***Aqua Strings*: Interactive Waterfall Display**

Final Report

EEL4914 – Senior Design I

Group 13

Department of Electrical Engineering and Computer Science

University of Central Florida

Authors:

Christopher Brown	**Mikaella Penagos**	**Trenton Morris**	**Ryan Bori**
Electrical Engineering	*Electrical Engineering*	*Electrical Engineering*	*Computer Engineering*

Reviewer Committee:

Dr. Qun Zhou Sun	ECE	Professor
Dr. Saleem Sahawneh	ECE	Lecturer
Dr. Justin Phelps	ECE	Professor

Mentor: Dr. Lei Wei

Table of Contents

Chapter 1: Executive Summary 1
Chapter 2: Project Description 2
2.1 Motivation & Background 2
2.2 Goals and Objectives 3
2.3 Project Features and Functionalities 5
2.4 Existing Products 7
2.5 Engineering Specifications Table 7
2.6 House of Quality 9
2.7 Hardware Block Diagram 10
2.8 Software Block Diagram 12
2.9 Prototype Illustration 13
2.9.1 Previous Prototype Illustration 13
2.9.2 Final Prototype Illustration 16
Chapter 3: Research and Investigation 18
3.1 Technology Comparison and Selection 18
3.1.1 Laser Harp Design 18
3.1.2 Waterproofing Options 18
3.1.2.1 Mineral Oil 19
3.1.2.2 Waterproof Enclosure 20
3.1.2.3 Potting 20
3.1.2.4 Moisture Protection (Conformal Coating/Silica Gel Packets) 21
3.1.3 Applicable Sensor Options for Water String Detection 22
3.1.3.1 Capacitive Sensors 22
3.1.3.2 Ultrasonic Sensors 23
3.1.3.3 Piezoelectric Sensors 23
3.1.3.4 Inferred (IR) Beam Sensors & IR Receiver 24
3.1.3.5 Laser Sensors & Photodiode (Selected Option) 24
3.1.4 Laminar Flow of Water Strings Options 26

3.1.4.1 Perforated Material 26
3.1.4.2 Honeycomb Flow Straighteners 26
3.1.4.3 Funnels and Nozzles 27
3.1.5 Applicable Sensors for Water Level Detection 29
3.1.5.1 Pressure Sensors 29
3.1.5.3 Float Switch Sensors 31
3.1.5.4 Resistive Sensors 32
3.1.6 Applicable Laser Beam Technology for Harp Strings 34
3.1.6.1 Diode Lasers (Point Laser) 34
3.1.6.1 Line Lasers 35
3.1.6.3 Infrared Lasers 36
3.1.6.4 Vertical-Cavity Surface-Emitting Lasers (VCSELs) 37
3.2 Part Comparison and Selection 38
3.2.1 Microcontroller Selection 38
3.2.1.1 ATmega (Arduino) 39
3.2.1.2 MSP430 39
3.2.1.3 ESP32 39
3.2.1.4 Teensy 4.0 40
3.2.1.5 Microcontroller Sound Generation Methods 41
3.2.1.6 Comparison of Microcontrollers 44
3.2.1.7 Selection of Microcontroller 44
3.2.2 Programming Language Selection 45
3.2.2.1 C/C++ with ESP-IDF 46
3.2.2.2 C++ with Arduino Framework 47
3.2.2.3 MicroPython 48
3.2.2.4 JavaScript (Espruino or Moddable SDK) 49
3.2.2.4 Rust 51
3.2.2.6 Programming Option Selection for Project 52
3.2.3 Speaker Selection 53
3.2.3.1 Passive Speaker (8Ω, 3W) 53
3.2.3.2 Active Speaker (8Ω, 3W) 54

3.2.3.3 Exciter Speaker 55
3.2.3.4 Piezoelectric Speaker 55
3.2.3.5 High Fidelity Speaker 56
3.2.4 Water String Detection Laser & Sensor Selection 58
3.2.4.1 Laser Beam Component Selection 58
3.2.4.2 Photodiode Component Selection 61
3.2.5 Water Level Sensor Selection 63
3.2.5.1 Part 1528-4965-ND 64
3.2.5.2 Part 42AK5700 64
3.2.5.3 Part 2010-1043-ND 64
3.2.5.4 Part B07G13JB57 65
3.2.6 Water Pump Selection 66
3.2.6.1 Condensate Water Pump 67
3.2.6.2 Submersible Water Pump 68
Chapter 4: Standards and Design Constraints 70
4.1 Design Standards 70
4.1.1 PCB Design Standards 70
4.1.1 Soldering Standards 71
4.1.2 IEEE Standards 71
4.1.2.1 IEEE Validations Standards 72
4.1.3 Safety Standards 72
4.1.4 IEC Standards 73
4.1.5 Audio Standards 73
4.1.6 ANSI/H1 Submersible Water Pump Standards 74
4.2 Constraints 74
4.2.1 Time Constraint 74
4.2.2 Economic Constraint 75
4.2.3 Environmental and Sustainability Constraints 76
4.2.4 Political Constraints 76
4.3.5 Safety Constraints 77
4.3.6 Manufacturability Constraints 77

Chapter 5: Comparison of ChatGPT with other Similar Platforms 78
5.1 Introduction to AI Platforms in Engineering Design 79
5.2 Comparative Analysis of AI Platforms 80
5.3 Integration with Engineering Workflow 87
5.4 Limitations and Potential Drawbacks 88
5.5 Best Practices for AI Integration in Engineering Design 89
5.6 Conclusion 90
Chapter 6: Hardware Design 91
6.1 Design Overview of Aqua Strings Hardware System 92
6.2 Power Delivery & Management 96
6.3 Laser Diodes and Photodiodes 98
6.4 ESP-32 MCU Design 101
6.5 Amplifier and Speaker Design 103
6.6 MicroSD Card Module 105
Chapter 7: Software Design 107
7.1 Main Program Design: 107
7.2 Use Case Diagram 108
7.3 System State Diagram 109
7.4 Planned Software Features and Stretch Goals 110
7.5 SD Card Audio Handling and Real-Time Playback 111
7.5.1 Audio File Organization 111
7.5.2 Streaming and Buffering Strategy 111
7.5.3 Responding to Polyphonic Playback 112
7.5.4 Intelligent File Selection 112
7.5.5 Error Handling and Fallbacks 112
7.6 Photodiode Calibration and Ambient Light Compensation 112
7.6.1 Calibration Procedure 112
7.6.2 Software Strategies for Reliable Sensing 113
7.6.3 Adaptive Calibration and Dynamic Adjustment 113
7.7 Software Design for Active Water Flow Regulation 114
7.8 Software Design for Concurrent Task Execution on the ESP32 115

7.8.1 Non-Blocking Execution Patterns 115
7.8.2 State Machines for Subsystem Control 115
7.8.3 Task Segregation with FreeRTOS and Dual-Core Utilization 116
7.8.4 Synchronization of Shared Resources 116
7.8.5 Software Timers and Event Scheduling 116
7.9 Software Stress Testing and Reliability Evaluation 117
7.10 Version Management and Team Collaboration with GitHub 118
Chapter 8: System Fabrication and Prototype Construction 119
8.1 PCB Design 119
8.1.1 Regulator PCB 120
8.1.2 Main Microcontroller PCB 121
8.1.3 Audio Amplifier PCB 122
Chapter 9: System Testing and Evaluation 124
9.1 Testing and Overall Integration 124
9.1.1 ESP32 124
9.1.2 Laser and Photodiode Sensor 125
9.1.3 Speaker and Amplifier 127
9.1.4 Water Pump 128
9.2 Plan for Senior Design 2 129
Chapter 10: Administrative Content 131
10.1 Budget and Financing 131
10.2 Project Milestones for SD1 and SD2 132
10.3 Work Distributions 134
Chapter 11: Conclusion 136
Appendix A: References 138
Appendix B: Copyright Requests 142
Appendix C: Copyright Approvals 143
Appendix D: Data Sheet 144
Appendix E: Software Code 145

