AUTOMATIC POOL MONITOR AND REGULATOR

By

Arnold Ancheril

Raymond Chen

Swarna Jammalamadaka

Final Report for ECE 445, Senior Design, Spring 2024

TA: Selva Subramaniam

1 May 2024

Project No. 53

Abstract

This report serves as a comprehensive overview of the design and implementation of an automated water treatment solution for swimming pools. The system aims to maintain optimal water conditions by accurately measuring and adjusting pH and chlorine levels. By utilizing microcontrollers, sensors, and a dispenser, the system automates the process, eliminating the need for manual intervention. It consists of several components, including sensors for measuring chemical levels and motors for controlling chemical dispensers. The design considerations include power requirements, system overview, and cost breakdown. The report presents a detailed cost analysis and a breakdown of the project, outlining the labor and successes and failures. The final section explores potential improvements, highlighting areas for further research and development to enhance the system's functionality and efficiency.

Contents

1. Introduction 1
    1.1 Problem 1
    1.2 Solution 1
    1.3 High-Level Design 2
    1.4 High-Level Requirements List 3
2. Design 4
    2.1 Design Procedure 4
        2.1.1 Sensor Unit 4
        2.1.2 Dispenser Unit 4
        2.1.3 Power 4
    2.2 Design Details 5
        2.2.1 Sensor Subsystem Design 5
        2.2.2 Dispenser Subsystem Design 7
        2.2.3 Breadboard Design 9
        2.2.4 Future Modifications 9
3. Design Verification 11
    3.1 Physical Testing 11
    3.2 Requirements and Verification 11
4. Costs and Schedule 12
    4.1 Parts Cost 12
    4.2 Labor Cost 12
    4.3 Schedule 13
5. Conclusions 14
    5.1 Accomplishments 14
    5.2 Uncertainties 14
    5.3 Ethical considerations 14

5.4 Future work 15
References 16
Appendix A Requirement and Verification Table 18

1. Introduction

Our project aims to create a design which helps make a task more efficient and accurate by automating the process. Our project is geared towards maintaining a healthy quality of water for pools.

1.1 Problem:

Pools must be properly maintained for the safety of swimmers and people around the area. Maintaining safe and hygienic swimming pool water conditions is an essential aspect of pool management. However, the need to regulate chemicals in pools through physical tests demands considerable lifeguard time and effort. Ensuring that the chemical composition of pool water meets safety and industry standards is important for safeguarding the health and well-being of swimmers. According to the CDC, maintaining the water quality of a pool by using chlorine concentration and pH "will help prevent the spread of germs that cause swimming-related illnesses" [1]. Yet, the conventional methods rely on manual testing such as test kits/strips which requires lifeguard efforts and time. This not only diverts lifeguards' attention but also can introduce human error in these tests. If chemical levels are too out of range, lifeguards also need to manually rebalance them or in extreme cases, close the pool down.

1.2 Solution

Currently, there are products in the market that can automatically monitor pool health for user access. However, these products are not only expensive, but they also do not automatically regulate and rebalance the chemicals in the pool. Automating both processes can reduce the responsibilities of managers and lifeguards and make sure that water quality is always at appropriate levels. Our project aims to address this problem by sensing three main components of pool water — pH, chlorine, and temperature — and dispensing appropriate chemicals to rebalance pH and chlorine to health ranges. The sensors will be separate from the dispenser unit to make sure the chemicals are adequately dispersed throughout the pool, and will wirelessly communicate with the dispenser.

1.3 High-Level Design

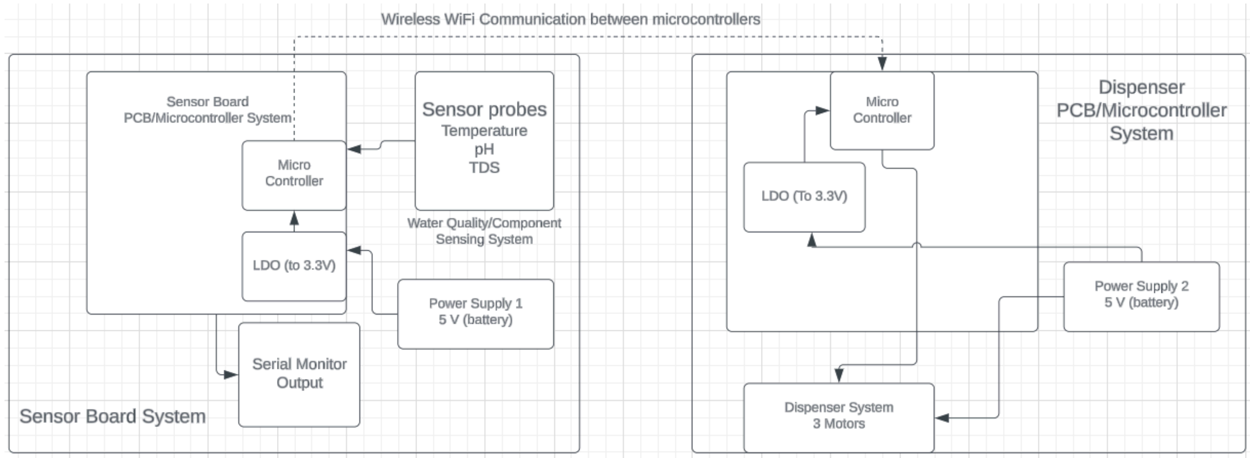

Figure 1: High-Level Block Diagram

As shown in Figure 1, our project has two main subsystems: the sensor board system and dispenser board system. The sensor board system contains the sensors and its circuitry, and the dispenser board system contains the dispenser and its circuitry.

