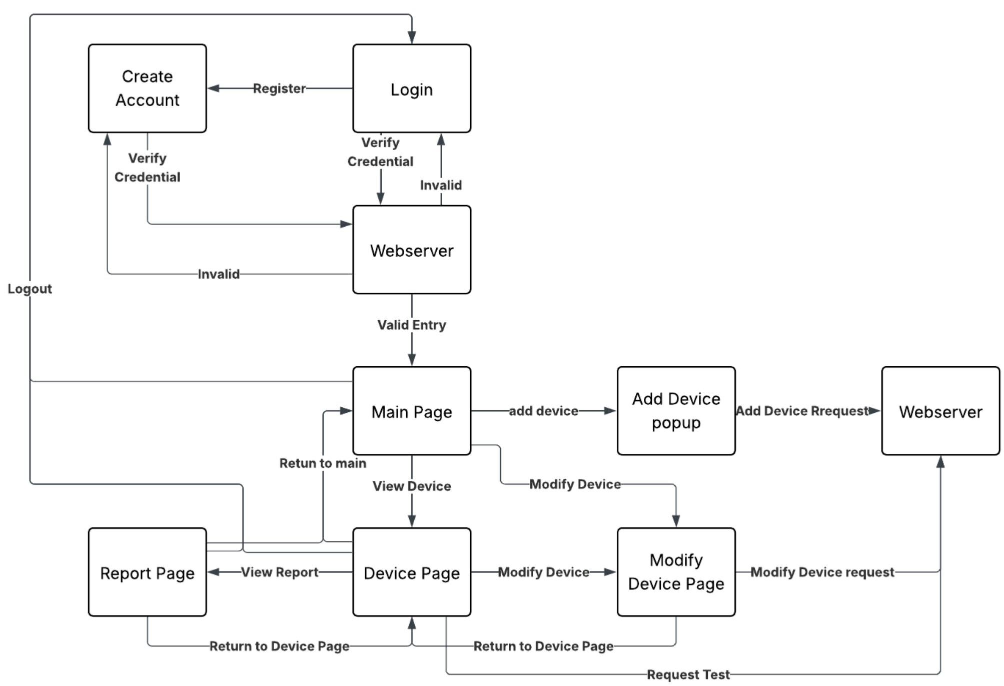


Figure 7.8: Website Flowchart

7.4.2.1 Login Page

The login page is a simple prompt for user credentials as shown in **Figure 7.9**.

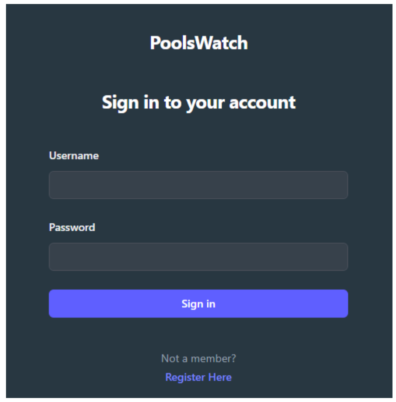


Figure 7.9: Login Page

7.4.2.2 Register Page

The register page is similar to the login page, just with some added prompts as shown in **Figure 7.10**

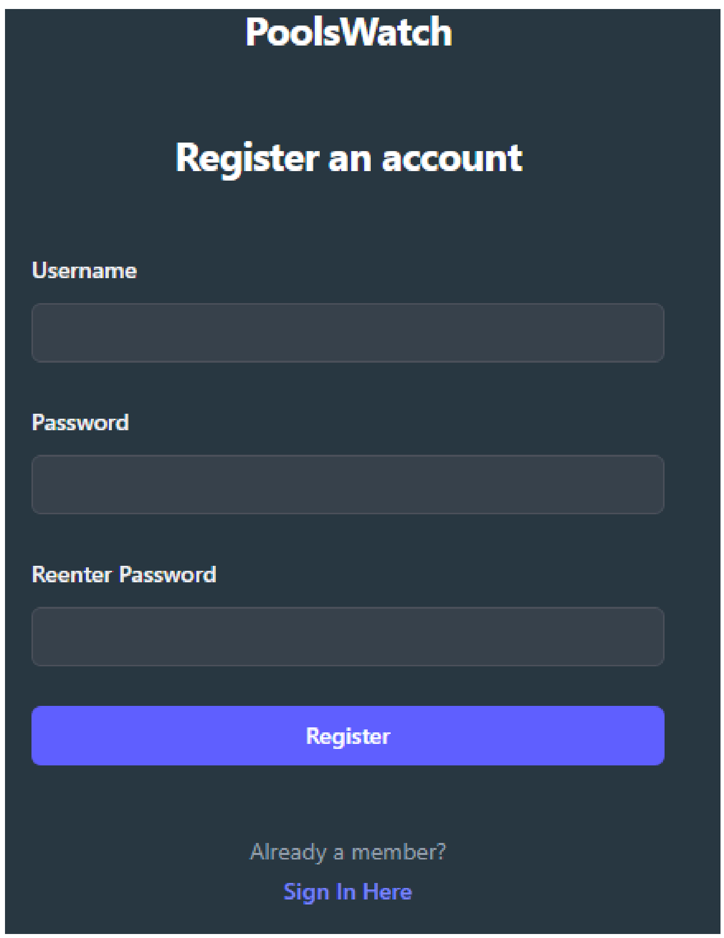


Figure 7.10: Register Page

7.4.2.3 Main Page

The Main Page contains a list of devices associated with the account as well as a way to add devices and to logout. The model can be seen in **Figure 7.11**.

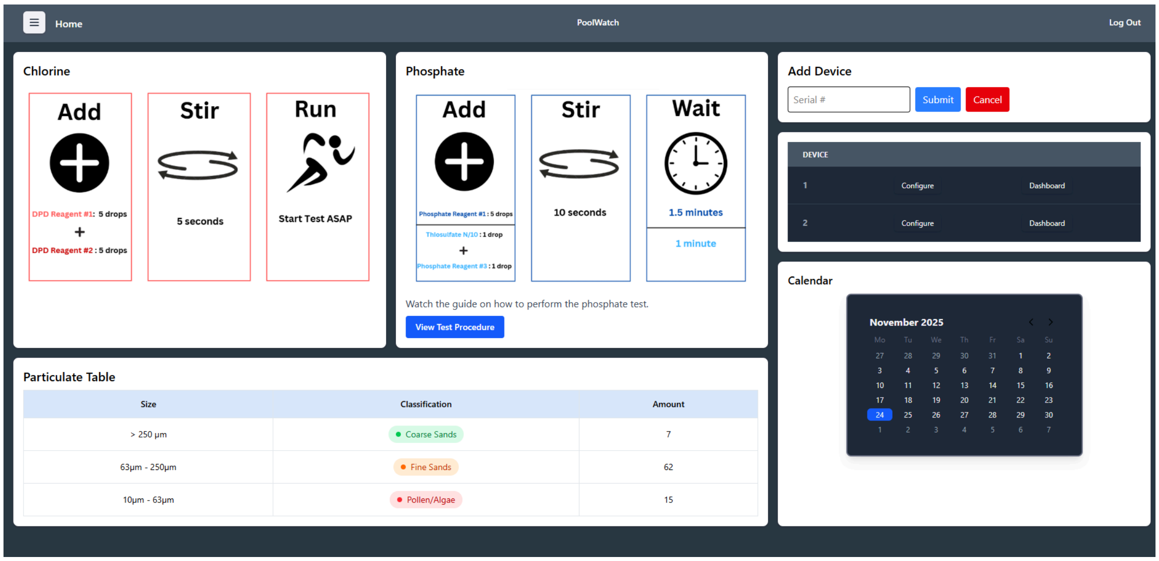


Figure 7.11: Main Page

7.4.2.4 Device Page

The Device page features test requesting, device statistics, and a list of reports. The page is laid out as shown in **Figure 7.12**.

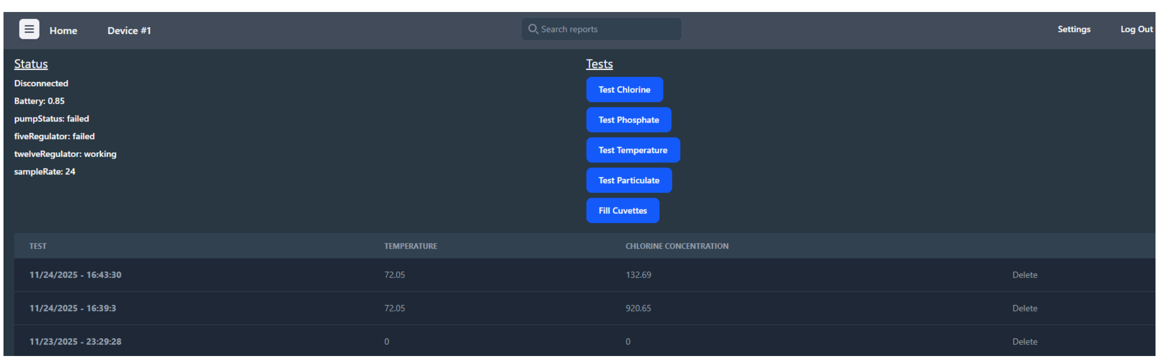


Figure 7.12: Device Page

7.4.2.5 Device Configuration Page

The device configuration page allows for the changing of variables and the setup of notification of a device. The layout is shown in **Figure 7.13**.

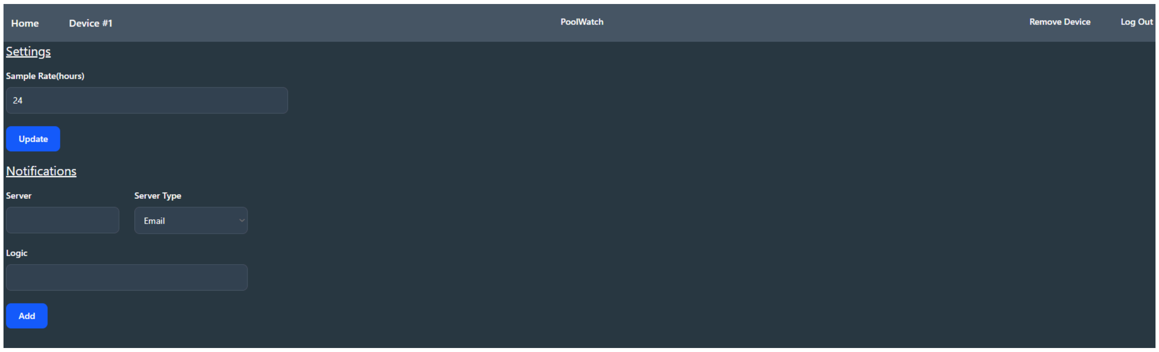


Figure 7.13: Device Configuration Page

7.4.2.6 Report Page

The Report Page shows the total results of a report and follows the layout shown in **Figure 7.14**.

Figure 7.14: Report Page

Chapter 8 Hardware Design

8.1 Main Board

The Main Board's purpose is to house the MCU and connections to peripherals, on the board there are the following features:

- The MCU
- Connections to Peripherals
- Battery Voltage ADC
- Reset Button
- Micro USB
- USB to Serial Converter
- Power Connections (Regulators and Batteries)
- Power Switch

Pictured in **Figure 8.1** is the mainboard's layout with the ground plate put in. The White squares represent the regulator boards taken area.

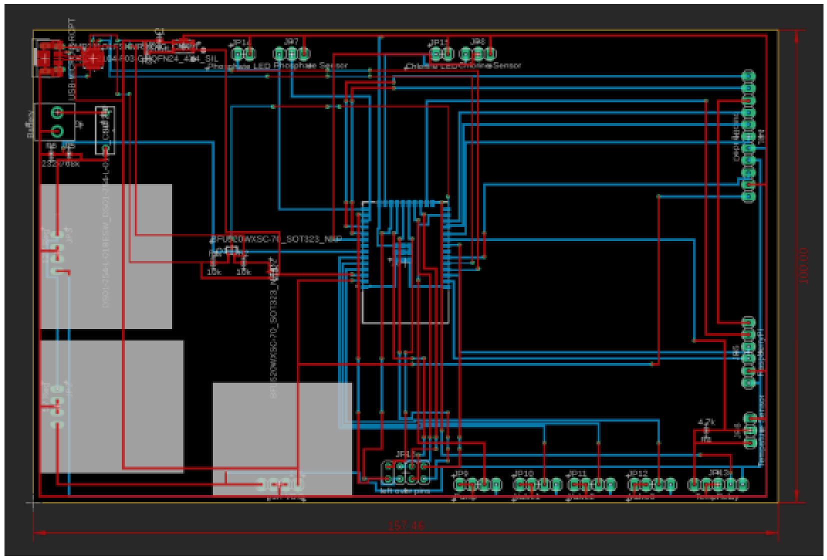


Figure 8.1: Main Board Layout

Since we changed the schematic **Figure 8.2** shows the final layout.