List of Tables

Table 2.5.1: Engineering Specifications Table 8
Table 3.1.6.4.1: Waterproof Technology Comparison Table 21
Table 3.1.3.5.1: Sensor Comparison 25
Table 3.1.4.3.1: Laminar Flow Specification Comparison 29
Table 3.1.5.4.1: Water Level Sensor Comparison Table 33
Table 3.1.6.4.1: Laser Technology Comparison Table 38
Table 3.2.1.5.4.1: Comparing Microcontroller Audio Systems 43
Table 3.2.1.6.1: Comparing Microcontrollers 44
Table 3.2.2.1: Comparison of Programming Languages 45
Table 3.2.3.5: Comparison of Speakers 57
Table 3.2.4.1.5.1: Laser Comparison 60
Table 3.2.4.2.4.1: Photodiode Comparison 63
Table 3.2.5.5: Water Level Comparison 65
Table 5.2.1: Comparison of LLMs 80
Table 10.1.1: Budget and Financing 131
Table 10.2.1: Senior Design 1 Milestones 133
Table 10.2.2: Senior Design 2 Milestones 133
Table 10.3.1: Work Distribution Layout 135

List of Figures

Figure 2.6.1: House of Quality 9
Figure 2.7.1: Hardware Diagram 10
Figure 2.8.1: Software Block Diagram 12
Figure 2.9.1.1: Prototype Illustration 1 13
Figure 2.9.1.2: Prototype Illustration 2 14
Figure 2.9.1.3: Prototype Illustration 2 Side View 15
Figure 2.9.2.1: Prototype Illustration 3 Front View 16
Figure 2.9.2.2: Prototype Illustration 3 Side View 16
Figure 5.2.2: Example Software Diagram 85
Figure 6.2.1: Internal schematic of the LM2576 regulator 96
Figure 6.2.3: The schematic of 3.3V and 5V regulator plug ins and 12V barrel plug on main board 98
Figure 6.3.1: Laser diode schematic with MOSFET on main board 99

Figure 6.3.2: Photodiode schematic on main board 100
Figure 6.4.1: ESP-32 module schematic/ layout..... 102
Figure 6.5.2: Efficiency v. Output Power graph of PAM 8403..... 105
Figure 7.1.1: Main Program Block Diagram..... 107
Figure 7.2.1: Use Case Diagram 108
Figure 7.3.1: System State Diagram..... 109
Figure 7.5.1.1: SD Card Audio System Architecture..... 111
Figure 7.6.3.1: Photodiode Calibration and Ambient Light Compensation..... 113
Figure 8.1.2.1: PCB Design of Microcontroller Board..... 122
Figure 8.1.3.1: PCB Design of Audio Amplifier Board 123
Figure 9.1.2: Prototype of Audio Triggering with Laser-Photodiode System ... 127

Chapter 1: Executive Summary

The Aqua Strings Interactive display project is an interactive, immersive system for generating an enticing, immersive water effect that responds to the user's interaction. At its core, the project merges both the visual aesthetic of water in motion with the audio aesthetic of music to provide a multidimensional experience that is enticing and active. The system is powered by water streams that are triggered in synchronicity with music. Once a user's hand interrupts the laser beams, these personalized triggers change the water streams whenever a user's hand interrupts the laser beams. Having this ability allows for a different form of interaction to take place since the water streams and music react to the individual user's movements in real-time. Thus, when the user's hand interrupts the triggering laser beams, the sensor performs the final and most specific iteration of a user-controlled interface. Laser detection sensors are a precise, reliable way to detect interruptions to trigger water flows and audio elements in harmony, which creates a visually and audibly stunning atmosphere.

The overall system combines a variety of different technologies to operate and provide the intended user experience. Precision water pumps and sensors are used to help control the water level and flow so that the display is constantly in operation and remains visually appealing. The water flow is electronically and finely controlled to ensure the fullness of each water string without unwanted gaps. Further, a variety of sensors monitor and control the water height within each reservoir to ensure continual operation within an acceptable and optimal water height range to prevent any overflow or underflow from occurring. This allows the display to be visually and functionally efficient.

Aqua Strings combines a person's physical touch with the digital world to create an enjoyable user experience. By detecting an interruption in a water string, the system will be able to respond accordingly to the specific user input at the specific water string. This project is more than an entertainment system, it demonstrates how technology can seamlessly blend with one another to create a more in-depth project than any one component alone.

Chapter 2: Project Description

Aqua Strings encompasses the combination of real time audio system and sensor integration to develop an engaging musical machine. By utilizing some novel approaches and embedded systems, digital signal processing and sensor technology, this unique musical machine can prompt users to interact in a more modern way by replacing traditional strings with laminar flows of water. The system has pairs of laser photodiodes to recognize brakes in the laminar flow of water caused by hand interactions, which a microcontroller processes and outputs a sound. The project incorporates principles of fluid dynamics, optoelectronic sensing, and digital audio synthesis to create a new platform for multimodal systems that are dependent on user input/ experience. Design with accessibility and innovation goals for inclusive interaction in mind allows Aqua Strings to be experienced by all. Directions within engineering have been steered toward issues of sustainability, immersive user experiences, as well as designing devices for human interaction and this project only supports that.

2.1 Motivation & Background

Over the years, the availability and concept of modern technology is constantly evolving, which allows individuals, such as engineers, to improve existing ways of life. One of these areas of life that is constantly evolving is the entertainment and music industry through how content is created, distributed, and experienced. These advancements allow for vast audiences to immerse themselves within a given culture and/or gain a sense of enjoyment in their day-to-day lives. More specifically, these technical advancements consist of new sensors, Digital Audio Workstations (DAWs), software instruments, and even AI in music composition. Through these advancements we can do a multitude of projects, one of them involves creating a water harp/ interactive waterfall display. The concept of a “water harp” combines the practical use of a traditional harp with the new visual water element which elevates the user and audience's experience.

Water harps, also referred to as water-based instruments, allow for a multisensory experience employing different modes and tones since the sounds are now electronically generated compared to the vibrations from a traditional harp. Whether people like to acknowledge it or not our society revolves around music whether it's listening to music in the car, while doing work or eating in a restaurant. Music is an experience that we all subject ourselves to daily, so why wouldn’t we look to evolve and immerse ourselves further in the experience? Since a water harp would consist of electronically generated sounds, it makes the machine more accessible to younger children of any person that doesn’t have the musical knowledge needed to tune and play a traditional string harp. In addition, this project also affects individuals with hand disabilities and are unable to pluck traditional strings. Aqua Strings would serve as a great solution for these individuals to continue playing music.

Besides the artistic value of the project, it also stands for an innovation of technological integration and a sense of sustainability. While water is used both as the working medium and as an aesthetic component, the integration of natural elements in a technologically heavy project such as this joins two groups together. The water harp combines modern engineering with creativity while focusing on a user-friendly experience.

2.2 Goals and Objectives

Our Interactive Waterfall Display project aims to integrate interactive technology into an artistic experience to further develop an existing instrument into a multisensory machine by integrating sound, visuals, and user interaction. By achieving our primary goals, we hope to create a more enjoyable musical experience.

The primary (basic) goals of our project are as follows:

- Design and construct a technologically advanced interactive instrument using a multitude of different sensors that will read user input and translate this information into a musical sound/note.
- Create 3 fully functioning strings of water that deal with only three notes and demonstrate how the other component of the project works.
- Implement an efficient real-time signal processing system to achieve responsive reactions with minimal latency, ensuring the expected result from the user's inputs (such as breaking a water string) in correspondence with the desired audio output.
- Maintain a constant, continuous water flow so there is no undesired break in the water (water pump calibration). This will also require additional water level sensors to be installed in each reservoir to make sure there is a constant flow of water through each string.
- Provide a multisensory user-experience involving the necessary audio tones/ notes, visuals (the colors of the laser sensors projected through the water string), and the actual tactile water element creating a laminar flow.