The sensor system will contain the sensor probes to extract data from the environment and a microcontroller that evaluates the data received from the sensors. The sensors we are using are a pH sensor, a Total Dissolved Solids (TDS) sensor, and a liquid temperature sensor. We are using a TDS sensor as a way to track chlorine levels. The sensors must be able to collect correct readings, which can be validated by seeing appropriate values from changing environments. Based on these readings, the microcontroller must calculate if these sensor readings are in the correct standard ranges for a pool. Temperature readings should be between 78-82 degrees Fahrenheit, TDS readings should be under 1000 ppm, and pH readings should be between 7.2-7.8. The microcontroller will output the readings and alerts to the serial monitor, or send a wireless WiFi signal to the dispenser system if the readings are not within the correct ranges as listed above.

The dispenser microcontroller must receive the signal sent by the sensor microcontroller and control the motors to dispense the appropriate chemicals into the pool. Each motor is connected to the microcontroller and will complete rotations to dispense chemicals into the pool.

Both systems also have a power subsystem that delivers power to microcontrollers, sensors, and motors. The primary power supply, a 4.5V battery, supplies power to the sensors and motors. The primary supply must be stepped down using a low dropout voltage regulator to 3.3V to power the microcontroller.

1.4 High-Level Requirements List:

For successful operations, our project must accomplish the following:

1. The pool sensors must accurately measure and output analog readings for the water quality, with temperature between 78-82 degrees, pH between 7.2 to 7.8, and total dissolved solids level (TDS) levels between 0-1000 ppm. The standard deviation for temperature should be within 1 degree, pH should be within 0.1, and TDS levels should be within 100 ppm.
2. The microcontroller must be able to calculate if readings are in an acceptable range or send a signal to the dispenser unit to release a quantifiable amount of necessary chemicals into the pool otherwise. If the TDS levels do not change from prior measurements, an external alert should be sent to the user. The controller should also display pool temperature, pH, and TDS levels on the LED displays.
3. The dispenser unit will release 0.018 cubic inches of chemicals with a 5% standard deviation ($0.0171in^3$ to $0.0189in^3$) using stepper motors according to microcontroller instructions.

2. Design

For our project, we had two main components which communicate with each other to work correctly: the sensor unit and dispenser unit.

2.1 Design Procedure

2.1.1 Sensor Unit

Our sensor unit consists of three sensors for measuring temperature, pH, and TDS. Our main design consideration for this unit was to protect the circuitry from having contact with the water, while making sure the sensors have complete contact with the water for accurate readings. To accomplish this, the sensor unit design is a box with three holes at the bottom of which the sensors hang out, and all the circuitry is in the protected and waterproof box. The box could be placed beside the pool, or in a hole on the side of the pool for better protection.

2.1.2 Dispenser Unit

The dispenser unit consists of three motors and our circuitry. The motors control the amount of chemicals being dispensed into the water based on the number of rotations of the motor. There is a specific amount of the chemical being dispensed into the pool with each rotation, so the number of rotations determines the amount of total chemical being added. Based on how much the environment changes for one rotation of chemicals, the number of rotations are decided. For a standard pool of 10,000 gallons, 1.5lbs of baking soda will increase the pH by one decimal point and 10oz of pH pool reducer powder will decrease the pH by one decimal point. For the pool we used in our project, two rotations of baking soda increased the pH and one rotation of pH reducer powder decreased the pH by one decimal point. Therefore, these were the calculations we used in our code. The circuitry for this unit includes the electrical components of the motor and wires from the breadboard/PCB. These components are fully protected by a box behind the dispenser where they will not come into contact with the water.

2.1.3 Power

In order to supply power to both units in our design, we used a 4.5V battery pack which would be included in the circuitry part of each unit and protected from the water accordingly. In order to make sure our battery was sufficient enough for the microcontroller and motors for receiving the rated power and voltage, we considered a few options. We could have potentially separated the power supply for both aspects, but doing so would increase the external power system complexity. Instead, we opted to step down the 5V voltage for motors to 3.3V for the microcontroller. However, this also poses challenges, since if we use an unregulated or non-isolated power supply, changing the load from the motor might fluctuate power and voltage to the microcontroller which can shut off or overpower it. To combat this, we should use a voltage step-down regulator instead of just a buck converter so that the voltage to the microcontroller is steady. If this does not work, then adding a capacitor between voltage and ground by the microcontroller should be enough to balance voltage ripples.

For the dispenser unit, we have to make sure that the power supply is sufficient for the rest of the circuit. In our design, the microcontroller and 3 motors consume the most power.