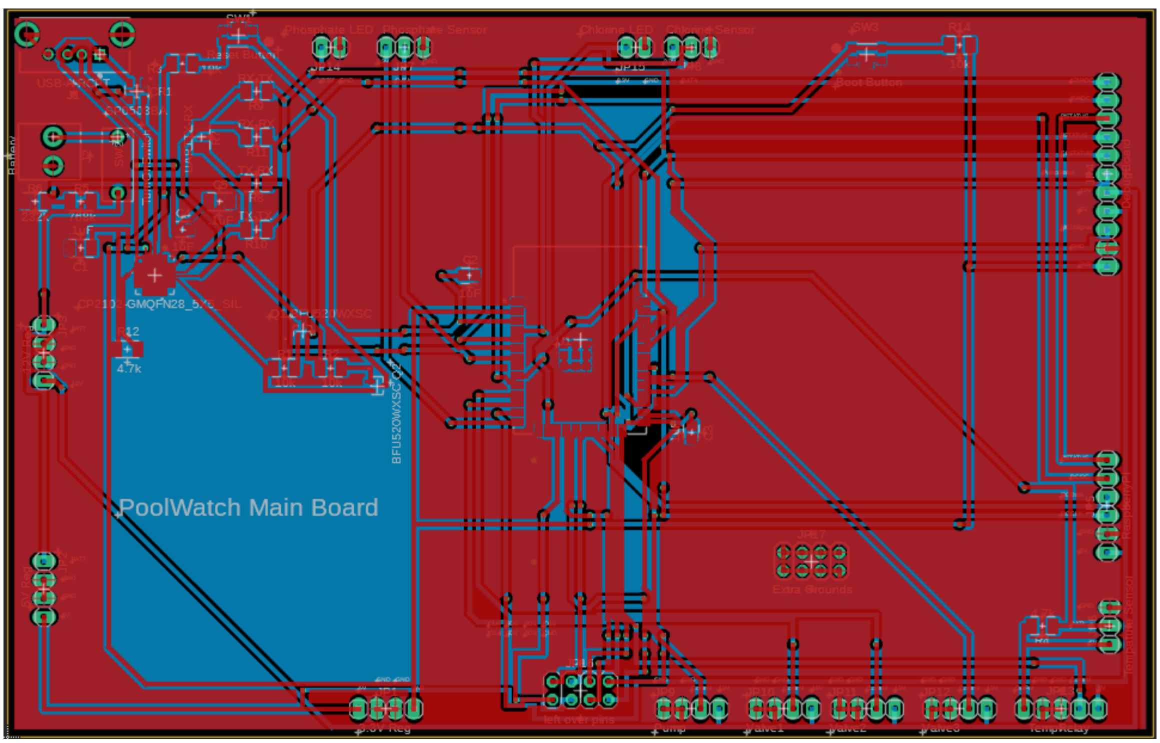


Figure 8.2: Final Main Board Layout

8.2 Chlorine Sensor

The Chlorine Sensor board is designed to hold the chlorine Sensor LED as well as the filter used on the board. Pictured in **Figure 8.3** is the board's layout.

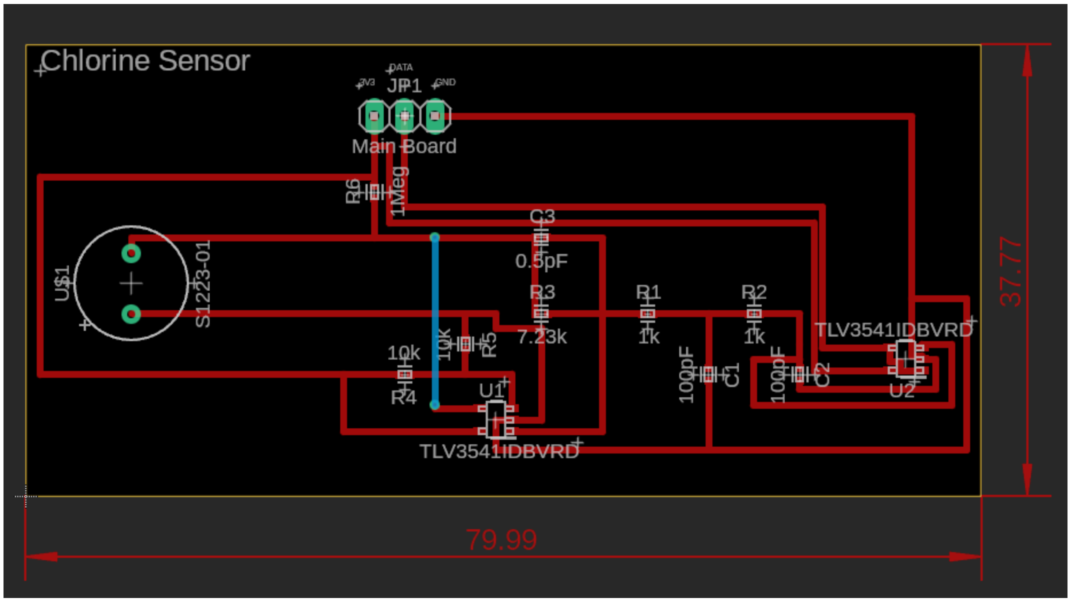


Figure 8.3: Chlorine Sensor Layout

8.3 Phosphate Sensor

The Phosphate Sensor board is designed to hold the phosphate Sensor LED as well as the filter used on the board. Pictured in **Figure 8.4** is the board's layout without the ground plate.

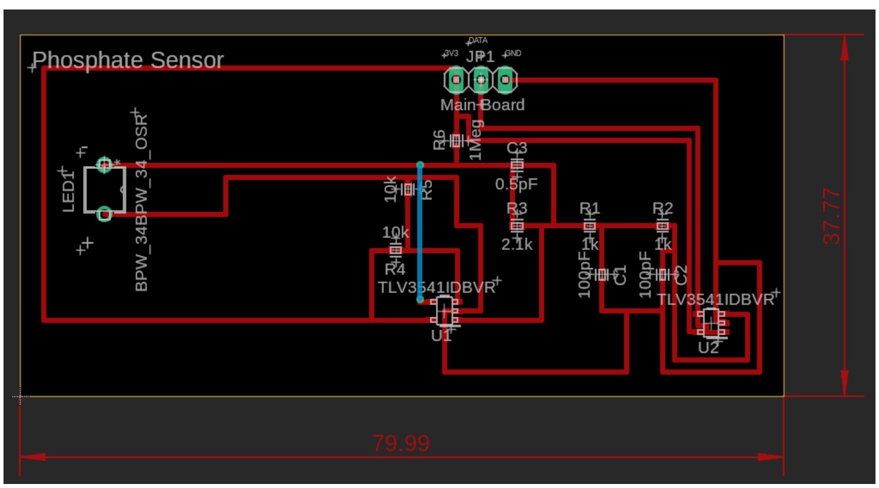


Figure 8.4: Phosphate Sensor Layout

8.4 LED Boards

The LED board is the board used for both the phosphate LED and the Chlorine LED, as they take the same component footprints. Pictured in **Figure 8.5** is this board's layout without the ground plate.

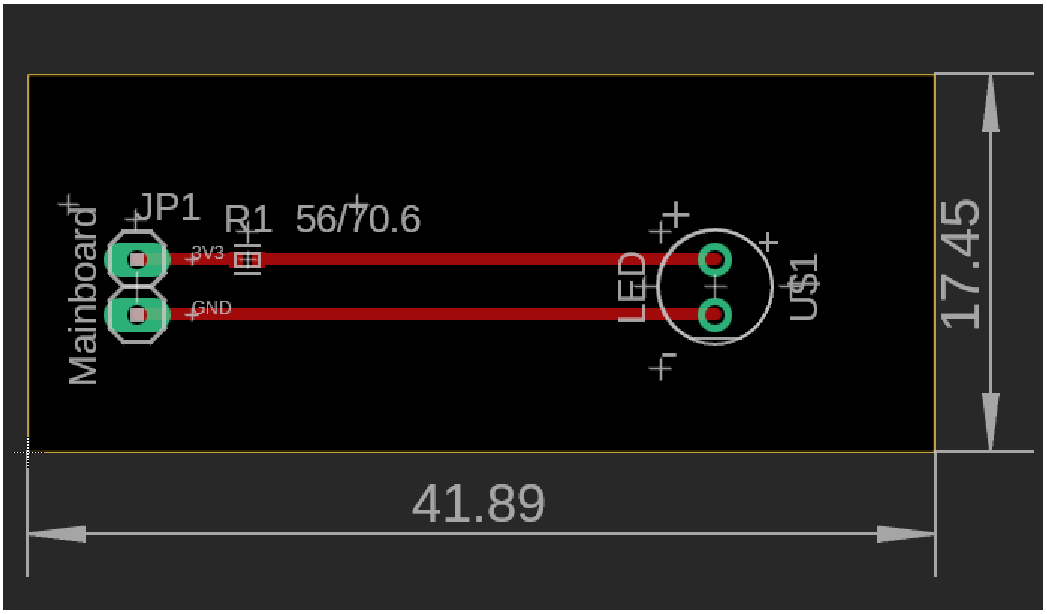


Figure 8.5: LED Board Layout

8.5 Debug Panel

The Debug Panel hosts several LEDs indicating the working status of important components of PoolWatch. The board also provides the ADC for the 12V regulator and the 5V regulator to indicate the working status of those components back to the MCU. The board also comes with a switch to turn it on and off. Pictured in **Figure 8.6** is this board's layout without the ground plate.

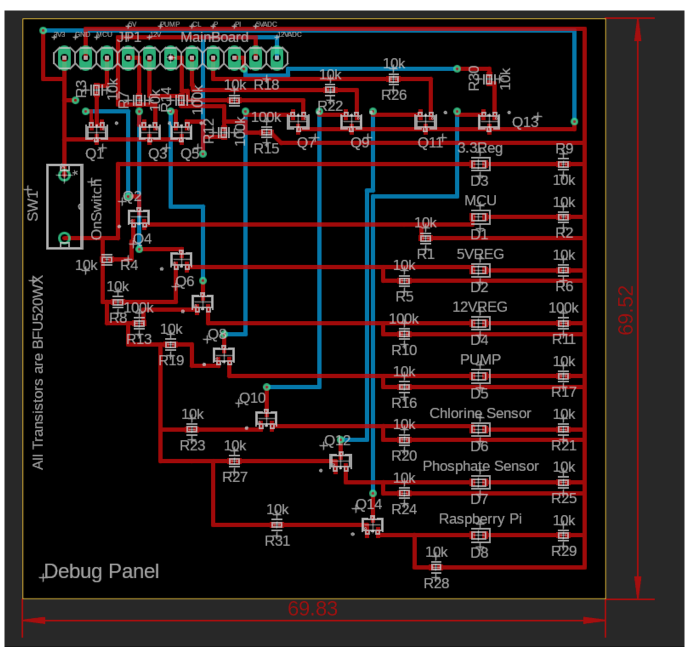


Figure 8.6: Debug Panel Board

8.6 12V Regulator

The 12V regulator board provides a 12Vs to the main board and is pictured in **Figure 8.7**, without the ground plate.

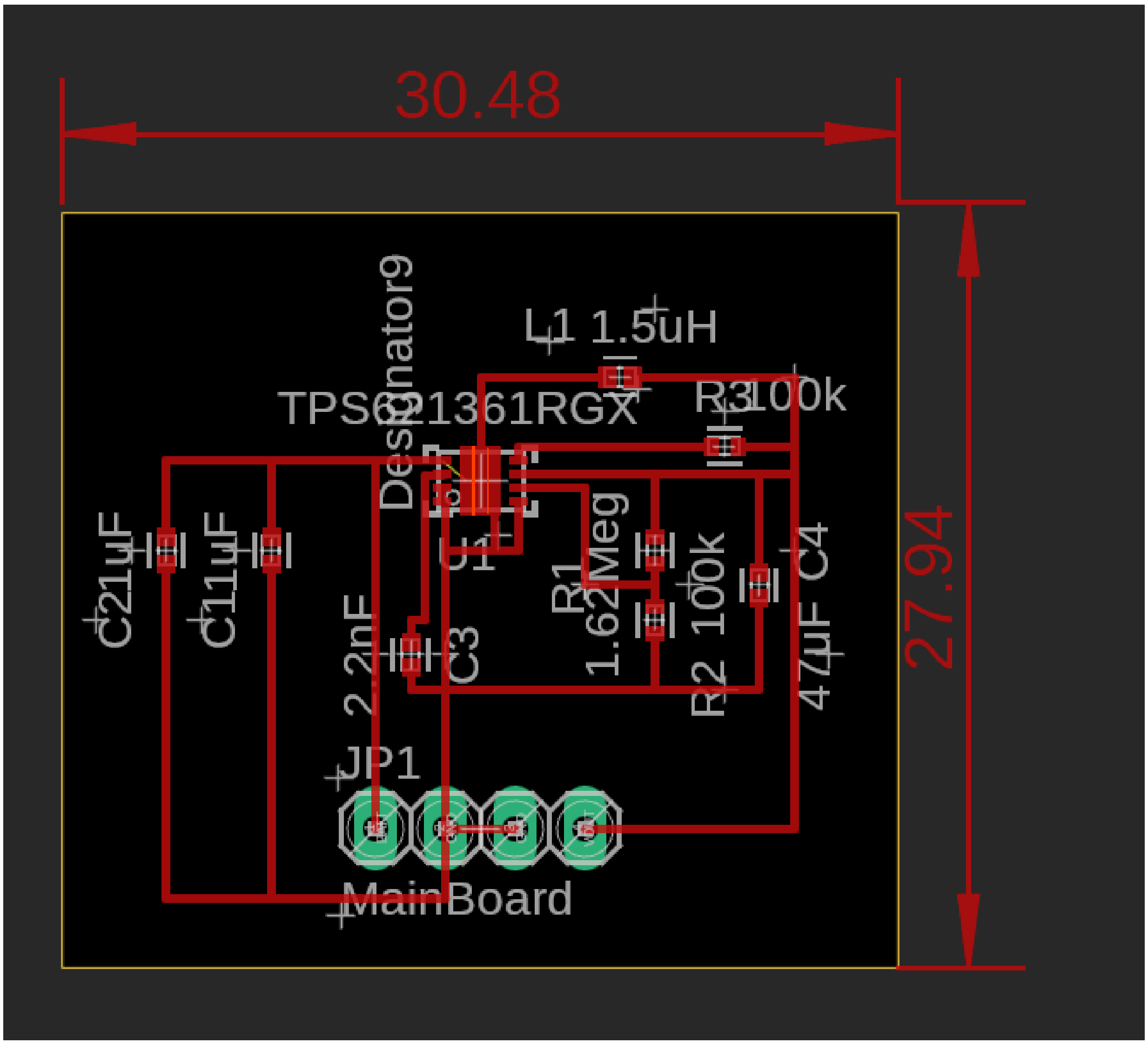


Figure 8.7: 12V Regulator Layout

8.7 5V Regulator

The 5V regulator board provides a 5.2V to the main board and is pictured in **Figure 8.8**, without the ground plate.