The advanced goals of our project are as follows:

- Complete 8 (5 additional) fully functioning strings of water, so that each note in a musical scale can be accounted for.
- Create different modes consisting of OFF, FOUNTAIN, and HARP. OFF halts all components of the project, FOUNTAIN allows the water to flow without an interactive component, and HARP adds the interactive component to the flowing water.
- Using either solenoid values or a sliding component to cut off the streams off water when needed/ desired.

The stretch goals for our project are as follows:

- Create different modes (possibly three) for different electronic sounds to be played when a string is broken. Modes could include musical, synthesized, etc. If not, we could instead have three different octaves of notes, low, middle, and high.
- Integrate small water turbines at the bottom of the water strings involving a DC motor and a 3D printed curved cup rotating component (Pelton Turbine).
- Using the values mentioned in the advanced goals, we could have motorized sequences that would create breaks in the water at specific times to have a water display without human interaction.
- Install a laser horizontally across the strings of water that would detect hand movement and play different notes for the same string of water depending on if the stream is broken higher or lower.

The overall objectives consist of fulfilling the listed goals from above and then some. The process and techniques we plan to use are subject to change as we research, and new information is discovered.

Our listed objectives are as follows:

- Correct implementation of our hardware design:
    - Incorporate accurate sensors such as laser diodes with photodetectors for interruption detection in water streams to provide steady and accurate input signals.
    - Design a water-pump system capable of generating a uniform, laminar flow for the water strings whose flow rate will vary for dynamic interaction.
- Development of firmware and microcontroller interface:
    - Program microcontroller (MSP430 or ESP 32, not decided yet) to process the inputs from the different sensors and map the user interactions to the necessary MIDI or audio files.
    - Utilize precising timing components, interrupt-driven processing, and other efficient aspects offered by the microcontroller.
- Audio sequence development and implementation:
    - Implement a module of sound generation that is capable to produce customizable tones as desired.
    - Employ DSP techniques to control the audio output to manipulate pitch, amplitude, and waveform shaping depending on the specific audio device chosen (utilizing a speaker).
- Closed loop, recycling water system:
    - Design a water management system that recirculates the water through the machine efficiently, reducing waste and the need for added resources.
    - Use flow sensors/ water level sensors so the system knows whether a specific reservoir is low or filled with water.
- Rigorous testing of the prototype and final project:

    - Ensure all components are acting as intended with little to no errors or mishaps.
        - Ex: sensor accuracy, response time, signal mapping, audio quality, overall latency between different systems
- Hardware safety/ waterproofing:
    - Waterproof will include, but not limited to, silicone casing for specific components to ensure separation of electrical systems from the water recirculation system.
    - Housing all possible electrical components of the machine above the upper water reservoir to ensure the crucial electrical components do not interact with the flowing water.

From this breakdown of our goals and objects we now have a check list of checkpoints we want/ need to complete for a functioning interactive waterfall display. The expected outcome of this project is to provide our team with actual design experience while also being able to produce a product that is innovative and efficient with a friendly user interface.

2.3 Project Features and Functionalities

The Aqua Strings: Interactive Waterfall Display project brings together the best of modern technology with sensitive designs for an interactive and captivating experience. The machine serves as a water harp and can read user inputs in the form of movement, as the user's fingers or hand pass through the flowing strings of water, causing a note or sound to play out of a speaker embedded in the machine. There is also a dial that switches the modes of OFF, FOUNTAIN and HARP that the user can use at their discretion. Our team will be utilizing both previously existing elements, as well as our own physical design and technical features to create a project that is versatile and unique.

Our project houses a custom engineered water pump with the ability to generate laminar water flow for our harp strings. These strings will keep a uniform look and have a sense of stability so the precise 'pluck,' interruption, of the water string can be detected. The pump system will be designed such that there is a constant and continuous stream of water flowing as our water harp strings since any break in the water can be interpreted as an interruption. This water management system also operates with low power consumption to conserve its energy pull while also maintaining its reliable performance.

Aqua Strings relies on laser-based sensing for the detection of any interactions done by the user. It has laser sensors positioned downward shooting through the water stream to a photodetector, when the laser is detected then the string is unbroken and when it isn't detected then there is an interaction that needs to be noted. This noted occurrence will then translate the input into a musical, audio tone to be played through the speaker. This setup allows for an elevated level of accuracy and real-time sensing, since the robust calibration of these sensors will reduce the possibility of false triggers that could occur for numerous reasons,

including the displacement of water done by the user's hand or changes of the lighting conditions in the room the device is in. This ensures there are little to no undesired outcomes, facilitating a flawless experience for the user.

The project also involves a sophisticated audio generation system that maps sensor inputs to digital notes that are saved on file. Users will be able to modify and adjust the different modes of the audio output, whether that entail different sounding notes or different octaves (not determined yet). In addition to this audio display, the laser sensors will also have a color being refracted through the beam, color dependent on the specific sensor we acquire and the wavelength of the laser, to add to the visual component of the project.

A closed loop, recirculating water management system is also embedded into the design of the machine allowing for sustainable operation by collecting and redistributing the water. Flow monitors, consisting of water level sensors, are used to track water levels in the upper and lower reservoirs ensuring consistent operation of the machine free of unnecessary interruptions. The system design is intuitive since it takes constant inputs which turns the water pump on or off in relation to water levels preventing overflows.

Safety and waterproofing of all systems are imperative as electricity and water aren't a desired mix in our project, especially if users are touching the water strings. All electronic parts are either isolated or encased to prevent users from encountering hazards. It would be a good idea to include sensors to either measure current being passed through the water and/or other overload protection systems to avoid mechanical and/or electrical failures.

Aqua Strings delivers real-time functionality with virtually immediate responses to the inputs detected in the machine. These inputs include water streams breaking causing tones to play, the switching between modes that change the sound of the notes played, and the monitoring of the water levels within the reservoirs. Little to no latency is a major component for this project because without it the project would be inconsistent and not act as expected.

Our project takes our designs of efficiency and user-friendly experience into one product. It operates in a simplistic way for the user and is a bit more complex on the backend but allows for an innovative redesign of a traditional musical instrument. Through concepts of sustainability to progressing technology, our project delivers an in-depth experience unique to this interactive machine.

is a machine that will simply serve as a water harp. Our machine will be able to read user inputs in the form of movement, as the user's fingers or hand pass through the flowing strings of water, causing a note or sound to play out of a speaker embedded in the machine. Our team will be utilizing both previously existing elements, as well as our own physical design and technical features to create a project that is versatile and unique.

2.4 Existing Products

Thread fountains are a type of water display that is available in today's market and are typically seen through modern and artistic sculptures. Being the focal points in many public spaces such as hotels, offices, or art galleries, the idea for a thread fountain is to have strings of water flow straight down in a linear cascade-like effect. There are a series of strings fixed from the top to the bottom of the fountain, which serve as a guide for each string of water to follow. The strands of thread in the fountains are one of the most important components of achieving the highly smooth illusion of the water flow. Thread fountains also vary in size, shape, and complexity. Some designs have up pre-programed patterns and effects to add another layer of visual entertainment to the viewer. Through the utilization of motors, magnets, and LED lights, thread fountains can achieve many types of visually artistic and entertaining effects and patterns.

An electric harp is a modern extension of the musical instrument, where instead of a musician having to physically pluck harp strings to generate a sound, the strings are replaced with sensor technology to detect finger movements along the empty space. Through various motion sensors along the harp frame, hand movement detected from a set sensor can be translated into a specific note played through a built-in sound system. Numerous adaptations of electric harps have been created and are in current use, primarily for art and entertainment purposes. Electric harps can range from having lasers serve as strings or having invisible string as a whole. Electric harps are not as widely seen or utilized in the professional music world, though they are seen to become more widespread for musicians who may have mobility limitations and are looking for a more accessible instrument to play.