However, since the pool will only require a change in pH and chlorine, there will only ever be two motors running at once (sodium bisulfate or sodium bicarbonate and chlorine). Using 3W for our microcontroller as was stated on the datasheet for this unit and 5V and 1.5A for the each motor, we can calculate the power our battery must have to

Power of each motor = 5V * 1.5A = 7.5
3W + 2(7.5) = 18W

Our battery pack is strong enough to handle this sort of power.

2.2 Design Details

2.2.1 Sensor Subsystem Design

The first task for the sensor system was to find the sensors. The lab had a liquid temperature sensor, so we decided on the Honeywell PK 87786 liquid temperature sensor. This sensor acts as a potentiometer based on the temperature of the liquid. Using a resistor divider with this sensor, we use the divided voltage as an analog measure of temperature. For the pH sensor, we opted for the Gravity Analog pH Sensor. Finally, to measure chlorine, we were looking for a chlorine sensor that would directly measure chlorine levels. However, the pricing of chlorine sensors was too far out of the scope of our project, so we had to change our design for measuring chlorine. We opted to go for a Total Dissolved Solids sensor. In pools, organic matter can lead to an increase in total dissolved solids readings. Chlorine can break down organic matter which can yield a decrease in these readings. We used the TDS sensor as an indirect measure of chlorine levels. All three sensors take 5V to power, which made our design of powering the sensors easier. The code also averaged the values of 30 samples of readings before analyzing it and sending it to the dispenser unit.

The sensor board is fairly straightforward and the biggest task is to connect sensor signals to the microcontroller. Because of the need for wireless functionality, we chose the ESP32-S3 microcontroller which has WiFi/Bluetooth functionality. This way, we can communicate with microcontrollers between the sensor and dispenser board. For the PCB board, we needed the boot circuitry for uploading programming to the microcontroller, sensor circuitry according to sensor datasheets, and power supply connection and step down. The boot circuitry was taken from the ESP32 Example Board from the ECE445 Wiki page, and this circuitry allows us to control when we want to upload programming. We added the USB-UART Bridge to connect the microcontroller to a computer. For the sensors, we want connectors for the board so that we can connect and disconnect the sensors with ease. We opted for a step-down regulator from 5V to 3.3V for the power supply. The full circuitry is shown in Figure 3.

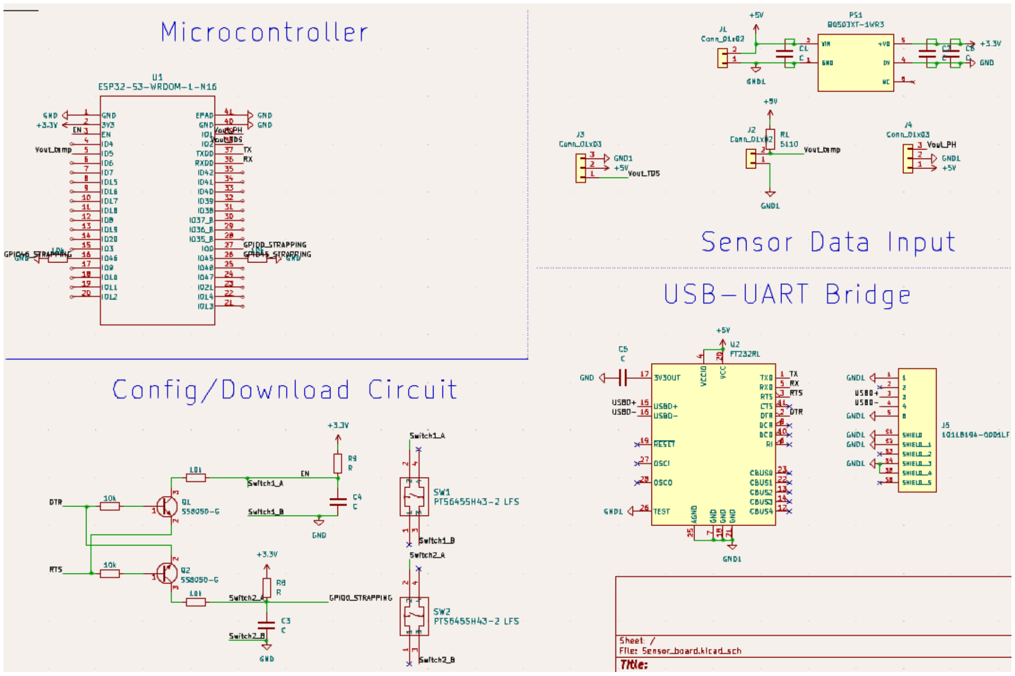

Figure 3: Sensor Board Schematic

This circuit was changed from a previous design which used a different version of the ESP32 microcontroller. As a result, the boot circuit was different, but we changed it to this current one since it was easier to interface with a computer.