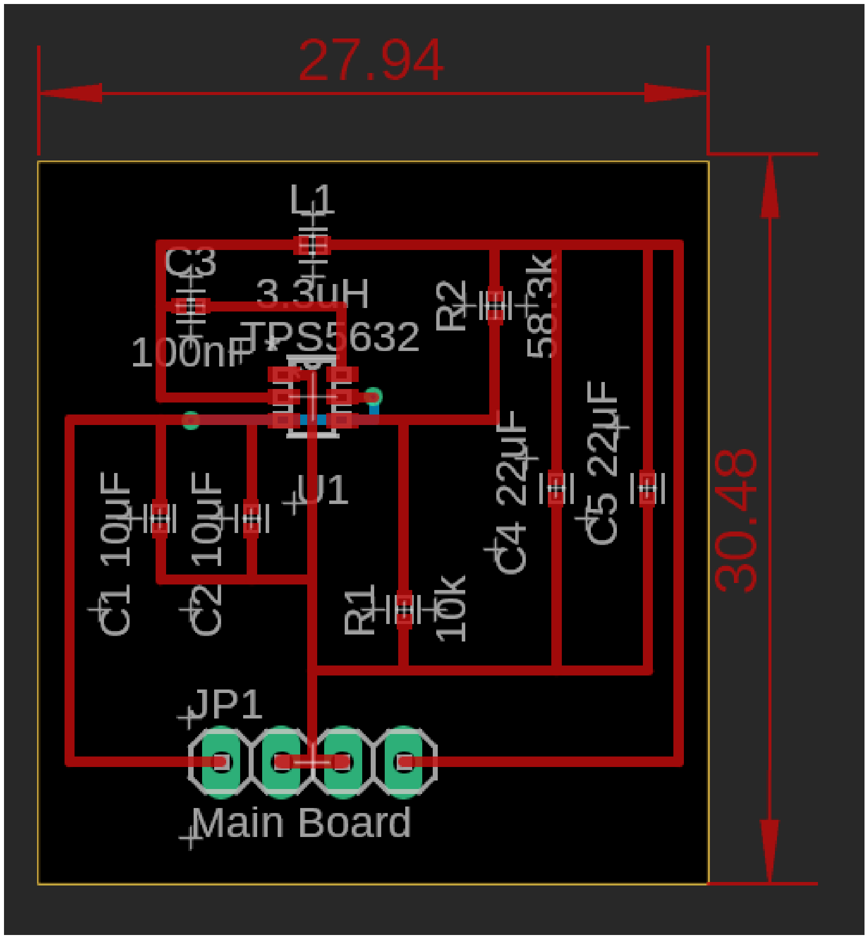


Figure 8.8: 5V Regulator Layout

With the changes we made to the schematic **Figure 8.9** is the final layout of the 5V regulator.

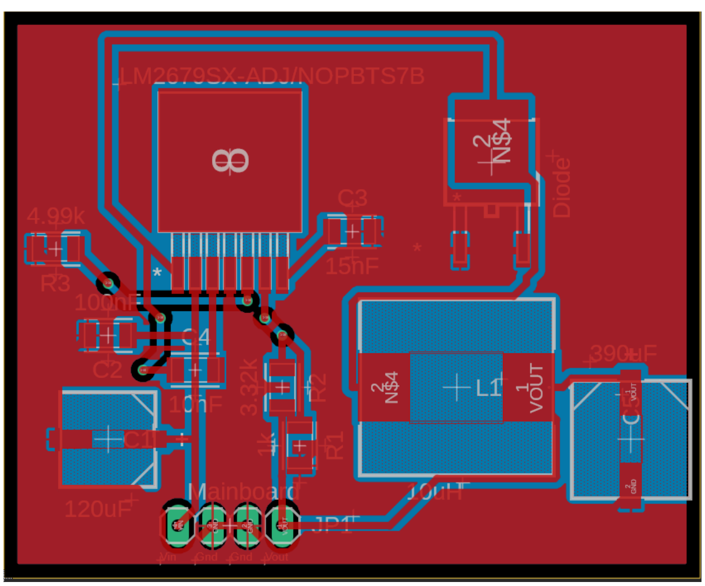


Figure 8.9: Final 5V Regulator Layout

8.8 3V3 Regulator

The 3V3 regulator board provides a 3V to the main board and is pictured in **Figure 8.10**, without the ground plate.

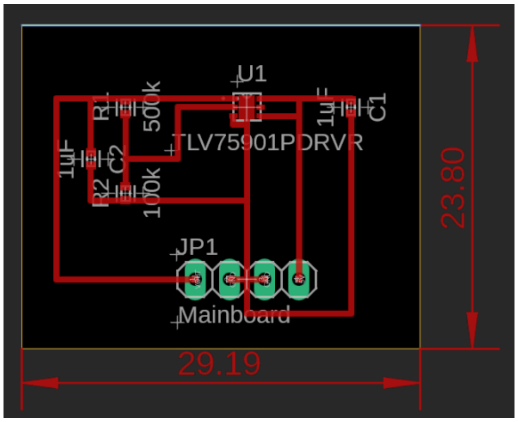


Figure 8.10: 3V3 Regulator Layout

With the changes made to the schematic **Figure 8.11** now represents the final layout.

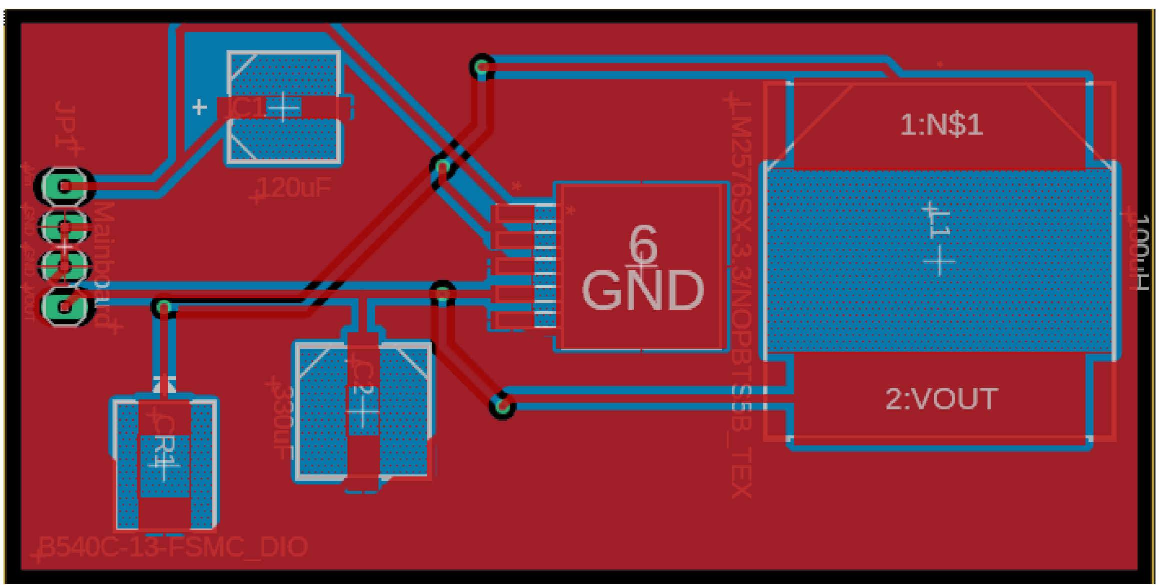


Figure 8.11: Final 3V3 Regulator Layout

Chapter 9 System Testing and Evaluation

9.1 Debug Panel

To test the BJT design for the debug panel in PoolWatch a simulation was performed using LTspice. In **Figure 9.1**, only one signal was set as high and the output was measured over the LED. The resulting voltage was 1pV, which is extremely close to zero and not enough to turn on the LED. **Figure 9.2** shows the output when both signals were high. The observed voltage over the LED and resistor was 2.41 V, which would allow current to flow through the LED and it would turn on.

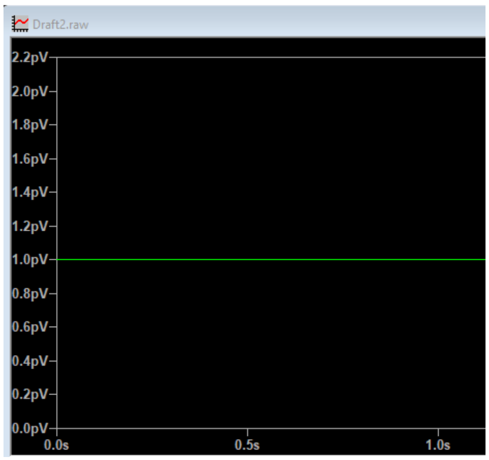


Figure 9.1: AND Gate with One High Simulation

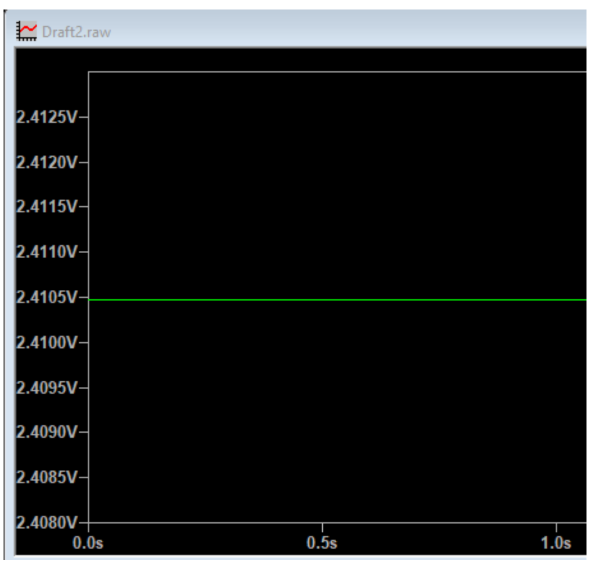


Figure 9.2: AND Gate with Both Signal High

To further verify the authenticity of this design a circuit was constructed on a bread board as shown in **Figure 9.3** and **Figure 9.4**. Two buttons were used to mimic the signal and as seen in the first figure the LED is off when one of the buttons is pressed, however in the second figure when both buttons are pressed the LED turns on.

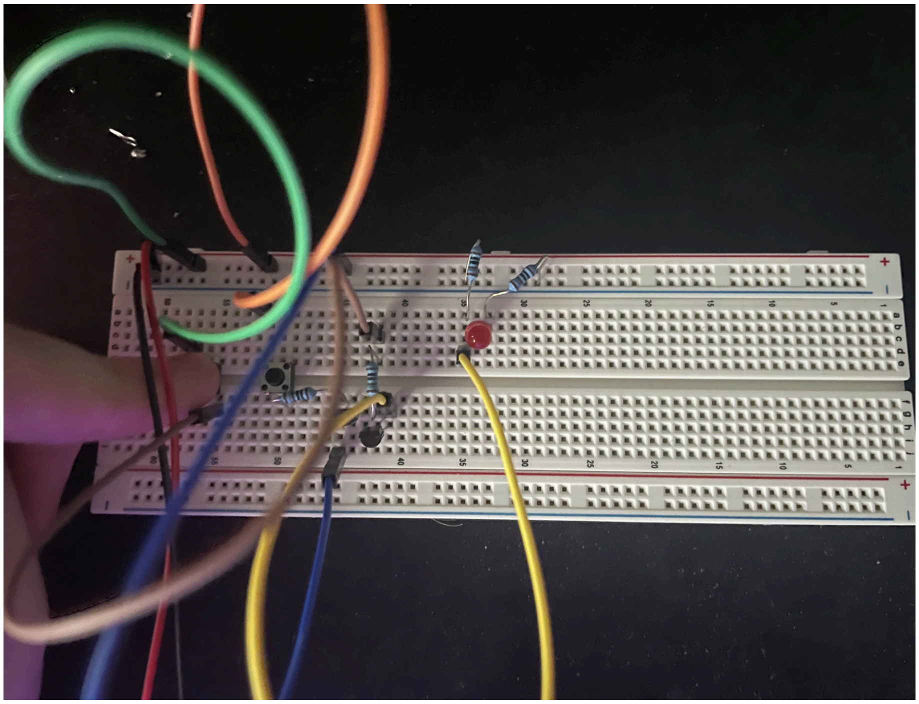


Figure 9.3: BJT AND Gate Single Signal

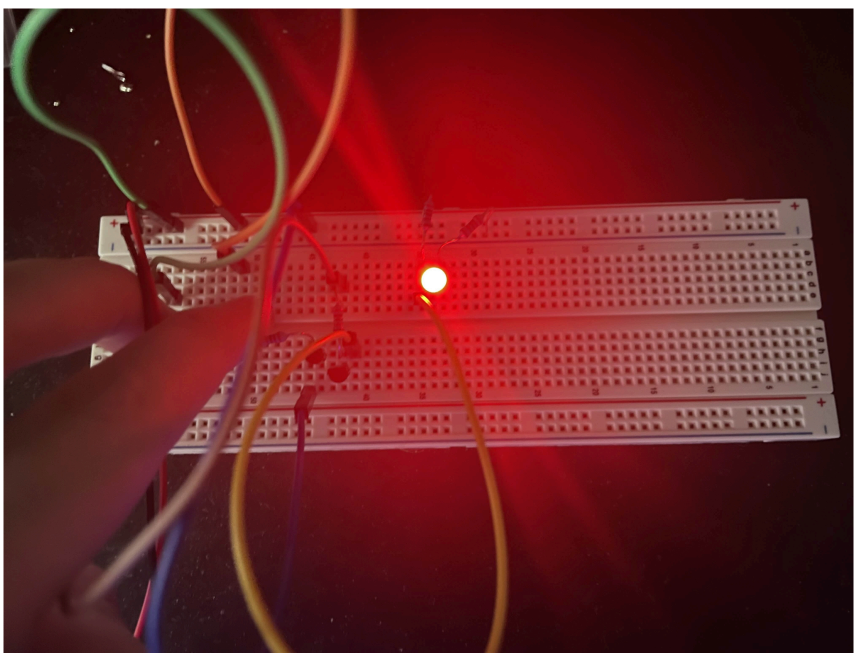


Figure 9.4: BJT AND Gate Both Signals

9.2 Optoelectronic Feasibility Study and Testing

In this study, the optoelectronics used in the midterm demo prove the components are ready to be integrated into the final design.