Our team gathered inspiration through both thread fountains and the electric harps. We wanted to design a project that has an element of art and entertainment as well as incorporating music and sound. Motion sensor-based harps can be seen to exist, however adding the element of water adds a unique sensory experience that has rarely been done before. In addition, the thread fountains that are in use today rely on the placement of the string for the water to flow. Aqua strings will have free flowing water drop from the top to the bottom of the structure without the utilization of a thread. This way, the generation of the sound will be solely reliant on the motion sensors rather than the physical pulling of a string. Our project will also utilize elements of both a harp and fountain through having different modes to switch to. Having this versatility adds another layer of uniqueness and creativity to our project, which has not been seen with previously existing products.

2.5 Engineering Specifications Table

The Engineering Specifications Table shows the key parameters and performance features of the system devices. The detection time and response time of the photoresistor must be high for efficient detection of light. The MCU and speaker and photoresistor have to respond instantaneously to inputs from the system. For the level sensors and water pump, the system must be capable of providing a

specified water level within a certain range for efficient water display control. The laser in the system must be low power and distinct wavelength for safety and functionality. The pump must be capable of providing a stable flow rate and sufficient lift height to attain the water display without overloading the system. Besides, the reservoirs are built in a way that they have enough water volume to allow the operation of the system.

Table 2.5.1: *Engineering Specifications Table*

Component(s)	**Parameter**	**Specifications**
Photoresistor	Detection Time	< 10 ms
Photoresistor, MCU, Speaker	Response Time	< 100 ms
Water Pump, Level Sensors	Water Level Maintenance	< 2 in of water
Laser	Wavelength	650 nm (Red)
Laser	Power	< 5mW
Pump	Flow Rate	2.3 gallons / minute
Pump	Lift Height	< 1.5 m
Reservoirs (Upper/Lower)	Capacity	2 gallons each

2.6 House of Quality

Correlations	
Positive	+
Negative	−
No Correlation	

Relationships	
Strong	●
Moderate	○
Weak	▽

Direction of Improvement	
Maximize	▲
Target	◇
Minimize	▼

Column #	1	2	3	4	5	6	7	
Direction of Improvement	▼	▼	▼	▼	◇	▲	▲	
Customer Requirements \ Engineering Requirements	Cost	Weight	Detection Time	Response Time	Sustained Water Height	Sensor Accuracy	Safety	Row #
Cost	●	▽	●	●	▽	●	○	1
Portability	●	●	▽	▽	●	▽	○	2
Ease of Use	●	○	●	●	▽	●	●	3
Reliability	●	○	●	●	○	●	●	4
Safety	○	▽	▽	▽	○	▽	●	5
Sound Accuracy	●	▽	●	●	▽	●	▽	6
Low Maintenance	●	○	○	○	▽	○	●	7

Figure 2.6.1: *House of Quality*

2.7 Hardware Block Diagram

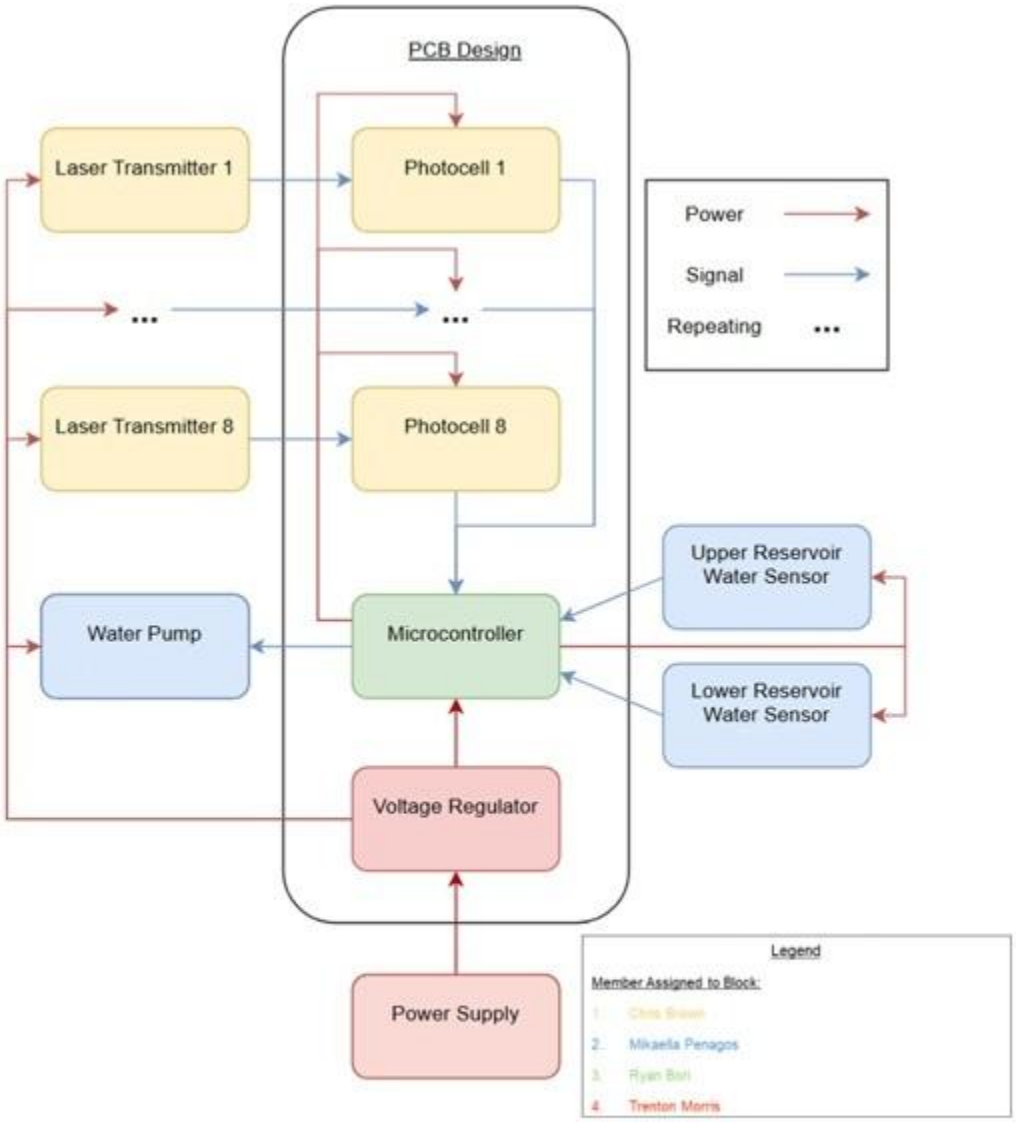

Figure 2.7.1: *Hardware Diagram*

Components' Inputs and Outputs

- Power Supply:
  - Input: External power (adapter/outlet) or battery.
  - Output: Power to all components through regulator or microprocessor.
- Microprocessor:
  - Input: Signals from photocells, water level readings, and switch knob.
  - Output: Control signals for MIDI sounds, playback, and pump system.
- Speaker:
  - Input:
  - Output: Sound waves corresponding to
- Photocell(s):
  - Input: Infrared laser beam (unless interrupted by hand).
  - Output: Signal to the microprocessor indicating an interruption.
- Laser:
  - Input: Power from power supply unit/voltage regulator.
  - Output: Continuous infrared beam.
- Pump System:
  - Input: Power from power supply unit/voltage regulator.
  - Output: Water flows to the upper reservoir.
- Reservoirs (Upper and Lower)
  - Input: Water from the pump (upper reservoir) and falling water (lower reservoir).
  - Output: Water columns (upper reservoir) and collected water for recirculation via pump (lower reservoir).
- Water Level Sensors:
  - Input: Power from microcontroller and stimulus of water.
  - Output: Signal for when the water level reaches a certain threshold.