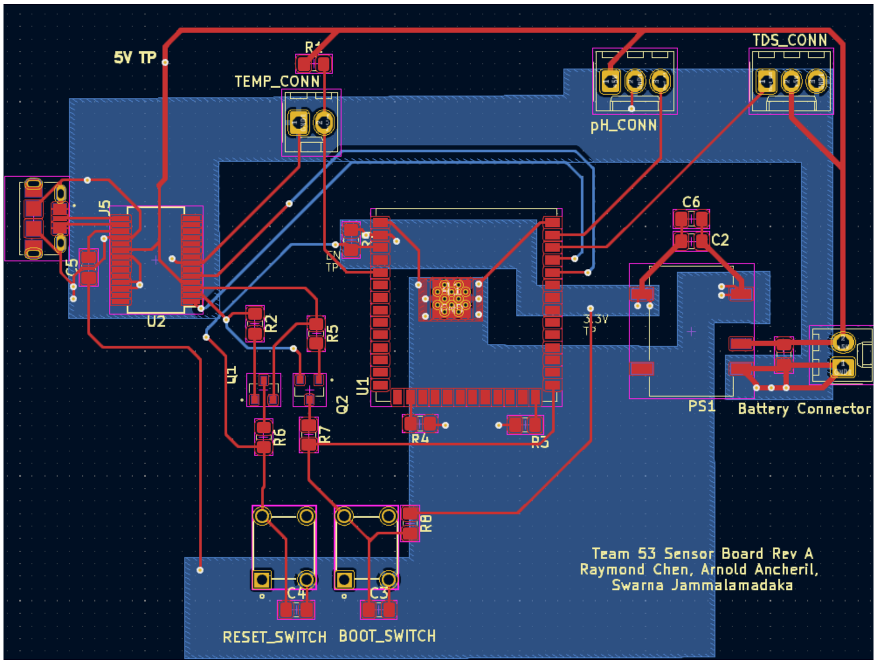

Figure 4: Sensor Board PCB

Figure 4 shows our finalized PCB design for the sensor board. Sensor connections are on the top side to make it easier to connect, and the battery connection is on the right and is close to the voltage step-down regulator.

2.2.2 Dispenser Subsystem Design

Our first challenge for the dispenser subsystem is finding the right type of motor to dispense the chemicals. We initially wanted a DC motor with encoding so that we could accurately control and track motor rotation. However, because our dispenser design only needed full rotations of the motor, we could go with less precision than what encoder motors provide. Our final design used 5V stepper motors to control the steps of rotation, but without the close-loop feedback that we would've gotten from encoder motors.

The following diagram shows our design choices for the dispenser. The three wheels are controlled by the motors in the center according to the microcontroller's instructions. The wheel has a small nook in it so that when it makes a rotation, it can collect powder when the nook is at the top and dispense it when it goes to the bottom, and be able to do this continuously.