9.2.1 Light Emitting Diode (LED)

The diode that was used for testing to ensure the chlorine detection system was a 532 nm LED with a 220Ω resistor. The power supply provided 3.3 V which means 15mA of current was driven to the LED from the resistor in the circuit.

Table 9.1: LED Study and Test

Measurements	Optical Power
Noise Level	34 nW
Initial LED optical power	Po = 58 µW ±1%
DPD reagent with filtered water	Po = 57.4 µW ±1%
DPD reagent w pool water	Po = 26.9 µW ±1%

To prove the system would work as the initial design suggests a demo was set up and data was collected in **Table 9.1** and recorded with a small amount of uncertainty. Instead of using a photodiode an optical power meter was used, and was tuned to 532 nm to emulate what is the main function to measure output intensity. The beam was collimated and focused using a plano-convex lens. Before inserting the cuvette the box hosted the components were covered in the dark to determine what the noise level would be. The time average was set to three seconds, so there was no need to be concerned whether the measured value on the optical power meter fluctuated or was not accurate.

After the initial LED power was measured, the cuvette was placed in the middle of the aligned optical system. If no DPD reagent was added, then there was not any noticeable change in the output optical power. However, once about 0.5 mL of DPD was added into the 3.0 mL of filtered water the external appearance did not change, but there was a slight decrease in output intensity. This was also done with a kindly donated jar, which hosted pool water. Similar to the previous sample the same volume measurements were performed. In contrast, the pool water with chlorine quickly turned to a magenta color after mixing for about 12 seconds. Once the cuvette was placed into the aligned beam path the output intensity decreased around 46% which proved Beer-Lambert Law was the main mechanism for this change in intensity. This was how the absorbance from the cuvette would be able to be measured and further used in a calibration curve.

9.3 Water Pump Testing

To verify the water pump's functionality, a 12 V relay module was connected to a 12 V power source, with the relay used to control the pump's off/on operation. Using the Arduino IDE, code was written to toggle the relay on and off every five seconds, which effectively toggled the pump on and off. This was the first test involving the relay, with similar tests planned for solenoid valves. The wiring connections for these tests will be the same, and further testing was conducted to confirm the proper sequence for the final system layout.

The pump and other actuators to follow are connected to the relay via the switching side of the relay at the COM and NO terminals. The positive 12 V from the regulator connects to COM, and NO connects to the pump's positive terminal, which includes a diode across its terminals to lessen EMF when switching. The pump's negative terminal is connected to the control side, DC+ is connected to 12 V, DC− to ground, and IN to GPIO4 on the ground of the 12 V regulator.

The relay is set to high trigger mode, so a high input voltage activates it. For this relay, voltages under 10.5V will trigger switching. The ESP32 is powered via USB currently during testing, which gets regulated down to 3.3 V internally. The green LED indicates module power, while the red LED signals which channel is active. Each actuator is controlled by a dedicated GPIO, with four GPIO pins reserved for the valves and pump.

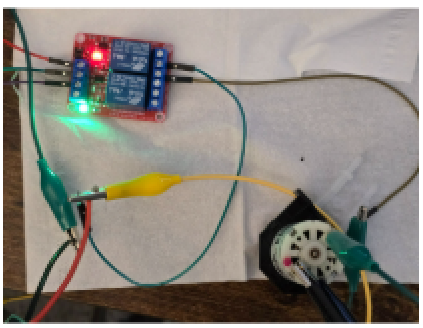


Fig. 9.5 Water Pump and Relay Test with LEDs On

As we assembled the final project, we realized that fully automating the reagent dispensing and mixing processes would not be feasible within the final design due to both time and space constraints. Since these features were considered stretch goals, we decided to exclude them in order to focus on delivering a complete and viable final product.

9.4 Web Testing

The web code of PoolWatch is stored within the github repository referenced in **[119]**. In that repository the code is split up into the folders src, test, webserverDrivers, devInfastructure, and PoolWatch. The src folder contains the frontend and backend of the webserver split into the client and server folders respectively. The test folder contains a device mimic program, which mimics messages the device might send to the webserver, the webserverDrivers folder contains the source code for the device to server interactions as well as an example for testing. The devInfastructure folder is a helm chart for supporting software for developers. The PoolWatch folder is the helm chart dedicated to the production deployment of PoolWatch. For the tests run within this sub chapter, everything was deployed on a local cluster named skadi. The version of PoolWatch used was 0.1.0, which is a function only model.

9.4.1 Infrastructure Tests

There are several supporting infrastructures related to pool watch, which are Podman, the image repository, the RKE2 cluster, the ingress, and the Mongo database. The Mongo database is tested in other sections as they would fail if the database was not working.

9.4.1.1 Image Generation and Uploading
To test Podman and the image repository the buildimage.sh script in the repository was used **[119]**. This script creates an image and tags it with the version number, then uploads the image to the repository. The repository on skadi is known as repo.skadi and is exposed using the ingress, there is also a script on skadi called tags, which queries for the available tags of an image on repo.skadi. After running buildimage.sh and querying the repo we receive the output seen in **[Code 1]**. Since there is now a 0.1.0 tag within the repo, we can conclude that the image has been successfully built and uploaded.

9.4.1.2 Site Deployment and Network Configurations
To test the deployment of the site the PoolWatch helm chart was used with the configuration to use version 0.1.0, and the hostname test.skadi **[119]**. The resulting resources deployed in the skadi cluster are shown in **[Code 2]**. In there we see that the helm chart successfully deployed 2 pods in the poolwatch namespace, one for the PoolWatch app, and the other one for the Mongo database. The helm chart also configured two services one for each of the pods, and finally an ingress rule was configured, which points traffic from port 80 with the host test.skadi to the poolwatch-service.

9.4.1.3 Site Access
To test the access to the website through the ingress, a simple curl command requesting the head of the base route tells us if the traffic is making it through. **[Code 3]** shows this command and its result. From this header we first see that the request was successfully resolved as we received a 200 status code. We also see that the server type is Express and the content type is html, which is what we expect to receive from our MENN stack front page.

9.4.2 Backend Tests

To test the backend of the webserver, the device mimic was used to mimic a device with the serial number of 1. This device checks if it exists then registers itself if it does not, then the device sends two reports separated by 10 seconds to the webserver.

9.4.2.1 Existence Check
Every device has an unique serial number associated with it, and when the device starts up it attempts to register with the PoolWatch webserver. To verify if the device has been registered yet, an existence check is made, an example check is shown in **Figure 9.5**.

```
POST /device/exists HTTP/1.1
Host: test.skadi:80
Content-Type: application/json
Content-Length: 19
Connection: close

{"serialNumber": 1}

HTTP/1.1 200 OK
Date: Tue, 22 Jul 2025 15:37:05 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 16
Connection: close
X-Powered-By: Express
Access-Control-Allow-Origin: *
ETag: W/"10-/Xn4Kh95AH5gGcOOqqetzlOj8pQ"

{"answer":false}
False
POST /device/create HTTP/1.1
Host: test.skadi:80
Content-Type: application/json
Content-Length: 121
Connection: close
```

Figure 9.6: Existence Check

From the above message we request to know if there is a device registered with the serial number 1, since there is not, we received the answer false from the webserver. This is exactly what we expected to hear back.

9.4.2.2 Device Registration

If a device has not been registered with the webserver it must now register itself, to do that we can send a register request which is pictured in **[Code 4]**. We asked the webserver to create a device with certain values initialized in it. The response we received was 200 OK, with an update of false. This is expected as there is nothing expected of the device and the 200 indicates that the device was registered.

9.4.2.3 Report Sending

[Code 5] shows two reports being sent to the webserver. We can see that the responses to the report messages all have a 200 status code and are responding with a false to update as no further action from the device is required. This is the expected behavior from report uploads, so we can conclude that this route is working.

9.4.3 Frontend Tests

The tests run in 9.4.2 have already been run on the webserver, which means for the following tests a device with the serial number 1 and 2 reports associated with it already exists within the system. The tests within this are all functionality and the images shown do not represent the final frontend design of the website.

9.4.3.1 Login/Register Page

The expected behaviour of the login page is to respond with invalid credentials, when the user credentials do not match and to prevent two users with the same username from logging in. To test this an invalid user was used in **Figure 9.6**.

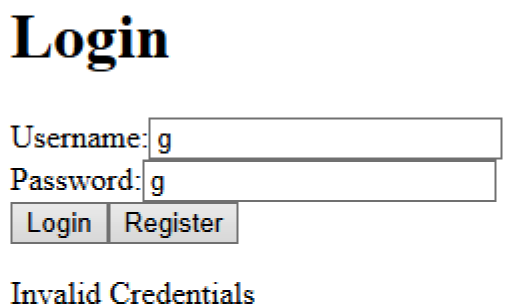


Figure 9.7: Invalid User

Since there is no user g, the error message Invalid Credentials was posted to the webpage, we then can create the user a with the password a, however for **Figure 9.7** we failed the password check which results in an error.

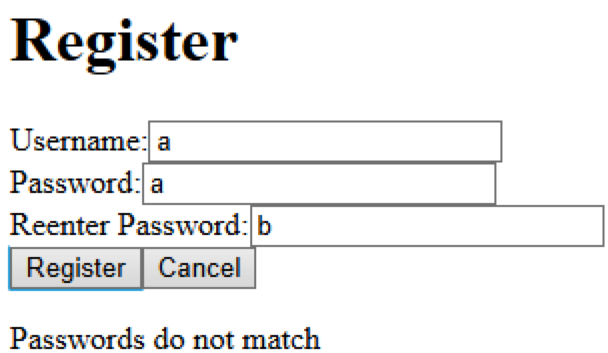


Figure 9.8: Mismatched Passwords

Once the user is created, we can now try to create a new user, which results in the error shown in **Figure 9.8**.

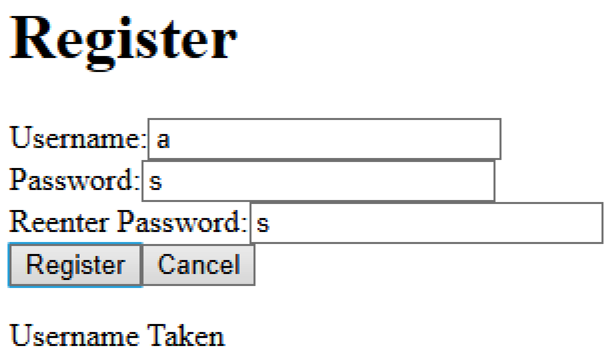


Figure 9.9: Existing User

From this we can conclude that the login page has all the expected functionality we require of it.

9.4.3.2 Main Page

For the main page we want to simply test adding device 1 to the page, which is done and shown in **Figure 9.9**.

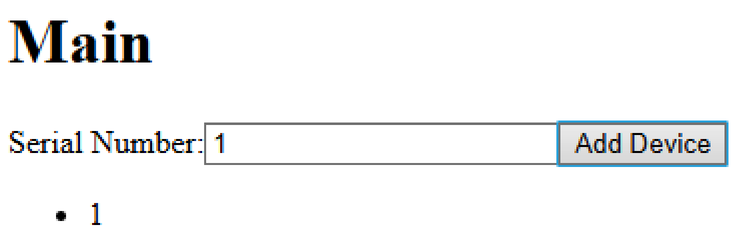


Figure 9.10: Main Page With Added Device

9.4.3.3 Device Page

From the device page we should see some statistics about the device and the reports housed in the device, which can be seen in **Figure 9.10**.

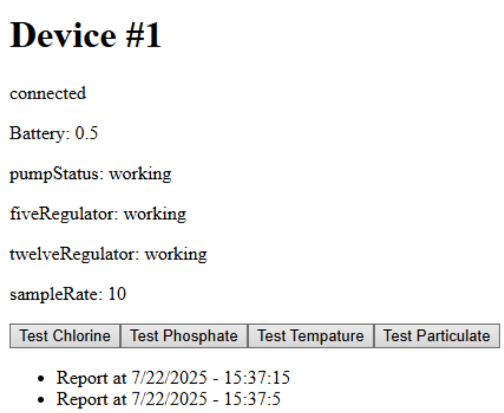


Figure 9.11: Device Page

We can also see in this figure that there is a 10 second difference between the reports which is what we expected from the tests in 9.2.2.3

9.4.3.4 Report Page

In the report page we expect to see the results of the report, which can be seen in **Figure 9.11**.

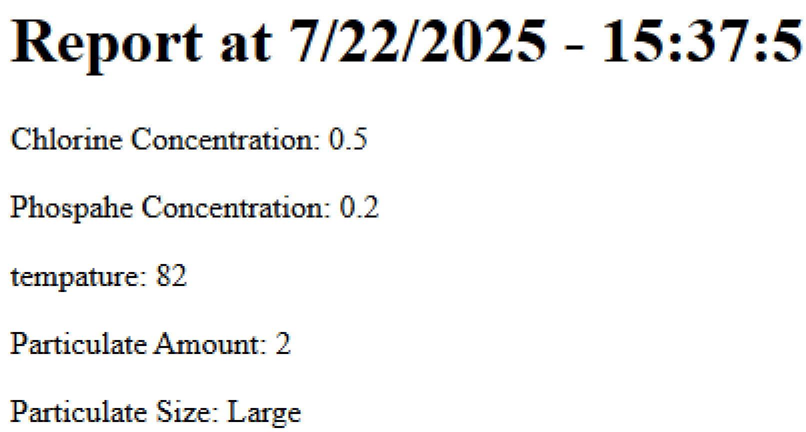


Figure 9.12: Report Page

9.4.4 Device to Webserver Tests

To test the device to server communication, I had an ESP32 programmed to register as a device and send a status and report to the webserver. I then requested that every single test was run on the device then restarted the device. The device checks for test requests whenever it uploads a status response. **[Code 6]** shows the response to the status update request from the device. From the response we can see that we are requesting that every test be run, and at the bottom of **[Code 6]** we see that the main program understands to run all four tests. We can also see that the device has been uploading reports and statuses to the webserver as **Figure 9.12** shows its current status and reports.