2.8 Software Block Diagram

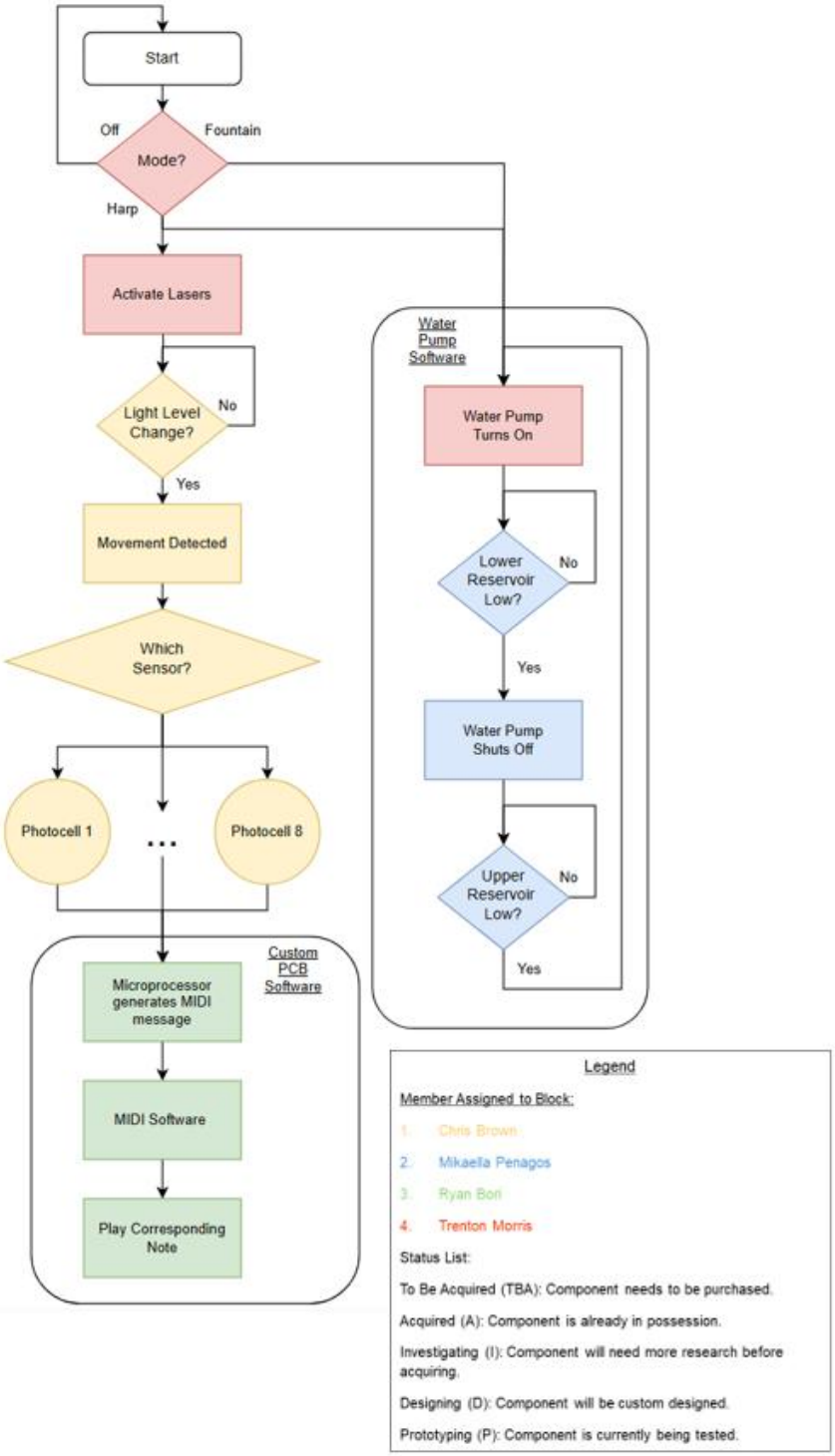

Figure 2.8.1: *Software Block Diagram*

2.9 Prototype Illustration

2.9.1 Previous Prototype Illustration

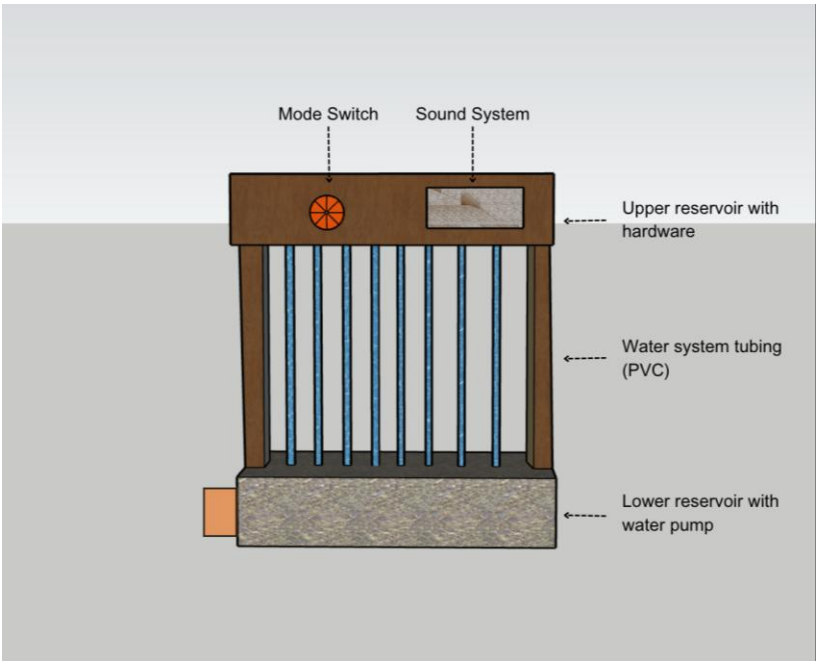

Figure 2.9.1.1: *Prototype Illustration 1*

The prototype illustration in the figure above displays our initial proposed design of project Aqua Strings, highlighting some of the key components of the construction and organizational structure. Firstly, the frame of the water harp can be seen to be made from a sealed wood finish or a waterproof material. The primary purpose of this framing material is to create durable and protective housing for the sensitive components of the project, including the water systems and electronic hardware. By ensuring a sealed and structured enclosure, the design focuses on preventing potential water damage and maintaining a safe environment for the electronic systems while preserving the aesthetic appeal of the instrument.

The frame is also designed to support the considerable weight of multiple water reservoirs, facilitate vertical water movement through the waterfall and pump system, and accommodate the battery, pipes, sensors, electronics, and sound system. The choice of materials and compartmentalization of the frame not only provides structural integrity but also contributes to the ease of maintenance and safety of the project and users.

The harp features an upper reservoir which houses the bulk of our mechanical elements. This section houses critical hardware, including the motion sensors responsible for detecting interactions with the water strings, a mode switch that offers user control over the instrument's settings, and the sound system that produces audio feedback in response to the water being 'played.' The interior of the upper reservoir is divided into two distinct compartments: the bottom section

contains the mechanical tubing and water pipes, while the top section securely holds water-sensitive electronics and hardware. This compartmentalization will be crucial to preventing water from meeting delicate components, thereby enhancing the safety and longevity of the system.

The separation of wet and dry zones within the upper reservoir creates a smooth and safe water flow while also maintaining the visual and functional integrity of a traditional harp instrument. This design decision also allows for a smoother visual, helping the water harp maintain its resemblance to the real string instrument.

The lower reservoir primarily functions to collect the falling water and recycles it back to the upper reservoir using the water pump at the bottom left portion of the frame. The lower section of the frame is designed to be as large as the top section to achieve a balanced distribution of weight and stability. Given the significant weight of the water pump and the volume of water required for the Aqua String's operation, maintaining this balance is important for both safety and performance. The bottom compartment's size also accommodates potential water splashes from the waterfall and provides additional space to manage excess water.

Lastly, the water system uses PVC tubing to channel the water from the pump to the upper reservoir efficiently. The piping is enclosed along the left column of the frame, facilitating a consistent and controlled flow of water while preventing overflow in the lower reservoir. The estimated height of the harp is approximately two feet, in order to have a balance between stability and portability. This design choice makes the harp ideal for offering ease of transport while retaining its full functionality as an interactive musical instrument.