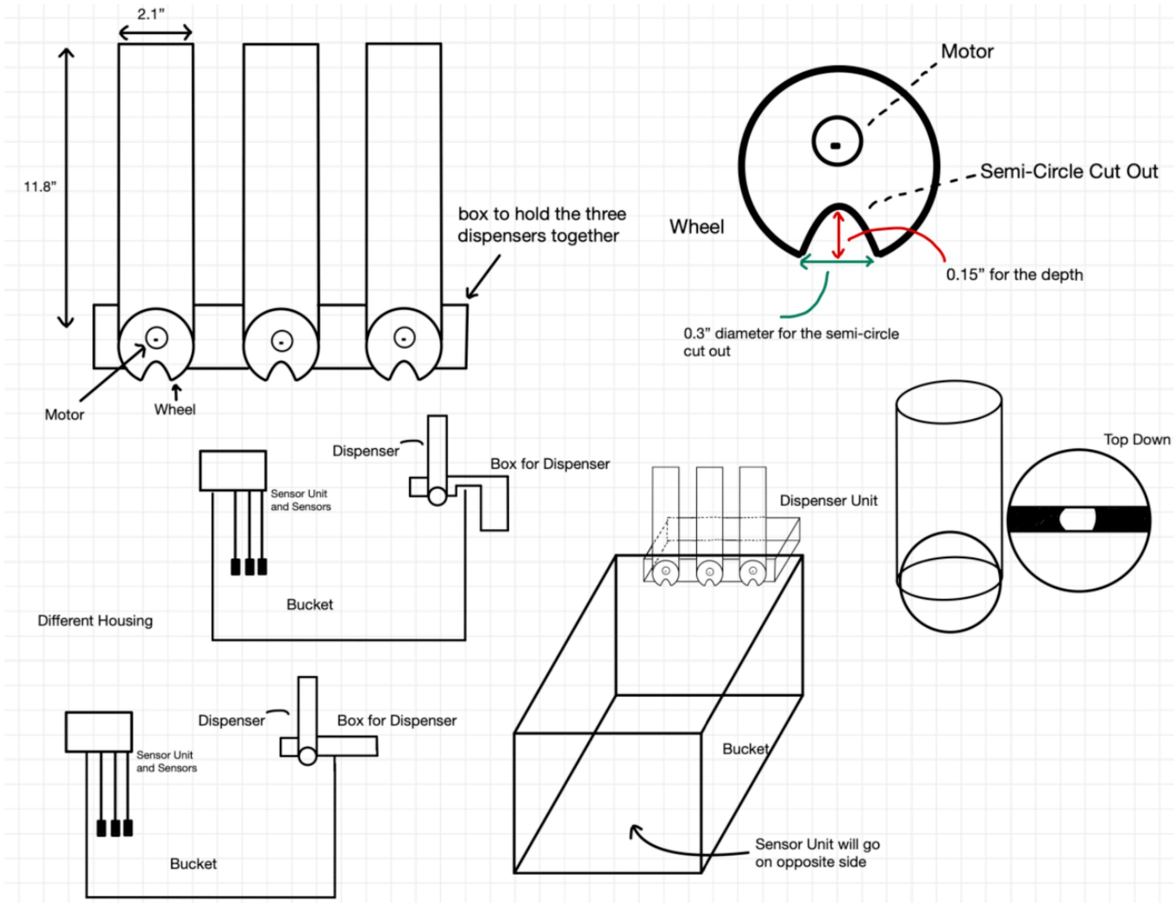

Figure 2: Dispenser Diagram

The dispenser board's biggest challenge is interfacing the microcontroller with the three motors. The design we chose uses L9110 H-bridges so we can have more precise control over the motors' speed and rotation. The boot circuitry and power supply are the same as the ones on the sensor PCB, but we added the circuitry for motor control, as seen in Figure 5. We also used a step-down regulator from 5V from the motors to 3.3V for the power supply.

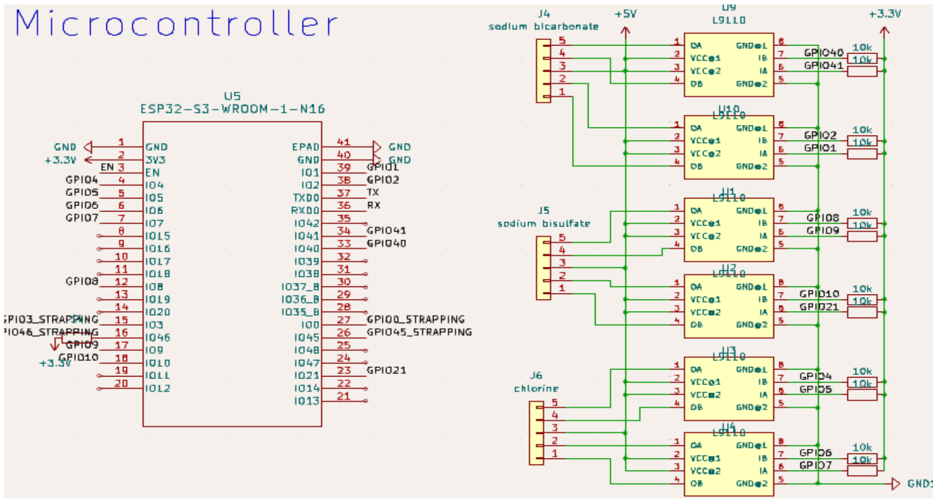

Figure 5: Dispenser Motor Circuitry

Our final PCB layout is shown in Figure 6, with motor connections at the bottom of the board.

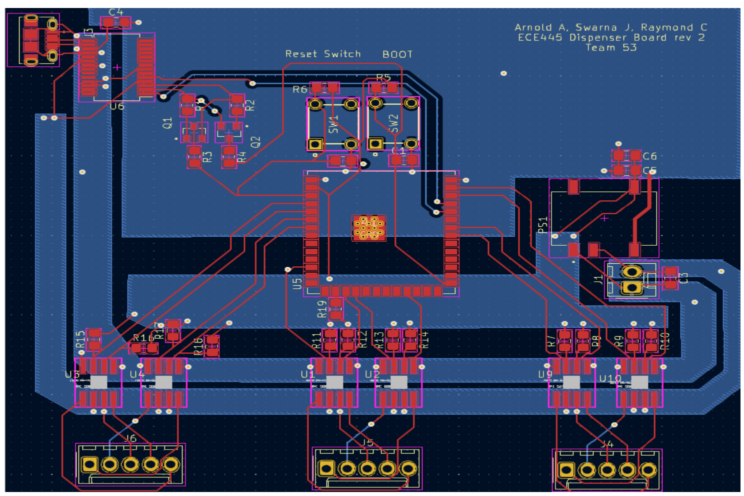

Figure 6: Dispenser PCB

2.2.3 Breadboard Design

Towards the end of our project, we could not get a working PCB. We could not establish a connection between the computer and microcontroller and as a result, we could not upload programs for the ESP32. We resorted to using a breadboard with ESP32-S3 DevKits instead to continue working on the functionality of our project. Many of our components were surface mounts which would not work with breadboards, so our designs had to change.

2.2.4 Future Modifications

There were many failures in our dispenser design that we could have changed. Since we are using one-directional and uni-speed rotation only for the motors, the H-bridge design was not needed since we do not need to change the speed or direction of rotation. Also, because the motors had a lower power rating, the motor did not have enough torque to spin in the dispenser design. Instead, we should have gotten a higher voltage or high power rating motor so that we did not have the issue of not having enough torque. Our PCB design also did not work, so we used a breadboard for the functionality.