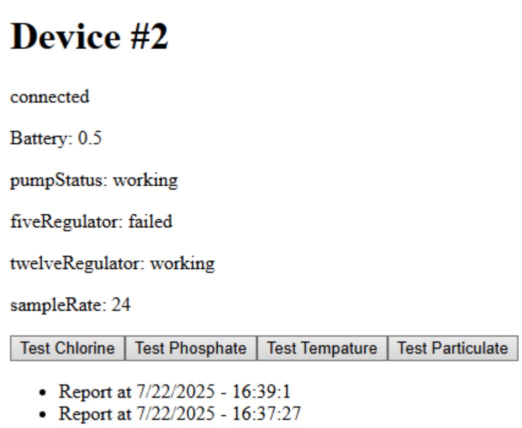


Figure 9.13: Device to Server Updates

9.4.5 Sensor to Webserver Tests

The purpose of this test was to evaluate the performance of a sensor and verify that our web application can successfully gather data from it. This included testing the connection between the sensor, the ESP32, and the website to ensure proper data transmission. This approach also allows future integration and testing of additional sensors, as well as the ability to send messages through the system. A temperature sensor that utilizes I2C was used to send temperature data to our functioning website. Currently, only the temperature sensor is configured to transmit data to the website. As a result, the reports contain updates only for temperature. Future integration enables the system to transmit additional data values, including chlorine levels, particulate size, pump status, and phosphate levels. Until those systems are online, their values are reported as -1 or null based on the driver configuration. The first figure below shows the temperature sensor connection to ESP32, with connections between the sensor and the SCL, SDA terminals of the ESP32 in order to enable I2C communication.The **second figure** below shows the reports received from the temperature sensor every second, with the time intervals showing that the report got sent every second. The second figure below shows an opened report for the device registered with the serial number 2, containing temperature data and demonstrating that the website successfully processed real-time sensor data.

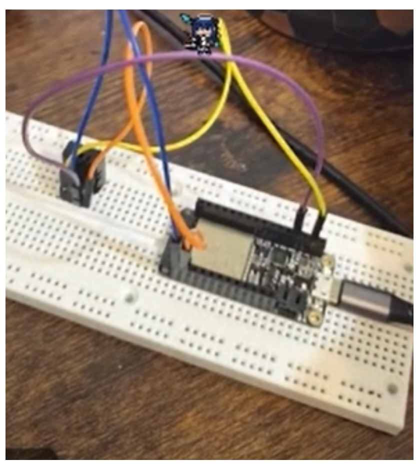


Fig 9.14:. Temperature Sensor Data Transmission Test

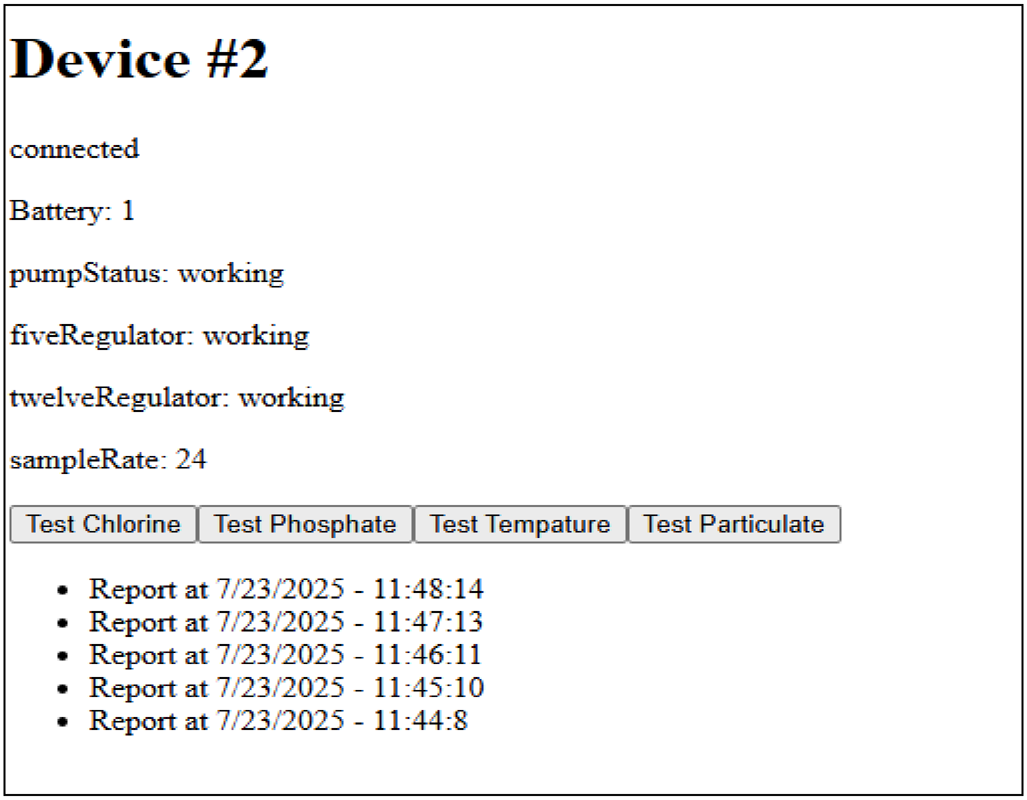


Fig 9.15: Temperature Data Reports Sent to Web Application

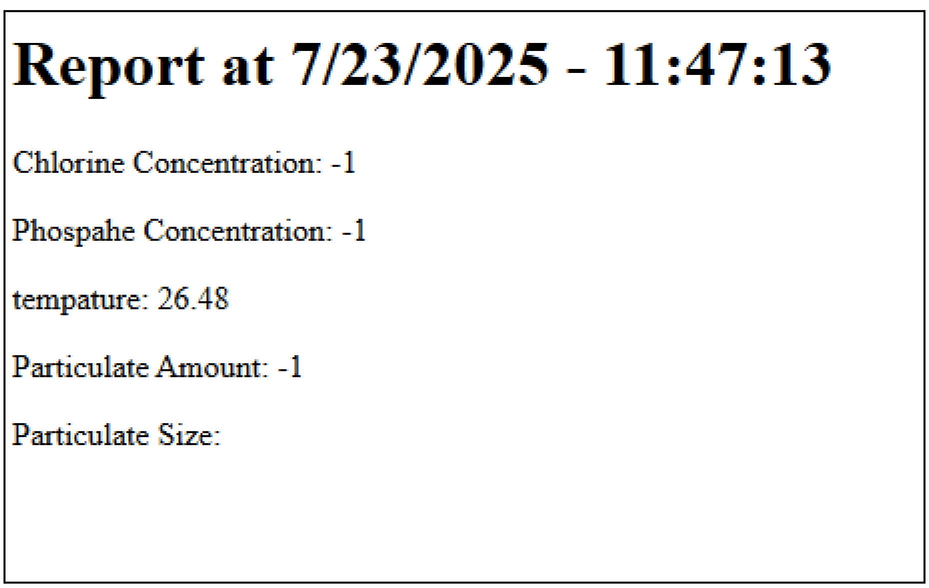


Fig 9.16: Temperature Data Report Opened

9.4.6 Calibration Curve Methodology

To successfully obtain accurate measurements from the chlorine and phosphate concentration analyzers a calibration curve had to be made. By controlling the concentration using the mass/volume formula, creating a stock solution, and pipetting a controlled amount into a 250 mL volumetric flasks would determine how much each sample would need to be diluted in parts per million for hypochlorous acid and parts per billion for orthophosphate. This had to be done one hour prior otherwise the electronic noise would output concentration values that didn't make any sense.

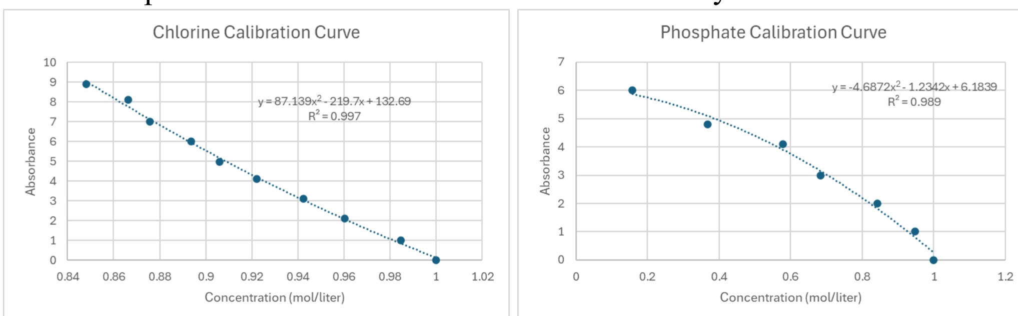


Fig 9.17: Calibration curves

9.4.7 Final Design Adjustments

In Senior Design I, automated reagent dispensing and mixing were included as stretch goals to implement if time and space allowed. During Senior Design II, we determined that adding these features would not be feasible within the constraints of the final enclosure and project timeline. Since they were not core requirements, they were left out of the final build so we could focus on completing the essential functionality.

Chapter 10 Administrative Content

10.1 Budget

All critical components needed for PoolWatch are paid for by each team member and are not funded by any external party or vendor. An account for every cost needed on each system for this project is given in **Table 10.1**.

Table 10.1: Budget for PoolWatch

Sub-Systems	Description	Cost

Optical System #1	This includes all lenses, mounts, and light sources.	$430.00
Optical System #2	This includes all lenses, mounts, and light sources.	$ 430.00
Electrical Hardware System	This includes PCB, Microcontrollers, etc.	$ 325.00
Mechanical System	Pump, valves, Servo motors, etc.	$90.00
Chemicals	DPD and Phosphomolybdic acid powders.	$35.00
Web Hosting	Hostname and web provider	$80.00
Total		$1,390.00

10.2 Project Setup

To stay on task and have a successful project, weekly group meetings were used for communication and progress checks. All team members are expected to contribute based on their respective schedules as shown in **Table 10.2**. This ensures accountability and helps the team stay on track. Prototypes are required to ensure that the final device is fully functional as specified.

Table 10.2: Project Setup

Start Date	Expected Date	Deadline	Task	Summary
5/12/2025	5/16/2025	5/20/2025	Create a team	Form a team with two other engineering disciplines: Austin Naugle (PSE), Jason

				Ser (PSE), Dylan Huges (CPE), and Ning Dim (CPE)
3/1/2025	5/22/2025	6/2/2025	Brainstorming	Coming up with a novel idea for a senior design project.
5/23/2025	5/29/2025	5/30/2025	Divide & Conquer Document and Committee Formation	Have 10 pages or more completed before the initial start of the meeting. Have at least 3 members on the committee.
5/23/2025	7/1/2025	7/7/2025	105-Page Document Landmark	The majority of the paper was completed at this time by assigning individual tasks.
5/23/2025	7/23/2025	7/29/2025	Completed Final Report	Satisfies the final report 120-page count with a fully functional demo of each system.
8/1/2025	8/18/2025	8/18/2025	Critical Design slides	Slides for the Critical design presentation.
8/18/2025	10/31/2025	11/24/2025	Final Project	Finalized prototype utilizing PCB boards.
8/18/2025	10/15/2025	10/15/2025	Final Web Server Design	Final design for the Webserver.
8/18/2025	9/10/2025	9/10/2025	Midterm Slides	Prepared slides for midterm.
8/18/2025	10/31/2025	11/24/2025	Final Demo Video	Final video for the demonstration of the product working.
8/18/2025	10/15/2025	10/15/2025	Final Website Design	Final project website with everything.
8/18/2025	10/15/2025	10/15/2025	Final Slides	Final slides for final presentation.

10.3 Work Distributions

All group members have a set of responsibilities to follow during the design and implementation process. In **Table 10.3**, the listed delegations were agreed upon for each team member to accomplish and provide enough work for all members to do within the scope of the project.