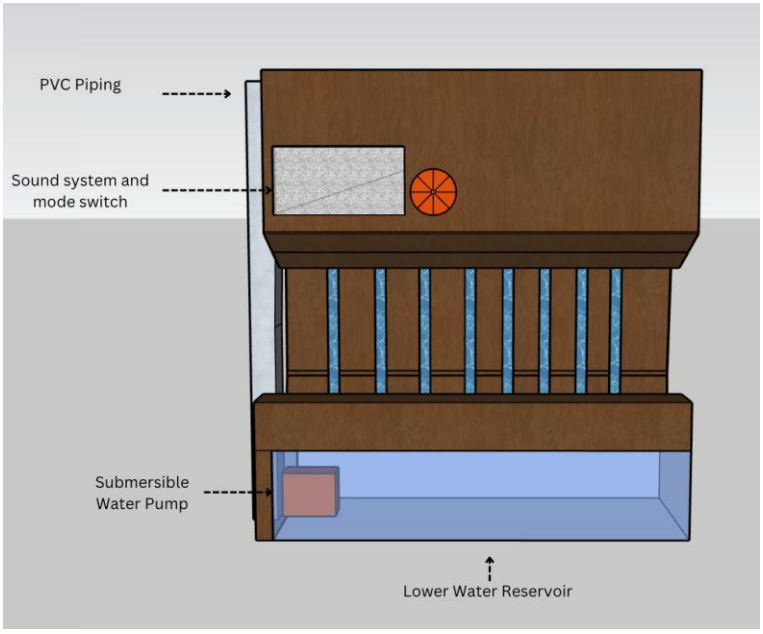

Figure 2.9.1.2: *Prototype Illustration 2*

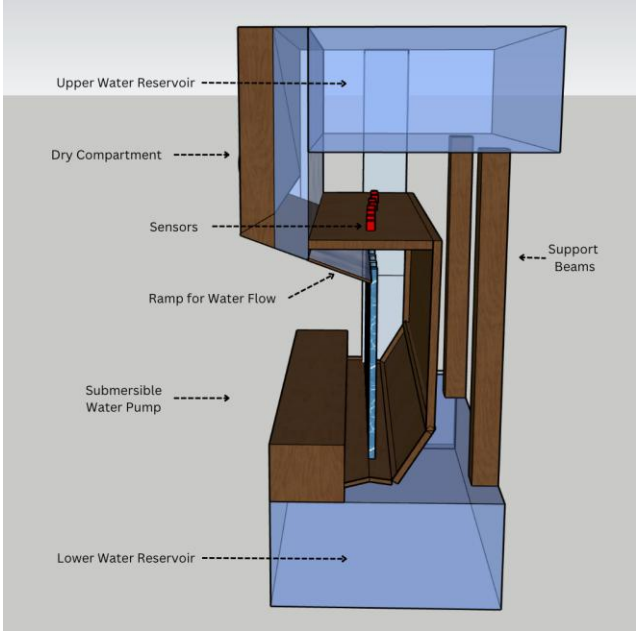

Figure 2.9.1.3: *Prototype Illustration 2 Side View*

The diagrams above highlight the second prototype of our project, with a slightly different approach. The most prominent change with this is water flow and a structured implementation of the lasers at the top of the structure. A dry compartment on the left side houses the electronic hardware, such as our microcontroller, speaker, mode switch, and all added wiring for the musical component of Aqua Strings. This try component is sealed and a separate space from the water, to ensure a fully safe and distant area for wet and dry spaces. Unlike the original idea of having the funnels of water directly stemming from the upper reservoir, this prototype includes a slight downward ramp at the top, directing water downward to then funnel into eight distinct water strings. Most importantly, this structure allows the lasers to be at the perfect angle to point down directly through each water stream. The ramp implementation is also put in place to assist in achieving laminar flow, which is just as critical for having the lasers work accurately. All together, these controlled streams allow for a structured and playable interface, where interaction occurs when a user disrupts the water flow.

As the sensor detects an interruption in the laser, this action triggers a corresponding note to be played for each string, with the end goal of achieving a scale with each note. On the back end of our structure, there are support beams to provide stability to the structure. As there is a large area of water at the top along with a ramp on the front end of the project, having added support beams on the back is important in maintaining the proper alignment of the water flow and sensors, as well as structural integrity of the frame. The closed-loop water circulation system is achieved through the PVC pipes, where the submersible water pump will transport the water from the lower reservoir to the top reservoir. Overall, these few additions and specifications from the original prototype are beneficial in achieving our fully functioning project.

2.9.2 Final Prototype Illustration

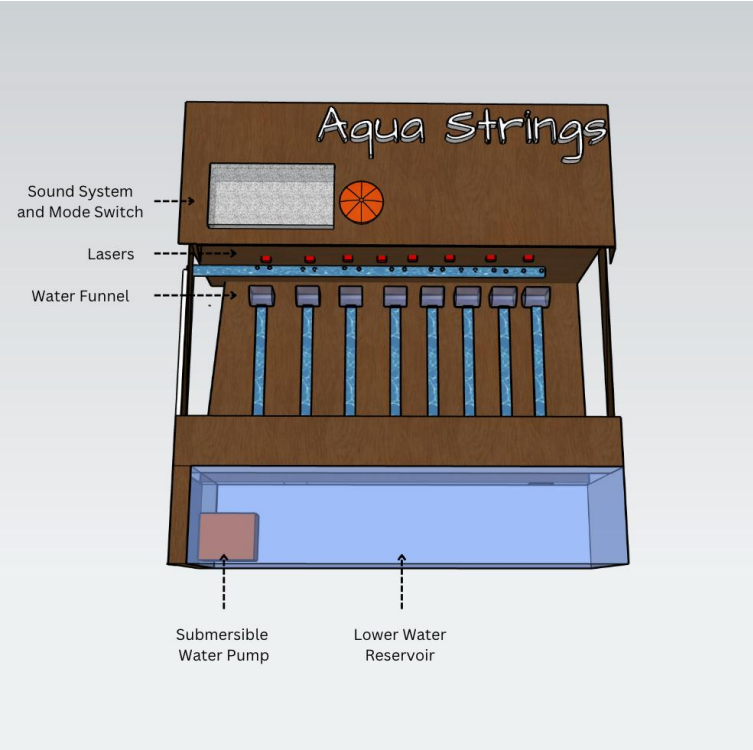

Figure 2.9.2.1: *Prototype Illustration 3 Front View*

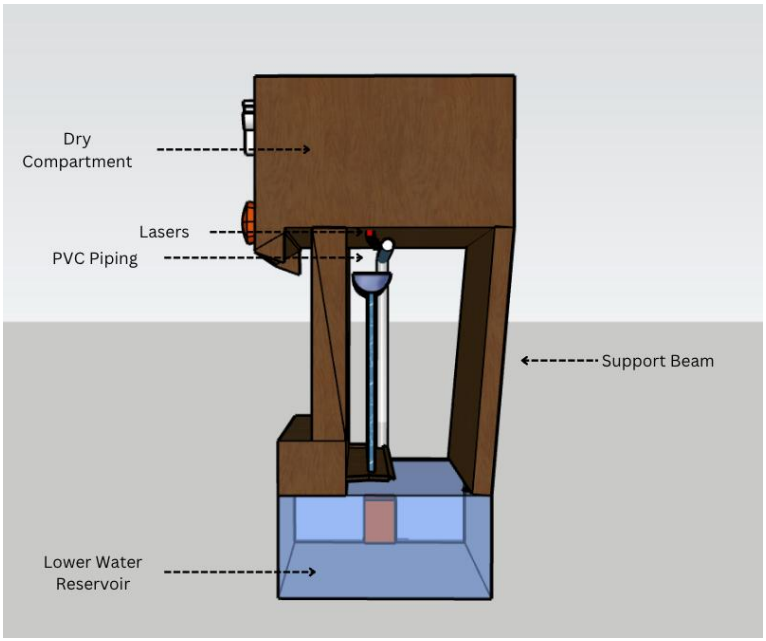

Figure 2.9.2.2: *Prototype Illustration 3 Side View*

In our groups third and final prototype, there are numerous changes that have been made to the structure of our project. This model also incorporates a design involving sensors to detect hand movement, however a few areas were

reorganized to make the fountain more optimal. Firstly, this prototype compares with the other two ideas, as each system has a lower reservoir with a submersible pump that continuously recirculates the water, and this serves the purpose of constantly providing water flow through the structure. Additionally, this structure incorporates an upper section that distributes the water into individual waterfalls, and this follows the same idea from the past two designs.