3. Design Verification

3.1 Physical Testing

The majority of our testing is through either PCB testing or software testing. Our PCBs were designed with test points as well as surface mount components, so we tested connections using Scopy's oscilloscope to make sure all voltage levels were correct. Software testing involved outputting to a serial monitor and a physical inspection of motors spinning.

3.2 Requirements and Verification

Despite the necessity to transfer to a breadboard design, we hit most of the requirements listed in our Requirements and Verification Table in Appendix A. Sensor board requirements were all achieved, although standard deviations could be modified due to some of the nature of the sensors. For pH, the standard deviation of 0.1 values could be increased since pH values did fluctuate a lot, usually more than 0.1 points. TDS standard deviation of 100 ppm could be decreased to a standard deviation of 5 ppm since TDS readings were very steady and did not fluctuate at all. The power subsystem requirements were also satisfied in both the breadboard and PCB design. The PCB power supplies and connections were tested using a multimeter, but the breadboard power supply was satisfied through the onboard LDO on the DevKit. Finally, most of the dispenser subsystem requirements were met except for requirement 3, which is dispensing chemicals into the pool, due to the inadequate torque of the motors. While we were able to test and verify the motor spinning accuracy without the powders in the dispenser, once we added the powders, we could not test or satisfy the requirement due to the inability to spin and dispenser.

4. Costs and Schedule

4.1 Parts Cost

Below is a list of parts that were purchased for this project.

Table 1: Parts Cost Breakdown

Part #	Description	Manufacturer	Quantity	Cost	Extra Info
1	Chlorine	HTH	1	$5.55	
2	Sodium Bisulfate	Clorox	1	$9.12	
3	Sodium Bicarbonate	Arm & Hammer	1	$2.23	
4	Temperature Sensor	Honeywell	1	Borrowed	From 445 Lab
5	TDS Sensor	CQRobot	1	$11.99	
6	pH Sensor	Gravity	1	$29.99	
7	5V Stepper Motor	ECE Supply Shop	4	$7.22	
8	Battery Pack (4.5V)	Arkare	1	$7.59	
9	Voltage Step down	Digikey	3	$3.87	
10	BJT Transistors	Digikey	10	$0.29	
11	H Bridges	Mouser	10	$1.50	
12	ESP32-S3	ECE Supply Shop	2	Borrowed	
Total Cost of Parts:				**$124.86**	

4.2 Labor Cost

The labor cost estimates for our project include our three teammates and the machine shop. For the average salary of a computer engineer being $45 per hour, our labor cost is shown below:

Arnold: $45 avg salary computer engineer x (2.5 overhead factor) x (80 hours) = $9,000
Raymond: $45/hour x 2.5 x 80 hours = $9,000
Swarna: $45/hour x 2.5 x 80 hours = $9,000

Machine shop labor hours: $30/hour x 25 (hours) = $750

This brings the total cost of our project including labor and parts cost to (3 x $9,000) + $750 + $124.86 = $27,874.90.

4.3 Schedule

Table 2: Schedule

	Arnold	Swarna	Raymond
2/18	Buy Sensors/Motors by the end of the week Complete parts list for all components		
2/25	Start Dispenser PCB design, submit for audit this week or next week	Machine Shop Request for Dispenser Assembly Help with dispenser PCB	Start Sensor PCB Design and submit for audit
3/3 (first round audit)	Begin coding for dispenser board microcontroller based on documentation	Begin coding for Sensor board microcontroller based on documentation	Begin soldering onto whichever board comes first
	List errors in boards to improve for next round of audit		
3/10 (spring break)			
3/17	Dispenser PCB ordered	Sensors coding	Help with dispenser PCB
3/24	New sensor + pcb boards ordered	Sensors coding	New sensor + pcb boards ordered
3/31	(waiting on parts) PCB Debugging for new round + coding		
4/7	Coding + final Machine Shop Communications	Finishing coding	PCB received + new round ordered
4/14 - Mock Demo	(all) PCB received + soldered + debugging		
4/21 - Final Demo	(all) Testing and combining parts		

5. Conclusions

Although there were some requirements and functionalities that were not met, with the modifications and improvements listed in this report, we believe that our project will work. Our project is small-scale for the scope of the class, but we can easily scale up the size of our project to match the functionality of a real pool.

5.1 Accomplishments

On the technical aspects, we successfully collected and calculated sensor readings for our project application. We established wireless connection and communication between microcontrollers using WiFi packets. Finally, we also interfaced between the microcontroller and motors and succeeded in the dispenser motor control. Outside the physical aspect of our project, we also learned a lot about PCB designs, microcontroller programming, and soldering which is useful for lab situations in the future. We also were able to communicate well with each other and learned to compromise and come up with solutions in changing environments.

5.2 Uncertainties

Throughout the project there were a few setbacks. Due to the specific order of weeks we ordered our PCBs, the delays from previous weeks allowed us to only have three total orders. We realized that the microcontroller we originally used, the ESP32-Wroom-c6-1 was incompatible with Arduino IDE and posed some more complication when making our PCBs. Therefore, we switched to the ESP32-S3 which made our project much more manageable. However, we were only able to do one PCB order with this microcontroller.