Table 10.3: Group Work Delegations

Deliverables	Primary	Secondary
Optical Chlorine and Phosphate Concentration Analyzer	Austin Naugle	NA
Particulate Imaging System	Jason Ser	NA
Packaging Design	Jason Ser	Austin Naugle
Temperature Sensor	Jason Ser	NA
Website Design and Administration	Dylan Hughes	NA
PCB Design and Integration	Dylan Hughes	Ning Dim
MCU Selection and Integration	Ning Dim	Dylan Hughes
Pump Design/Implementation	Ning Dim	Austin Naugle
Power Supply Selection and Integration	Ning Dim	Dylan Hughes

10.4 Bill of Materials

Table 10.4: Bill of Materials

Vendor	Item Name	Part No.	Description/Notes	Cost	Qty	Total Price
Amazon	Gikfun Water Pump	N/A	12V Peristaltic Pump	$13.69	1	$13.69

Amazon	Relay Module	N/A	12V 2-Channel Relay	$8.59	4	$17.81
Cole-Parmer	Koflo Disposable Mixer	SU-0.25-12H-PVDF	Inline Mixer	$4.20	1	$4.29
Liion Wholesale	Battery	4S1P	14.4V	$36.99	1	$36.99
Amazon	Beduan ¼” Inlet Solenoid Valve	1kl-04400-WIV	12V	$9.57	1	$9.57
DigiKey	Photodiode	S1223-01	center wavelength at 515nm wavelength	$28.70	1	$28.70
Mouser	Photodiode	BPW34FAS-Z	center wavelength at 880nm wavelength	$28.70	1	$11.28
Syronoptics	515 nm Narrow Bandpass Filter	SO3010225		$50.00	1	$50.00
Syronoptics	880 nm Narrow Bandpass Filter	SO301664		$50.00	1	$50.00
Thorlabs	0.5" Plano-Convex Lens	LA1074-B	B AR coating	$58.11	1	$58.11
Thorlabs	0.5" Plano-Convex Lens	LA1560-B	B AR coating	$56.07	1	$56.07
Thorlabs	0.5” Plano Convex Lens	LA1560-AB	AB AR coating	$56.07	1	$56.07

Lab4US	Fluorescence Quartz Cuvette (2-pack)	X002B3 D40X	Spectral range from 190-2500 nm	$94.99	1	$94.99
Taylor Technologies	Chorline Reagent	81331	Total Chlorine Residential Test Kit	$16.99	1	$16.99
Taylor Technologies	Phosphate Reagent	220793	Phosphate Residential Test Kit	$11.99	1	$11.99
Newport	38.1 mm EFL Achromat	PAC025	Front Lens in Particle Imager	$105.00	1	$105.00
Newport	50.8 mm EFL Achromat	PAC040	Back Lens in Particle Imager	$105.00	1	$105.00
Thorlabs	30 mm EFL Plano-Convex	LA1289-AB	Lens for 515nm LED Collimation	$41.11	2	$82.22
Raspberry Pi	Raspberry Pi 4 Model B	NA	4 GB Ram	$64.20	1	$64.20
BOJACK	Temperature Sensor	NA	Waterproof Version of DS18B20	$3.00	1	$3.00
innomaker	GS Camera Module	NA	Mono IMX296 Sensor	$43.00	1	$43.00
BacoFlo	Solenoid Valve	1/4"NPT	NA	$14.99	2	$29.98
DigiKey	56Ω Resistor	RC1206 FR-0756 RL	NA	$0.1	2	$0.1
DigiKey	70.6Ω Resistor	RN73H2 BTTD70 R6F100	NA	$0.23	1	$0.23
DigiKey	1kΩ Resistor	RC1206 FR-071 KL	NA	$0.1	4	$0.4
DigiKey	2.1kΩ Resistor	RC1206 FR-072 K1L	NA	$0.1	1	$0.1

DigiKey	4.7kΩ Resistor	RC1206FR-074K7L	NA	$0.1	1	$0.1
DigiKey	7.23kΩ Resistor	RN73R2BTTD7231B25	NA	$0.23	1	$0.23
DigiKey	10kΩ Resistor	RC1206FR-0710KL	NA	$0.1	29	$2.90
DigiKey	58.3kΩ Resistor	RT0805BRD0758K3L	NA	$0.1	1	$0.1
DigiKey	100kΩ Resistor	CRCW1206100KFKEAHP	NA	$0.23	9	$2.07
DigiKey	232kΩ Resistor	RC1206FR-07232KL	NA	$0.1	1	$0.1
DigiKey	500kΩ Resistor	CRMA1206AF500KFKEF	NA	$0.48	1	$0.48
DigiKey	768kΩ Resistor	RC1206FR-07768KL	NA	$0.1	1	$0.1
DigiKey	1MΩ Resistor	RNCS1206BKE1M00	NA	$0.17	2	$0.34
DigiKey	1.62MΩ Resistor	RMCF1206FT1M62	NA	$0.1	1	$0.1
DigiKey	.5pF Capacitor	CC1206CRNPO9BNR50	NA	$0.23	2	$0.46
DigiKey	100 pF Capacitor	1206GC101KAT1A	NA	$0.35	4	$1.40
DigiKey	2.2 nF Capacitor	C3216C0G2J222J115AA	NA	$0.33	1	$0.33
DigiKey	0.1uF Capacitor	CL31B104KBCNNNC	NA	$0.11	1	$0.11

DigiKey	1uF Capacitor	CL21B105KBFNNNE	NA	$0.08	4	$0.32
DigiKey	10uF Capacitor	CL31B106KAHNNNE	NA	$0.08	3	$0.24
DigiKey	22 uF Capacitor	CL21A226MQQNNNE	NA	$0.12	2	$0.24
DigiKey	47 uF Capacitor	CL31A476MQHNNNE	NA	$0.31	1	$0.31
DigiKey	1.5 uH Inductor	DFE252012P-1R5M=P2	NA	$0.22	1	$0.22
DigiKey	3.3 uH Inductor	TMS252012ALM-3R3MTAA	NA	$0.46	1	$0.46
DigiKey	op-amp	TLV3541IDBVR	For filters	$1.41	4	$5.64
DigiKey	NPN Transistor	BFU520WX	For Debug Logic	$0.36	16	$5.76
DigiKey	Debug LED	150080BS75000	For Debug indication	$0.19	8	$1.52
DigiKey	Switch	DS01-254-L-01BE	For turning on and off Power	$0.42	2	$0.84
DigiKey	USB to UART Conv.	CP2104-F03-GM	For programming MCU	$4.96	1	$4.96
DigiKey	MCU	ESP32-WROOM-32E-N4	Main Computing Unit	$4.84	1	$4.84
DigiKey	Micro USB	10103594-0001LF	For programing	$0.75	1	$0.75
DigiKey	Button	KMR211G LFS	For Reset	$0.59	1	$0.59
Mouser Elec.	Chlorine LED	WP7113ZGC	Test LED	$0.66	2	$1.32
DigiKey	Phosphate LED	XTHI30W	Test LED	$0.61	1	$0.61
DigiKey	Battery Connection	B2P-VH	To Connect Battery	$0.13	1	$0.13

DigiKey	12V Reg	TPS621361RGXR	For Power Management	$1.72	1	$1.72
DigiKey	5V Reg	TPS563200DDCR	For Power Management	$0.9	1	$0.9
DigiKey	3V3 Reg	TLV75901PDRVR	For Power Management	$0.46	1	$0.46
Total:						$954.64

Chapter 11 Conclusion

With pools being a common addition to houses across the United States many homeowners are faced with the task of keeping it maintained. This can be very difficult and time consuming as there are many things to keep in mind. For one, the chlorine and phosphate levels are important to regulate as imbalances can cause a lovely pool to become a mess. Sediment levels can also cause problems, making your pool feel dirty to be in. Sometimes you also want to be able to easily view the water temperature to see if your pool is at a comfortable temperature to swim in. PoolWatch hopes to help the average pool owner stay on top of all of these things.

PoolWatch utilizes a chlorine and a phosphate measurement system to track both of these different chemicals. The two optical systems rely on the varying absorbance of pool water when mixed with various chemicals to determine the concentrations. With narrow band pass filters as well as UV Quartz Cuvettes the systems can provide a high level of accuracy.

Also part of the PoolWatch system is its particulate imager. The system tracks the sediment that can be found in your pool. Providing the user with the quantity of different types of sediment this can also help homeowners see what's in their pool and how their filter is working. The system also corrects the chlorine measurement to provide the most accurate results possible. With two achromatic lenses leading to a global shutter camera, being illuminated with collimated light, the system minimizes aberrations to provide accurate counts of the particles found.

Since the two optical systems require water, PoolWatch uses a pump system. With a peristaltic pump the system is able to automatically provide water to the optical subsystems with a high precision. This prevents possible damage caused by overflowing or more work being given to the pool owner, forcing them to manually put water in for measurements. Part of this automation is the automatic dispensing of reagents required by the chlorine and phosphate system. The pump network utilizes a static inline mixer to properly combine the reagent and pool water before dispensing it into the cuvettes for measurements.

Knowing the temperature of your pool can also be a nice luxury for pool owners. With a high quality waterproof thermistor, PoolWatch is able to provide automatic and accurate measurements easily.

To make all of these different systems easily accessible to the users of PoolWatch is implemented with a website. Here the pool owners are able to see all of the data from their pool. They have control of starting different measurements and seeing how their pool is actively doing. Also available to them are the history of measurements. They are able to track the trend over time, how often they have needed to add chlorine, how often spikes in the phosphate levels happen, the amount of particulate that gets washed into their pool. With all of this information pool owners are able to make more informed decisions about how to maintain their pool and actively prevent potential problems.

Running all of these systems within PoolWatch is a centralized MCU. With the MCUs wifi connectivity it is able to provide easy communication between the website and the system. When needed it sends control signals to take tests as well as handles the automated periodic testing.

Providing the power to everything in the system is our central power unit. With its internal battery users do not need to worry about having a power outlet close enough to plug into. Its battery life of 30 days also ensures that the pool owner do not need to constantly worry about it running out of charge.

Overall PoolWatch aims to be a helpful, convenient, and easy to understand tool to help users maintain their pool. Helping them understand what needs to be done as well as show how their pool reacts over time.

Appendix A
References

[1] Null, N. (2024, August 6). *IntelliChem® Water Chemistry Controller. Pool Automation* | Pentair Pool. https://www.pentair.com/en-us/products/residential/pool-spa-equipment/pool-automation/intellichem_waterchemistrycontroller.html

[2] Swimming Pool and Spa Controllers with Analog Outputs. (n.d.). Hanna Instruments. Retrieved May 29, 2025, from https://hannainst.com/swimming-pool-and-spa-controllers-with-analog-outputs-bl121-bl123.html?gad_source=1&gad_campaignid=21273870174&gbraid=0AAAAADjk2PqplF-XVu5XgZBIMqaAsVcg0&gclid=EAIaIQobChMIx-7cueDIjQMVoq9aBR2iliX2EAQYAyABEgIbDfD_BwE

[3] WaterGuru SENSE - Smart Pool Monitoring System. (n.d.). Retrieved May 29, 2025, from https://waterguru.com/products/waterguru-sense?skip_ab=true&gad_source=1&gad_campaignid=19813456026&gbraid=0AAAACxYZ8CR2Zt4x6PXsjpQE5HBalCq5&gclid=EAIaIQobChMI7eTMut3IjQMVAZtaBR061jP3EAQYASABEgKPxfD_BwE

[4] Sutro, Inc. (n.d.). Sutro Smart Pool Water Monitor System. Retrieved May 29, 2025, from https://mysutro.com/?srsltid=AfmBOoqqg8HOq9knVx-9V1-lnuehPOfPFwB9B1RRQA168tteL68FZQPN

[5] Moberg, Ludvig, and Bo Karlberg. "An Improved N,N′-Diethyl-P-Phenylenediamine (Dpd) Method for the Determination of Free Chlorine Based on Multiple Wavelength Detection." Analytica Chimica Acta 407, no. 1 (2000/02/29/ 2000): 127-33. https://doi.org/https://doi.org/10.1016/S0003-2670(99)00780-1. https://www.sciencedirect.com/science/article/pii/S0003267099007801.

[6] "5.8: The Colourimetric Determination of Phosphate." 2024. Chemistry LibreTexts. September 2024. https://chem.libretexts.org/Ancillary_Materials/Laboratory_Experiments/Wet_Lab_Experiments/Organic_Laboratory_Experiments/05%3A_Analytical_Applied_and_Food_Chemistry_Laboratory_Experiments/5.08%3A_The_Colourimetric_determination_of_Phosphate.

[7] Docker. (n.d.). *What is a Container*? Docker. https://www.docker.com/resources/what-container/

[8] Aleksic, M. (2023, December 12). *Podman vs Docker: Everything You Need to Know | phoenixNAP KB*. Knowledge Base by PhoenixNAP. https://phoenixnap.com/kb/podman-vs-docker

[9] *Harvester*. (2025). Harvester. https://harvesterhci.io

[10] Kaplarevic, V. (2024, February 7). *LAMP Stack - What Is It, Advantages & Alternatives | phoenixNAP KB*. Knowledge Base by PhoenixNAP. https://phoenixnap.com/kb/what-is-a-lamp-stack

[11] Manoj, H. (2024, March 6). *Mern Stack Advantages And Disadvantages*. Skill Vertex. https://www.skillvertex.com/blog/mern-stack-advantages-and-disadvantages/

[12] Carroll, M., & Hanlon, R. (2025, February 14). *Sunsetting Create React App – React*. React.dev; React Blog. https://react.dev/blog/2025/02/14/sunsetting-create-react-app

[13] Charisol. (2023, October 9). *7 Advantages of Next.js Over the Create React App - Charisol | End to End Product Development Services*. Charisol. https://charisol.io/7-advantages-of-next-js-over-the-create-react-app/

[14] *Production-Grade Container Orchestration*. (n.d.). Kubernetes. https://kubernetes.io

[15] *Namespaces*. (2024, September 3). Kubernetes. https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/

[16] *Labels and Selectors*. (2024, July 30). Kubernetes. https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/

[17] *Pods*. (2025, April 7). Kubernetes. https://kubernetes.io/docs/concepts/workloads/pods/

[18] *Deployments*. (2025, June 11). Kubernetes. https://kubernetes.io/docs/concepts/workloads/controllers/deployment/

[19] *Service*. (2025, April 9). Kubernetes. https://kubernetes.io/docs/concepts/services-networking/service/

[20] *Ingress*. (2024, September 13). Kubernetes. https://kubernetes.io/docs/concepts/services-networking/ingress/

[21] *ConfigMaps*. (2024, September 11). Kubernetes. https://kubernetes.io/docs/concepts/configuration/configmap/

[22] *Secrets*. (2024, November 19). Kubernetes. https://kubernetes.io/docs/concepts/configuration/secret/

[23] *Persistent Volumes*. (2025, March 24). Kubernetes. https://kubernetes.io/docs/concepts/storage/persistent-volumes/