However, the water distribution of this third design differs. In the first prototype illustration, water is pumped directly into an upper reservoir, from which it falls freely into the eight water strings. This method relies on gravity for even distribution and a natural cascading effect. On the other hand, the finalized model does not have an upper water reservoir. Instead, it utilizes a PVC pipe with pre-drilled holes that direct water into a cylindrical funnel before it flows down the strings. This alternative method provides a more controlled distribution of water compared to the free-fall design we had originally planned.

Another key difference is the location and space of the hardware integration. The first two illustrations keep all the electronic components within a dry compartment by the upper reservoir, while the final design integrates a mode switch and sound system within the entire top wooden structure. There is no upper reservoir, so this allows for much more space and flexibility with the hardware in the project. The use of a cylindrical funnel as seen in the figures is put in place to manage the water by ensuring even flow and creating a laminar water stream to its highest extent. Consequently, the lasers located on the top of the funnel will be able to detect hand movement accurately.

Chapter 3: Research and Investigation

Aqua Strings is a project that will incorporate numerous connected systems that will leverage a variety of different technologies and physical properties. These systems, while independently serving their own function, need to work seamlessly in tandem. As such, an extensive level of research is required to properly solve the number of complexities regarding the project's design. If these systems were isolated or if the project were downsized, this might not require as much preparation. However, Aqua Strings reflects several different layers of systems coming together. What this means is combining sensing technology, electronics, and audio, all potentially submerged in water, requires a delicate level of planning. Questions such as "how to protect the electronics from water damage," "how to control the flow of water" and "what methods of sensing are most optimal" are essential to ask in this section. The following research will cover considerations carried out in regard to the selection of relevant technologies and components deemed appropriate for use in Aqua Strings.

3.1 Technology Comparison and Selection

3.1.1 Laser Harp Design

Fundamentally, Aqua Strings' core design is that of a laser harp with the additional element of flowing water. While this element adds significant complexity to the electrical design, the priority of design is how to construct the harp first and add water later. There are several ways to design a multiple stringed laser harp. The first we had considered was using a single high wattage laser that was refracted into multiple beams. The beam would be aimed at a small mirror sheet attached to a galvanometer. This galvanometer would be vibrating the sheet at high speeds of approximately 20,000 positions per second. As the beam shoots out the laser, it continuously hits the mirror at precise angles every few milliseconds. This refracts the single beam in multiple directions so quickly that it appears continuous. This design would allow us to allocate more of our budget towards a single, higher wattage laser instead of multiple cheaper lasers. However, a significant drawback is that all the beams shoot from the same point (where the mirror is held by galvanometer). This method would be impossible to implement with the water element as the beams would be angled but the water columns would need to be vertical. Thus, we considered another design where a laser would be positioned above each column of water. Using this design, the laser beam would be parallel with the flow of water and imposed inside of it. The downside of this implication would be an increased budget, as every musical note, or "string," would need its own laser, photocell, and any additional components to moderate the water column.

3.1.2 Waterproofing Options

Once we had a general idea of the electrical system without considering the water element, we then needed to brainstorm how to protect the electronics while still delivering on the Aqua portion of the project. We knew we needed to introduce

several water-sensitive peripherals inside of water reservoirs. For these inputs and sensors to function, they needed to properly isolated from their environment. However, if this protection inhibited their ability to function (i.e., interrupted the laser beam before it could reach its corresponding photocell) then that level of protection cannot be used.

3.1.2.1 Mineral Oil

At first, we considered that the laser and the photocell would be isolated in corresponding enclosures that would be submerged in the upper and lower reservoirs, respectively. This would mean that the enclosure would need to be transparent enough for the laser beam to pass through twice (once at the top after the laser and once at the bottom before the photocell). It would also need to be removable to access each component and perform maintenance if necessary.

However, this method assumes we use water as our liquid for interaction. Water is a natural default, but its properties conflict with the electronic system. We turned our attention to alternatives for the liquid element. One we had considered was mineral oil. Unlike water, mineral oil is non-conductive and will not cause short-circuits if these components are submerged. This would also allow these optical components to avoid being distorted or refracted by containers. Initially, this idea seemed strong. Some PCs even use food-grade mineral oil for use in cooling as an alternative to water-cooling. While mineral oil would be great in an application such as thermal cooling, its optical clarity presents a significant con. The oil possesses a different refractive index than water, altering the laser differently than water would. This will require recalibration if we switch liquids throughout the project. Additionally, mineral oil tends to grow cloudy over time due to contamination. Unless we sealed each reservoir, which would be difficult during the prototype and testing phases, this cloudiness dust, debris, or bacteria would occur. And if we sealed the reservoir, it would be difficult to clear any bubbles that may form in the oil. Trapped air does not dissolve easily in the material and may negatively impact the optics. This does not consider how the pumps required to move the liquid would interact with this alternative. The pumps we have considered so far are rated for water. Mineral oil is much more viscous than water and may interfere with the functionality of the pumps. On top of this, mineral oil is significantly more expensive than water. Procuring gallons of electronic-safe mineral oil would be magnitudes more costly than something as cheap and accessible as water. Finally, the user would need to interact with oil columns to strum the harp. This would mean that oil would coat the user’s hands and skin. The oily residue would leave users uncomfortable and make it harder to clean the instrument. Due to these many logistical downsides, water will be considered as liquid of choice.

This returns us to waterproofing the electronics. For something versatile, we considered using waterproof spray. Acrylic or silicone-based sprays work for most conformal coating applications. They are recommended to protect against harmful environmental conditions like moisture, static, and vibration, but aren’t ideal for full submersion, as the lasers and photocells will be. A rubberized spray would be better suited for long term submersion. However, water will also be moving

throughout the system, and continuous use may wear down the layers of sealant. While not a bad consideration, spray should be regarded as an additional means of protection and not the baseline.

If we are looking for full water-sealed protection, we then need to consider enclosing the electronics in waterproof housing. The two main options are:

1. enclosing electronics within watertight acrylic or other transparent boxes for short-term seal between maintenances.
2. potting individual components for permanent seal.

3.1.2.2 Waterproof Enclosure

Our first option would be to use a clear material-based box to enclose the electronics. Acrylic boxes would offer the most versatility while still providing a waterproof environment for the electronics to operate within. One of the biggest constraints is that the optical elements of the project would still need to operate despite being enclosed. Acrylic is a suitable candidate for this. Its transparent properties are ideal for an enclosure and the material could be cut to size. This is especially important as each enclosure might need to be custom made for each electronic. For economic purposes, only one side of the enclosure needs to be acrylic. The rest of the box could be 3D-printed and adhered to the acrylic panel. To run a cable through, we could cut a hole in the side of the box or design the 3D print with an opening. To seal any wiring or penetration points, a cable gland with a rubber compression seal can be used to create a watertight seal around any external cables. Finally, we can seal any additional seams (i.e., between the PLA and acrylic) with silicone or rubber sealant. Alternatively, for a more permanent bond, we could use epoxy to close these gaps. Realistically, the former will be used in the event that the enclosure must be opened. Thus, this first approach will be considered for components that might need maintenance, such as the laser.