Our dispenser also did not function completely due to the friction of the chemical powders and board. The powder was getting stuck between the wheel and wood. The motors were not strong enough to keep turning with this issue. However, stronger motors might have fixed this issue.

Our last uncertainty is how operational our product would be in a real pool. There are many other factors such as the jets and fluctuations of people swimming in the pool consistently. Therefore, the next step would be to test this in a real pool.

5.3 Ethical considerations

The main safety concerns center around improper chemical handling, potential excess chemicals, accessibility of chemicals, and cybersecurity vulnerabilities. Leaks, spills, malfunctions, or calibration errors could expose users, pets, and the environment to harmful chemicals or create dangerous pH imbalances. Easy access to these chemicals poses additional risks. While handling chemicals, gloves or some form of skin protection should be used as skin contact can be mildly harmful [2]. There can be a discussion about safety issues with dispensing chemicals into the pool where people are swimming. If there is a technical malfunction and too many chemicals are put into the pool, it could be harmful [2]. Usage of the product while people are in the pool may have some security risk. Usage of expired chemicals is also something that may need to be considered. For this product, chemicals should be replaced every year [3]. Some safety measures to counteract this are to dispense chemicals evenly across multiple locations in the pool and have the dispenser directly inaccessible to other people.

There are also safety concerns about using electricity around water, especially around pools where water is dynamic and can splash around. To combat this concern, we should protect all circuitry with water-resistant casing and make sure there are no exposed wires or circuits to the environment. We can also practice IEEE's powering and grounding electronics [4]. Any moving parts associated with the motors should also be encased to prevent injury.

Ethically, transparency about chemicals used, potential risks, and safety measures is crucial. The environmental impact of the chemicals and their potential discharge raises sustainability concerns. Additionally, the cost of such devices might create an access barrier, furthering existing inequalities. Data privacy becomes an issue if the device collects usage or chemical-level data. However, our product is not storing private information, so it is not a major concern.

Beyond these concerns, the device's reliability, maintenance needs, and user training are vital. Fail-safe mechanisms and regular maintenance are essential for safe operation. Pool owners need proper training on chemical handling, potential risks, and emergency procedures. Finally, adherence to relevant safety and environmental regulations is critical.

5.4 Future work

To extend this project further, we could test this product in an actual pool to see how well the product performs in a real-life setting. We would also use a chlorine sensor to measure the chlorine levels of the pool and dispense more chlorine when necessary. Another factor we could consider is water filtration throughout the pool. This project mostly focused on maintaining a healthy environment for pool water, it could be further extended to drinking water, as well.

References

[1] CDC. "Water Treatment and Testing." *Centers for Disease Control and Prevention*, U.S. Department of Health and Human Services, 2 Apr. 2022, www.cdc.gov/healthywater/swimming/residential/disinfection-testing.html#:~:text=As%20a%20residential%20pool%20or,friends%20from%20swimming%2Drelated%20illnesses.

[2] "Safety Data Sheet Sodium Bicarbonate," Drill Chem, https://files.dep.state.pa.us/OilGas/BOGM/BOGMPortalFiles/IndustryResources/InformationalResources/HDD_Saftey_Data_Sheets/Sodium_Bicarbonate_SDS.pdf (accessed Feb. 20, 2024).

[3] Toedter, Alicia. "How to Store Pool Chemicals the Right Way." Pool Supplies, Service & Repair, Leslie's, Inc., 20 July 2022, lesliespool.com/blog/how-to-store-pool-chemicals-the-right-way.html.

[4] "IEEE code of Ethics," IEEE Code of Ethics, https://www.ieee.org/about/corporate/governance/p7-8.html (accessed Feb. 20, 2024).

[5] Espressif Systems. "ESP-NOW." ESP-IDF Programming Guide [Online]. Retrieved From https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/network/esp_now.html

[6] DFRobot. "Gravity: Analog TDS Sensor/ Meter for Arduino." DFRobot, www.dfrobot.com/product-1662.html.

[7] "ESP32-S3-WROOM EXAMPLE BOARD: MOTOR CONTROLLER" ECE445 Wiki, https://courses.engr.illinois.edu/ece445/wiki/#/esp32_example/index (accessed Feb. 20, 2024).

[8] "Sodium bisulfate," Safety Data Sheet, https://beta-static.fishersci.com/content/dam/fishersci/en_US/documents/programs/education/regulatory-documents/sds/chemicals/chemicals-s/S25535A.pdf (accessed Feb. 20, 2024).

[9] "Swimming after Adding Pool Chemicals." *In The Swim Pool Supplies & Equipment*, In The Swim, intheswim.com/eguides/swimming-after-adding-pool-chemicals.html#:~:text=You%20may%20have%20asked%20at,wait%20between%20adding%20pool%20chemicals%3F%22&text=It%20is%20recommended%20to%20wait,after%20adding%20water%20balancing%20chemicals. Accessed 20 Mar. 2024.