[24] *Introduction | RKE2*. (2025, June 16). Rke2.Io. https://docs.rke2.io

[25] *Helm*. (n.d.). Helm.sh. https://helm.sh

[26] *Modern Virtualization*. (2025, May 27). Portworx. https://portworx.com/modern-virtualization/?utm_medium=ppc&utm_source=google&utm_campaign=%7Bcampaign%7D&utm_term=kubevirt&gad_source=1&gad_campaignid=21831857714&gclid=CjwKCAjw3rnCBhBxEiwArN0QEwIR_pjfAN0FW3avSJL9RfkJUif5oUKVdilr7As159rYBjxp3_fJoRoCmEQQAvD_BwE

[27] Docker. (n.d.). *Overview of Docker Compose*. Docker Documentation. https://docs.docker.com/compose/

[28] Docker. (n.d.-b). *Specify a project name*. Docker Documentation. Retrieved June 16, 2025, from https://docs.docker.com/compose/how-tos/project-name/

[29] Docker. (n.d.-a). *Get started with Docker Compose*. Docker Documentation. https://docs.docker.com/compose/gettingstarted/

[30] Docker. (n.d.-b). *Networking*. Docker Documentation. https://docs.docker.com/compose/how-tos/networking/

[31] Docker. (n.d.-d). *Set environment variables*. Docker Documentation. https://docs.docker.com/compose/how-tos/environment-variables/set-environment-variables/

[32] Docker. (n.d.-d). *Secrets in Compose*. Docker Documentation. https://docs.docker.com/compose/how-tos/use-secrets/

[33] Rancher. (2025, June 16). *Requirements | RKE2*. Rke2.Io. https://docs.rke2.io/install/requirements

[34] DigitalOcean. (n.d.). *Droplet Pricing | DigitalOcean*. Digitalocean.com. Retrieved June 16, 2025, from https://www.digitalocean.com/pricing/droplets#basic-droplets

[35] Techtutorsti. (2023, November 29). *Google Cloud VM Pricing*. Medium; Medium. https://medium.com/@techtutorsti/google-cloud-vm-pricing-4c912400f73a

[36] Amazon. (n.d.). *AWS Pricing Calculator*. Calculator.aws. https://calculator.aws/#/createCalculator/ec2-enhancement?nc2=h_ql_pr_calc

[37] Microsoft. (n.d.). *Pricing Calculator | Microsoft Azure*. Azure.microsoft.com. https://azure.microsoft.com/en-us/pricing/calculator/?cdn=disable

[38] Nginx. (n.d.). *nginx*. Nginx.org. https://nginx.org/en/

[39] Istio. (n.d.). *The Istio service mesh*. Istio. https://istio.io/latest/about/service-mesh/

[40] *Chapter 3 Basic MOSFET logic gates 3.1 Inverter*. (n.d.). https://pages.jh.edu/aandreo1/216/Archives/2014/Handouts/POP_Ch3-4.pdf

[41] . *Product profile 1.1 General description 1.4 Quick reference data BFU520W NPN wideband silicon RF transistor*. (2014). https://www.nxp.com/docs/en/data-sheet/BFU520W.pdf

[42] Power, D. (2025, March 25). *How to calculate battery run time*. Lithium Ion Battery Manufacturer and Supplier in China-DNK Power. https://www.dnkpower.com/how-to-calculate-battery-run-time/

[43] Battery University. (2022c, September 20). *BU-802b: What does Elevated Self-discharge Do?* https://batteryuniversity.com/article/bu-802b-what-does-elevated-self-discharge-do

[44] Amazon. (n.d.). *KBT 12V 5200mah rechargeable li-ion battery, bare leads wire replacement battery pack with charger compatible with 12V devices RC car, boat, robot, DIY, LED light strip, CCTV camera*. Amazon. https://www.amazon.com/KBT-5200mAh-Rechargeable-Replacement-Compatible/dp/B0C242DYT1

[45] Amazon. (n.d.). *KBT 12V 5200mah rechargeable li-ion battery, bare leads wire replacement battery pack with charger compatible with 12V devices RC car, boat, robot, DIY, LED light strip, CCTV camera. Amazon.* https://www.amazon.com/KBT-5200mAh-Rechargeable-Replacement-Compatible/dp/B0C242DYT1

[46] *Buck Switching Regulators: Principles, performance, and Applications | RECOM*. (n.d.-b). https://recom-power.com/en/rec-n-buck-switching-regulators-principles,-performance,-and-applications-396.html?0

[47] ABLIC Inc. (2021, April 16). *What is a Switching Regulator?* https://www.ablic.com/en/semicon/products/power-management-ic/switching-regulator/intro-2/

[48] Toshiba Electronic Devices & Storage Corporation. (n.d.). *Basics of Low-Dropout (LDO) Regulator ICs*. https://toshiba.semicon-storage.com/info/application_note_en_20250213_AKX00141.pdf?did=13766

[49] Toshiba Electronic Devices & Storage Corporation. (n.d.). *Basics of Low-Dropout (LDO) Regulator ICs*. https://toshiba.semicon-storage.com/info/application_note_en_20250213_AKX00141.pdf?did=13766

[50] *Adafruit HUZZAH32 - ESP32 Feather*. (2017, May 10). Adafruit Learning System. https://learn.adafruit.com/adafruit-huzzah32-esp32-feather/power-management

[51] dfrobot.com. (n.d.). *20W Adjustable DC-DC Buck Converter with Digital Display*. https://www.dfrobot.com/product-1552.html?srsltid=AfmBOooqFTj7-dT3gjYtS2vj53NmQCcI-v7iwzWWimWAYf47ICiQm3K3

[52] dfrobot.com. (2025, May 10). *DC-DC Multi-Output Buck Converter (3.3V/5V/9V/12V)*. https://www.dfrobot.com/product-2599.html?srsltid=AfmBOoplvQDD93fLk4hFrNbFYQtUTleK_-B6U1rDs2Pq7TiZbPUHyZmy

[53] Texas Instruments Incorporated. (2021). *TPS6213X 3-V TO17-V, 3-A Step-Down Converter in 3-mm × 3-mm QFN Package*. https://www.ti.com/lit/ds/symlink/tps62132.pdf?ts=1751677686830

[54] Texas Instruments Incorporated. (2023). TPSM86325X 3V to 17V input, 3A, Synchronous Buck module in QFN package. In *TPSM86325x 3V to 17V Input, 3A, Synchronous Buck Module in QFN Package* [Technical report]. https://www.ti.com/lit/ds/symlink/tpsm863253.pdf?ts=1751697822608

[55] Amazon. (n.d.). *Gikfun peristaltic liquid pump with silicone tube and connector for Arduino aquarium lab analytic water (Pack of 1). Amazon.* *https://www.amazon.com/Gikfun-Peristaltic-Connector-Aquarium-Analytic/dp/B01IUVHB8E*

[56] Amazon. (n.d.). *Kamoer peristaltic dosing pump 12V for aquarium, hydroponics, lab nutrient and analytical water dosing. Amazon.* *https://www.amazon.com/Kamoer-Peristaltic-Hydroponics-Nutrient-Analytical/dp/B07GWJ78FN*

[57] Amazon. (n.d.). *Gikfun 12V mini aquarium air pump cooled diaphragm pump for Arduino (Pack of 1) EK1856. Amazon.* *https://www.amazon.com/Gikfun-Aquarium-Cooled-Diaphragm-EK1856/dp/B0744FWNFR*

[58] Amazon. (n.d.). *ALAMSCN submersible water pump 3W 220L/H, mini water pump for aquariums, fountains, hydroponics, 5.9 ft high lift, ultra quiet with 2 nozzles*. *Amazon*. https://www.amazon.com/ALAMSCN-Submersible-Aquariums-Fountain-Hydroponics/dp/B08PBQ1N1G

[59] Amazon. (n.d.). *HiLetgo DC 12V micro water pump brushless motor pump 3M 240L/H for pumping, circulation, aquarium*. *Amazon*. https://www.amazon.com/HiLetgo-Water-Brushless-Pumping-Aquarium/dp/B07X9L5RNS

[60] Industries, A. (n.d.). *Peristaltic Liquid Pump with Silicone Tubing - 12V DC Power*. https://www.adafruit.com/product/1150?srsltid=AfmBOopx8G87EP4FIMAkxSVYhCnIsfeFz-8mpMvmLLkS8EH4q7WCeU4B

[61] Amazon. (n.d.). *U.S. Solid 1/4" NPT electric solenoid valve 12V DC normally closed, brass body, for air water oil*. *Amazon*. https://www.amazon.com/Electric-Solenoid-Normally-Solid-U-S/dp/B00APDNPXG

[62] Cole-Parmer. (n.d.). *Koflo PPT-25-12-P polypropylene tube disposable static mixers, 1/4" OD, 1/8" ID, 12-element*. https://www.coleparmer.com/i/koflo-ppt-25-12-p-polypropylene-tube-disposable-static-mixers-1-4-od-1-8-id-12-element/1466903

[63] Amazon. (n.d.). *Koflo disposable static mixers – 1/2" pipe size, 6-element mixing tube for chemical and fluid blending*. *Amazon*. https://www.amazon.com/Koflo-Disposable-Static-Mixers/dp/B0F8R4TGJC

[64] Amazon. (n.d.). *Static mixer 1/2" tube, gas-liquid mixing pipe, plastic inline static mixer for laboratory and industrial applications*. *Amazon*. https://www.amazon.com/Static-Mixer-Gas-Liquid-Mixing-Pipe/dp/B0CZL2R9CT

[65] Arduino. (n.d.). Arduino UNO Rev3. https://store-usa.arduino.cc/products/arduino-uno-rev3

[66] Arduino. (n.d.). *UNO R4 WiFi*. https://store-usa.arduino.cc/products/uno-r4-wifi

[67] Industries, A. (n.d.-a). *Adafruit HUZZAH32 – ESP32 Feather Board*. https://www.adafruit.com/product/3405?srsltid=AfmBOoqr1F4YU9KTFmmP_pcRPHzhsbJa7o9UnVmbCINhYNVL7bKWyXpp

[68] Amazon. (n.d.). *HiLetgo 2-channel 12V relay module with optocoupler, low level trigger expansion board for Arduino*. *Amazon*.

https://www.amazon.com/HiLetgo-Channel-Isolated-Optocoupler-Trigger/dp/B00LW15F42

[69] Hertz, J. (n.d.). *Understanding the flyback diode: your essential guide to functionality and why you need one*. Flux.ai. https://www.flux.ai/p/blog/understanding-the-flyback-diode-your-essential-guide-to-functionality-and-why-you-need-one

[70] Fielding, R., Nottingham, M., & Reschke, J. (2022). *RFC 9110 HTTP Schematics* (No. 2070-1721). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc9110.html

[71] "Chlorine." *Encyclopædia Britannica*, Encyclopædia Britannica, inc., 15 May 2025, www.britannica.com/science/chlorine.

[72] NOAA Office of Response and Restoration, US GOV. "Search Chemicals." NOAA. Accessed June 9, 2025. https://cameochemicals.noaa.gov/chemical/2862.

[73] "N,N-Diethyl-P-Phenylenediamine." National Center for Biotechnology Information. PubChem Compound Database. Accessed June 10, 2025. https://pubchem.ncbi.nlm.nih.gov/compound/n_n-diethyl-p-phenylenediamine.

[74] NOAA Office of Response and Restoration, US GOV. "Search Chemicals." NOAA. Accessed June 10, 2025. https://cameochemicals.noaa.gov/chemical/20194#section1.

[75] Astuti, Maryani, Rongjing Xie, and Nicky Aziz. "Laboratory and Pilot Plant Scale Study on Water Dechlorination by Medium Pressure Ultraviolet (Uv) Radiation." MATEC Web of Conferences 101 (01/01 2017): 02003. https://doi.org/10.1051/matecconf/201710102003.

[76] "Marvin JS." Marvin JS | Chemaxon Docs. Accessed June 11, 2025. https://docs.chemaxon.com/display/docs/marvin-js_index.md.

[77] Modanese, C., Laine, H. S., Pasanen, T. P., Savin, H., & Pearce, J. M. (2018). Economic Advantages of Dry-Etched Black Silicon in Passivated Emitter Rear Cell (PERC) Photovoltaic Manufacturing. *Energies*, *11*(9), 2337. https://doi.org/10.3390/en11092337

[78] Liu, Xin, Huahua Cai, Jing Zou, Zijun Pang, Baoling Yuan, Zhenming Zhou, and Qingfeng Cheng. "Spectrophotometric Determination of Trace Permanganate in Water with N,N-Diethyl-P-Phenylenediamine (Dpd)." Chemosphere 212 (2018/12/01/ 2018): 604-10. https://doi.org/https://doi.org/10.1016/j.chemosphere.2018.08.087. https://www.sciencedirect.com/science/article/pii/S0045653518315637.

[79] "Phosphorus." Encyclopædia Britannica, May 17, 2025. https://www.britannica.com/science/phosphorus-chemical-element.

[80] NOAA Office of Response and Restoration, US GOV. "Search Chemicals." NOAA. Accessed June 10, 2025. https://cameochemicals.noaa.gov/chemical/1337.

[81] National Center for Biotechnology Information (2025). PubChem Compound Summary for CID 61578, Ammonium Molybdate. Retrieved July 3, 2025 from https://pubchem.ncbi.nlm.nih.gov/compound/Ammonium-Molybdate.

[82] O'Dell, James W. "Method 365.1, Revision 2.0: Determination of Phosphorus by Semi-Automated Colorimetry." epa. Accessed July 3, 2025. https://www.epa.gov/sites/default/files/2015-08/documents/method_365-1_1993.pdf.

[83] Yaakob, M. A., Mohamed, R. M. S. R., Al-Gheethi, A., Aswathnarayana Gokare, R., & Ambati, R. R. (2021). Influence of Nitrogen and Phosphorus on Microalgal Growth, Biomass, Lipid, and Fatty Acid Production: An Overview. *Cells*, *10*(2), 393. https://doi.org/10.3390/cells10020393

[84] Drummond, L., and W. Maher. "Determination of Phosphorus in Aqueous Solution Via Formation of the Phosphoantimonylmolybdenum Blue Complex. Re-Examination of Optimum Conditions for the Analysis of Phosphate." Analytica Chimica Acta 302, no. 1 (1995/02/10/ 1995): 69-74. https://doi.org/https://doi.org/10.1016/0003-2670(94)00429-P.