[10] Amazon. *Amazon.Com*: Online Shopping for Electronics, Apparel, Computers, Books, Dvds & More, www.amazon.com/. Accessed 1 May 2024.

[11] Amazon. *Amazon.Com*: Swimming pool Care Shock, www.amazon.com/. Accessed 1 May 2024.

[12] Amazon. *Amazon.Com*: sodium bisulfate , www.amazon.com/. Accessed 1 May 2024.

[13] "Arm & Hammer Pure Baking Soda - 1lb." *Target*, www.target.com/p/arm-38-hammer-pure-baking-soda-1lb/-/A-15133726#lnk=sametab. Accessed 1 May 2024.

[14] Amazon. *Amazon.Com*: TDS Sensor, www.amazon.com/. Accessed 1 May 2024.

[15] Amazon. *Amazon.Com*: CQRobot-Ocean-Compatible-Scientific-Laboratory, www.amazon.com/. Accessed 1 May 2024.

[16] Amazon. *Amazon.Com*: GAOHOU-PH0-14-Detect-Electrode-Arduino, www.amazon.com/. Accessed 1 May 2024.

[17] 290-028 Digilent, Inc. | Motors, Solenoids, Driver Boards/Modules | *Digikey*, www.digikey.com/en/products/detail/digilent-inc/290-028/7068780. Accessed 2 May 2024.

[18] B0505XT-1WR3-TR Mornsun America, LLC | Power Supplies - Board Mount | *DigiKey*, www.digikey.com/en/products/detail/mornsun-america-llc/B0505XT-1WR3-TR/13168117. Accessed 2 May 2024.

[19] SS8050-G Comchip Technology | Discrete Semiconductor Products | *DigiKey*, www.digikey.com/en/products/detail/comchip-technology/SS8050-G/6138901. Accessed 2 May 2024.

[20] 4489 Adafruit | *Mouser*, www.mouser.com/ProductDetail/Adafruit/4489?qs=OlC7AqGiEDl68%2B2nRT7k7w%3D%3D. Accessed 2 May 2024.

[21] Esp32-C6-DEVKITC-1-N8 ESPRESSIF Systems | Development Boards, Kits, Programmers | *DigiKey*, www.digikey.com/en/products/detail/espressif-systems/ESP32-C6-DEVKITC-1-N8/17728861. Accessed 2 May 2024.

Appendix A Requirement and Verification Tables

Table 3: Water Quality/Component Sensing

Requirements	Verification	Verification Status (Yes or No)
The subsystem must be able to acquire sensor data from all sensors and transmit it to the microcontroller.	Verify that different values are being collected when the water environment changes. Accuracy will be determined by the sensor PCB system.	YES

Table 4: Sensor PCB

Requirements	Verification	Verification Status (Yes or No)
Must be able to receive and convert analog data from the sensors into the right units for temperature (F), pH (unit), and TDS (ppm) based on the respective datasheets for the sensors.	We can print values to a screen or the display to verify if the data is correct or within range. Also, acquire multiple readings of the same environment to verify that the standard deviation for temperature (±1°F), pH (±0.1), and TDS levels (±100 ppm) are within a valid range.	YES
Must be able to determine if there are any readings out of bounds. Maintain temperature between 78-82 F, pH between 7.2-7.8, and TDS below 1000 ppm.	We can artificially change the environment of the sensors to produce values that are out of range. We can print if values are out of range, or light an LED if any values are out of range.	YES
Must be able to send signals to the dispenser subsystem or computer according to calculations.	Validate from Dispenser PCB if signals are sent.	YES

Table 5: Dispenser PCB/Dispenser Unit

Requirements	Verification	Verification Status (Yes or No)
It must be able to receive signals from the microcontroller.	Validate if the data sent from microcontroller Tx is the same as the data received by the dispenser microcontroller.	YES
Must be able to move motors to dispense chemicals at the correct amount	Use a weight to determine if the correct grams of powdered chemical are released from the dispenser.	YES
The motor must be able to spin 360 degrees ± 5 degrees to maintain consistent performance for repeated actions. 0.018 cubic inches of chemicals will be dispensed every rotation, with 5% standard deviation $0.018 \pm 0.0009 in^3$.	Spin the motor for multiple rotations and see if motor positioning is significantly off or not. Verify this with a real load from the dispenser with powder on top of the shaft/disk.	YES able to spin the motor for multiple rotations and was able to check positioning, NO, was not able to check with real load
Dispenser microcontroller must delay dispensing after a chemical is released to avoid spikes in concentration due to repeated releases before water quality updates.	After a dispense, prevent another dispense for another 5 minutes (~30 minutes in normal pools) or delay sensor reading for the same amount. In normal pools, chemicals will be dispersed uniformly from the built-in pool filter, but for verification/demo, manual dispersion is needed.	YES

Table 6: Power

Requirements	Verification	Verification Status (Yes or No)
Must supply 12±0.5V of power to the dispenser unit	Use a multimeter across the ground and power supply to measure the voltage	YES
Must supply 3.3±0.3V of power to the microcontrollers	Use a multimeter across the ground and a power supply to measure the	YES

	voltage. Include test points on the board to validate trace voltages.	