[85] Newport Corporation. (n.d.). PAC025 achromatic doublet lens (12.7 mm, MgF_2-coated, 400–700 nm). Newport. https://www.newport.com/p/PAC025

[86] Thorlabs, Inc. (n.d.). AC127-050-A – f = 50 mm, Ø½″ achromatic doublet, AR-coated 400–700 nm. Thorlabs. https://www.thorlabs.com/thorproduct.cfm?partnumber=AC127-050-A

[87] Newport Corporation. (n.d.). PAC024AR.14 visible achromatic doublet lens (12.7 mm, MgF_2-coated, 400–700 nm). Newport. https://www.newport.com/p/PAC024AR.14

[88] Newport Corporation. (n.d.). PAC040 achromatic doublet lens (25.4 mm, MgF_2-coated, 400–700 nm). Newport. https://www.newport.com/p/PAC040

[89] Thorlabs, Inc. (n.d.-a). AC254-050-A — f = 50 mm, Ø1″ achromatic doublet, AR-coated 400–700 nm. Thorlabs. https://www.thorlabs.com/thorproduct.cfm?partnumber=AC254-050-A

[90] Thorlabs, Inc. (n.d.-b). AC254-050-AB — f = 50 mm, Ø1″ achromatic doublet, AR-coated 400–1100 nm. Thorlabs. https://www.thorlabs.com/thorproduct.cfm?partnumber=AC254-050-AB

[91] Thorlabs, Inc. (n.d.). *LB1258 N‑BK7 bi‑convex lens, Ø ½" (12.7 mm), f = 30.0 mm, uncoated*. Thorlabs. https://www.thorlabs.com/thorproduct.cfm?partnumber=LB1258 thorlabs.co.jp+14GoPhotonics+14thorlabs.co.jp+14

[92] Thorlabs, Inc. (n.d.). *LA1289‑A N‑BK7 plano‑convex lens, Ø ½", f = 30 mm, AR‑coated 350–700 nm*. Thorlabs. https://www.thorlabs.com/thorproduct.cfm?partnumber=LA1289-A Thorlabs+7GoPhotonics+7thorlabs.co.jp+7

[93] Thorlabs, Inc. (n.d.). *LA1289‑AB N‑BK7 plano‑convex lens, Ø ½", f = 30 mm, AR‑coated 400–1100 nm*. Thorlabs. https://www.thorlabs.com/thorproduct.cfm?partnumber=LA1289-AB

[94] INNO-MAKER. (n.d.). CAM-MIPI296RAW-Trigger global shutter camera module (Sony IMX296). INNO-MAKER. https://www.inno-maker.com/product/cam-mipi296raw-trigger/

[95] Arducam. (n.d.-a). OV9281 MIPI 1 MP monochrome global shutter camera module (M12 mount). Arducam. https://www.arducam.com/ov9281-mipi-1mp-monochrome-global-shutter-camera-module-raspberry-pi.html

[96] Arducam. (n.d.-b). 5 MP CS-mount camera module for Raspberry Pi. Arducam. https://www.arducam.com/arducam-5mp-cs-mount-camera-module-raspbery-pi.html

[97] Raspberry Pi Foundation. (n.d.-a). Raspberry Pi 5. Raspberry Pi. https://www.raspberrypi.com/products/raspberry-pi-5/

[98] Raspberry Pi Foundation. (n.d.-b). Raspberry Pi 4 Model B. Raspberry Pi. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[99] Raspberry Pi Foundation. (n.d.). Raspberry Pi Zero [Product page]. Raspberry Pi. https://www.raspberrypi.com/products/raspberry‑pi‑zero/

[100] Marquet, C. (2023, February 7). *Influence of the tubing size on the flow of a peristaltic pump*. Darwin Microfluidics. https://blog.darwin‑microfluidics.com/influence-of-the-tubing-size-on-the-flow-of-a-peristaltic-pump/ (blog.darwin-microfluidics.com)

[101] "Chlorine Testing DPD Method." CHLORINE RESIDUAL TESTING FOR DRINKING WATER SYSTEMS USING DPD COLORIMETRIC TEST KITS, 2017. https://health.hawaii.gov/sdwb/files/2017/07/2017ChlorResTestDPDColorimetric.pdf.

[102] "Colorimetric Analysis of P Using the Molybdenum Blue Method." DGT Research, 2025. https://www.dgtresearch.com/detailed-user-guides/Colorimetric-analysis-of-P-A2-20.pdf.

[103] "Polyethylene Chemical Resistance Chart." CDF Technical Memo, 2004. https://www.cdf1.com/technical%20bulletins/Polyethylene_Chemical_Resistance_Chart.pdf.

[104] National Electrical Manufacturers Association. (n.d.). *American national standard for portable lithium primary cells and batteries: Safety standard (ANSI C18.3M Part 2)*. NEMA. https://www.nema.org/standards/view/american-national-standard-for-portable-lithium-primary-cells-and-batteries-safety-standard

[105] IEEE. (2023). *IEEE guide for the characterization and evaluation of lithium-based batteries in stationary applications* (IEEE Standard 1679.1-2023). https://ieeexplore.ieee.org/document/11006491

[106] Kelechava, B. (2025, May 30). *ANSI C18.2M: Portable Rechargeable Batteries Specifications*. The ANSI Blog. https://blog.ansi.org/ansi/ansi-c182m-portable-rechargeable-cell-batteries/

[107] IEEE. (2023). *IEEE standard for terminology and test methods for analog to digital converters* (IEEE Standard 1241-2023). Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/IEEESTD.2023.10001695

[108] Texas Instruments. *TLV759P: 500-mA, Low-Dropout (LDO) Voltage Regulator*. Dallas, TX: Texas Instruments; 2023. Available from: https://www.ti.com/lit/ds/symlink/tlv759p.pdf

[109] Texas Instruments. *TLV761: 500-mA, Low Dropout Regulator With Foldback Current Limit*. Dallas, TX: Texas Instruments; 2023. Available from: https://www.ti.com/lit/ds/symlink/tlv761.pdf

[110] U.S. Environmental Protection Agency. (n.d.). Sediments. CADDIS. https://www.epa.gov/caddis/sediments

[111] Duracell US Operations, Inc. *Article Information Sheet (AIS): Lithium Coin Batteries*. Duracell; March 2023. Accessed July 6, 2025. https://www.duracell.com/wp-content/uploads/2023/03/AIS-Lithium-Coin-2023.pdf

[112] Procell. *Article Information Sheet (AIS): Lithium Coin Batteries*. Procell; January 2023. Accessed July 6, 2025. https://www.procell.com/wp-content/uploads/2023/01/AIS-Procell-LiCoin-2023-.pdf

[113] "Comparative Analysis of Quartz Cuvette Models for Scientific & OEM Use." Machined Quartz, May 29, 2025. https://machinedquartz.com/comparative-analysis-of-quartz-cuvette-models-for-scientific-oem-use/.

[114] *IEEE Standard for Information Technology — Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks — Specific Requirements* (No. 802.11). (2025). IEEE Computer Society. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10979691

[115] Dell. (2024, December 9). *Wi-Fi Standards: IEEE 802.11ac, 802.11ax, and Wireless Internet Standards*. Dell.com; Dell. https://www.dell.com/support/contents/en-ke/article/product-support/self-support-knowledgebase/networking-wifi-and-bluetooth/wi-fi-network-standards-overview

[116] ESP. (2025). *ESP Product Selector*. Espressif.com. https://products.espressif.com/#/product-selector?names=ESP32-S3

[117] *What are the common wireless standards?* (2025). Konicaminolta.eu. https://manuals.konicaminolta.eu/ineo-4000P-4700P/EN/ntwk_guide/what-are-wireless-standards-topic.html

[118] Docker. (2024). *Build and push your first image*. Docker Documentation. https://docs.docker.com/get-started/introduction/build-and-push-first-image/

[119] Senior Design Group 4. (2025). *GitHub - Mudrock1776/PoolWatch: Group 4 Senior Design Project Repository*. GitHub. https://github.com/Mudrock1776/PoolWatch

[120] Adafruit. (n.d.). *Waterproof DS18B20 digital temperature sensor + resistor*. Pi Shop US. https://www.pishop.us/product/waterproof-ds18b20-digital-temperature-sensor-resistor/DigiKey+15pishop.us+15pishop.ca+15

[121] Texas Instruments. (n.d.). *LM35DZ-NOPB analog temperature sensor (TO-92-3, 0 °C to 100 °C, 10 mV/°C)*. Digi-Key Electronics. https://www.digikey.com/en/products/detail/texas-instruments/LM35DZ-NOPB/32489

[122] Jaybva. (n.d.). *RTD Pt100 temperature sensor probe, waterproof stainless steel thermistor, three-wire, –50 °C to 200 °C* [Product page]. Amazon.com. https://www.amazon.com/Temperature-Jaybva-Waterproof-Stainless-Thermistor/dp/B07YVGVSDM

[123] International Organization for Standardization. (2014). *ISO 13322-1:2014 – Particle size analysis: Image analysis methods – Part 1: Static image analysis methods*. ISO. https://www.iso.org/standard/51257.html

Appendix B
Software Code and Output

[Code 1]

```
skadi@skadi:~/programs/PoolWatch$ tags poolwatch
{"name":"poolwatch","tags":["0.0.1","0.1.0"]}
```

[Code 2]

```
^Cskadi@skadi:~/programs/PoolWatch$ kgpo -n poolwatch
NAME                               READY   STATUS    RESTARTS   AGE
mongo-production-664799d799-hj5dg  1/1     Running   0          60s
poolwatch-9644cb954-8nsmh          1/1     Running   0          60s
skadi@skadi:~/programs/PoolWatch$ k get services -n poolwatch
NAME                      TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)     AGE
mongo-production-service  ClusterIP   10.43.239.95   <none>        27017/TCP   78s
poolwatch-service         ClusterIP   10.43.198.42   <none>        8080/TCP    78s
skadi@skadi:~/programs/PoolWatch$ k get ingress -n poolwatch
NAME                CLASS    HOSTS        ADDRESS        PORTS   AGE
poolwatch-ingress   <none>   test.skadi   192.168.1.69   80      2m3s
```

[Code 3]

```
skadi@skadi:~/programs/PoolWatch$ curl --head "http://test.skadi/"
HTTP/1.1 200 OK
Date: Tue, 22 Jul 2025 15:26:09 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 5083
Connection: keep-alive
X-Powered-By: Express
Access-Control-Allow-Origin: *
Accept-Ranges: bytes
Cache-Control: public, max-age=0
Last-Modified: Tue, 22 Jul 2025 14:39:50 GMT
ETag: W/"13db-198329396f0"
```

[Code 4]

```
POST /device/create HTTP/1.1
Host: test.skadi:80
Content-Type: application/json
Content-Length: 121
Connection: close

{"serialNumber": 1, "battery": 0.5, "pumpStatus": true, "fiveRegulator": true, "twelveRegulator": true, "sampleRate": 10}

HTTP/1.1 200 OK
Date: Tue, 22 Jul 2025 15:37:05 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 16
Connection: close
X-Powered-By: Express
Access-Control-Allow-Origin: *
ETag: W/"10-do6uB5t9n+N7lyRaEx6jXcf7Kd4"

{"update":false}
```

[Code 5]

```
POST /report/add HTTP/1.1
Host: test.skadi:80
Content-Type: application/json
Content-Length: 127
Connection: close

{"serialNumber": 1, "report": {"ClCon": 0.5, "PCon": 0.2, "tempature": 82, "particulateAmount": 2, "particulateSize": "Large"}}

HTTP/1.1 200 OK
Date: Tue, 22 Jul 2025 15:37:05 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 16
Connection: close
X-Powered-By: Express
Access-Control-Allow-Origin: *
ETag: W/"10-do6uB5t9n+N7lyRaEx6jXcf7Kd4"

{"update":false}
POST /report/add HTTP/1.1
Host: test.skadi:80
Content-Type: application/json
Content-Length: 128
Connection: close

{"serialNumber": 1, "report": {"ClCon": 0.2, "PCon": 0.5, "tempature": 420, "particulateAmount": 3, "particulateSize": "Small"}}

HTTP/1.1 200 OK
Date: Tue, 22 Jul 2025 15:37:15 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 16
Connection: close
X-Powered-By: Express
Access-Control-Allow-Origin: *
ETag: W/"10-do6uB5t9n+N7lyRaEx6jXcf7Kd4"

{"update":false}
```

[Code 6]

```
HTTP/1.1 200 OK
Date: Tue, 22 Jul 2025 16:39:00 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 120
Connection: close
X-Powered-By: Express
Access-Control-Allow-Origin: *
ETag: W/"78-/G0dYwyuXgnow4dZ1EaD6rjNSkI"

{"needUpdate":true,"sampleRate":24,"testChlorine":true,"testPhosphate":true,"testTempature":true,"testParticulate":true}

true
true
true
true
true
Update Sample Rate
24.00
Run Chlorine Test
Run Phosphate Test
Run Tempature Test
Run Particulate Test
```

Appendix C
ChatGPT Prompts and Outcomes

[Chat 1] https://chatgpt.com/share/6862bcf6-457c-8005-89a4-8be25db701cd

[Chat 2] https://chatgpt.com/share/6862c743-da40-8005-9ebf-34b49d356b4c

[Chat 3] https://chatgpt.com/share/6862ca05-6498-8005-b1ad-d76a04636aeb

[Chat 4] https://chatgpt.com/share/6867392b-72e4-8000-89dd-a01425adedec

[Chat 5] https://chatgpt.com/share/6869b3b9-8d20-8006-a5d9-3ae100f3aa31