3.1.2.3 Potting

Our second option would be potting the electronics. This might work particularly well for the photocell sensor, as it would not need much physical maintenance. One primary concern is how the compound will affect the optical transparency of the enclosure. There is a chance the sealant might alter the way the sensor can read the laser beam within its protection. The epoxy or potting compound must be clear and non-scattering to allow the laser to pass through. Bubbles or other imperfections may reduce the accuracy of the sensor. To help with this, the thickness of its encapsulation should be considered. Ideally, it must be thick enough to provide protection but still thin to best minimize light loss. Another main concern is that sealing the electronics by potting them would make maintenance impossible. We will need to evaluate and ensure that each element works perfectly before potting. This will be complicated when considering how water will affect the optical system. This approach may be used later in the design process once prototyping has been completed. Once we know for sure how elements will affect the system, then we can permanently seal the electronics using something like potting.

A potential third option could be to combine the two. Instead of fully potting the photocell, we could pot everything except the sensor area and place a clear acrylic panel over the sensor. This would maintain optical clarity while still ensuring the electronics stay waterproof. While this approach would not be as ideal for parts that might need maintenance or be difficult to replace, such as the laser, it could work better for components like the sensors. If a complication with the component arose, it would be easier to replace it instead of attempting to reopen the potted electronic. Replacing lasers wouldn't make much economic sense, but lower cost electronics like photocells would. This approach will be considered for such items.

3.1.2.4 Moisture Protection (Conformal Coating/Silica Gel Packets)

When we consider a means to enclose the electronics, we need to consider the possibility of trapped moisture. While this might not be an issue for potted electronics, those that will be protected using the first method may be subject to condensation. This may happen due to several reasons. For instance, if the electronics were sealed in an already humid environment, they would be sealed in air that's already moist. Compounding this, temperature changes within the system may support condensation. The microcontroller or laser may generate heat when running, and when they cool down, moisture can form. This issue may worsen if there are small enough leaks for air to move through. Shifts in temperature and pressure will draw in air and moisture along with it. To protect against unwanted moisture within the enclosure, we could use silica gel packets. Silica gel absorbs moisture from the air inside the enclosure, reducing the odds that condensation will form. These packets will be placed inside the enclosure before sealing it. To guarantee they will not interfere with the electronics, we plan their placement not to be in direct contact. These will also only be placed in electronic enclosures that can be reopened and maintained. Their function would drastically decrease if they became saturated with water, thus they won't be initially considered for use in permanent sealed electronics. If we can guarantee a permanent seal where airflow is not possible (i.e., in a prototype), then silica gel packets within potting compounds will be reconsidered.

Revisiting a previous idea, an additional layer of protection would be conformal coating. While we had considered this unsuitable as a baseline level of protection, a spray would work best once the electronic has been placed inside its enclosure. A thin, waterproof coating applied to circuit boards or other electronics would protect against condensation and moisture. This extra process would be done before sealing the electronics and again at specific maintenance instances. This will add even more redundancy against moisture in case of leaks or failure of the silica gel.

Table 3.1.6.4.1: *Waterproof Technology Comparison Table*

Feature	Mineral Oil	Waterproof Enclosure	Potting Electronics	Moisture Protection
Functionality	Submerge electronics in a non-conductive	Seal electronics inside of an	Fully encapsulate electronics in	Additional measure utilizing silica

	liquid to circumvent water damage	enclosure that allows them to function while submerged	resin for permanent waterproofing	gel or coating to prevent moisture inside options 2 and 3
Advantages	No need for airtight sealing	Reusable and maintenance accessible, potentially compatible with laser	Offers most extreme waterproofing and durability	Good for additional layer of protection and synergizes with enclosure and potting
Disadvantages	Messy, difficult to clean, not suitable for user interaction	Requires proper sealing and cable gaskets	Inaccessible to maintenance, may block laser	Not enough on its own, needs to paired

3.1.3 Applicable Sensor Options for Water String Detection

To have a fully functioning prototype and final project we need affordable, reliable, and precise sensors that work properly when used in conjunction with water. These sensors need to be able to travel roughly 1 to 2 feet between the transmitter to receiver with minimal refraction in the water string.

3.1.3.1 Capacitive Sensors

Capacitive sensors work by detecting changes in the dielectric properties of a given environment. This system consists of two plates that form a capacitor and when a conductive material, such as the water strings, enters the field the capacitance between the plates will change, causing the sensor to detect when a break in the water occurs. One advantage of this type of sensor is that it is highly sensitive and is capable of sensing both solid and liquid conductive materials making. Another advantage is how these sensors are able to detect changes in the water through a material making it a good choice when trying to waterproof the electrical components of the project.

However, even though there are some advantages the disadvantages make it not a good fit. For example, since a water component is involved and this project is taking place in a state where humidity is typically high, this could lead to false triggers of the capacitive sensor due to high humidity or temperatures. Another disadvantage is the sensors' limited range, especially when detecting liquids, making it difficult to accurately detect when a string of water is broken.

Capacitive sensors offer a variety of benefits including high sensitivity options to detect liquids along with their non-contact detection approach but due to environmental factors this type of sensor would not be suitable for our project’s needs.

3.1.3.2 Ultrasonic Sensors

Ultrasonic sensors work by detecting and measuring the distance from the sensor itself to an object, the object being the string of water. These sensors could be mounted in the lower part of the system to detect any changes of where the water lands at the bottom of each water string. These sensors would send high frequency sound waves and wait for these waves to bounce back (called an echo). When there is a displacement of where the water is landing there would be a change in the normal pattern of echoes, causing the sensor to detect when a water string is broken. One advantage of this type of sensor is its non-contact form of detection since it doesn't need to be submerged in the water.

The disadvantages of these sensors include their high sensitivity to environmental factors such as temperature and humidity, causing inaccuracies when detecting water breaks. These sensors also aren't quite precise as other sensor options meaning they wouldn't be able to detect small changes, such as if a water string is broken for a very short amount of time. A third disadvantage is how it would handle the water movement of the lower reservoir. Since we would have up to 8 water strings flowing into the same reservoir this will cause movement within the water as it falls, possibly triggering a false detection.

Even though ultrasonic sensors pose a creative and innovative solution, there are numerous issues that would arise if these sensors were implemented to detect the breaks in the water strings.

3.1.3.3 Piezoelectric Sensors

Piezoelectric sensors work by generating an electrical charge due to mechanical stress. In relation to applications such as this Aqua Strings project, the sensor would detect unusual disturbances such as vibrations that could be caused due to a break in a string. Without a break, the flow of each water string would be at a constant rate, so where there is a break there would be a change in the tension of the string causing the sensor to trigger. This would convert mechanical energy into electrical energy through the sensor. These sensors non-contact type of sensor would be an advantage since it would help with waterproofing the system. These types of sensors are also very compact and low power since they convert mechanical energy into electrical current.

Even with these advantages, the disadvantages make it a poor choice for our project's needs. One disadvantage is the output signal complexity which typically requires amplification and/or filtering of the voltage signal, which would overcomplicate the system as a whole. Another disadvantage is how sensitive it is to small vibrations which could cause false triggers such as if string 1 breaks but string 2's sensor detects the movement and throws a false trigger. A third disadvantage or challenge would be determining where to mount the sensor so that only the intent string would affect each sensor, which would be an intensive task.

Due to these three main issues that concern the reliability and accuracy of this type of sensor, it would not be a wise choice to implement them in our project.

3.1.3.4 Inferred (IR) Beam Sensors & IR Receiver

Inferred sensors work by using both a transmitter and receiver to detect any break between the two points. Depending on the type of transmitter used the inferred beam could either be a continuous beam or a pulsating beam but regardless the beam would need to be calibrated and aimed directly at the receiver opposite the transmitter. This beam would have to be pointed downward to go through the water string into the waterproof receiver since the water string would be falling on top of the component. One advantage is that multiple manufacturers have epoxy or silicone encased parts already so that solves the water proofing concern. Also, these sensors have a fast response time which is a main requirement for our project since it is supposed to trigger a note to play when a string is broken.

One disadvantage to using IR sensors would be having to embed the receiver component in the water but since waterproof encased components are already manufactured that solves that concern. Another disadvantage would have to be the difficulty with aligning the laser beam to hit the receiver. Since infrared beams are typically greater than 900nm in wavelength that makes the beam inviable to the naked human eye so it will be hard to make sure the beam is hitting the receiver properly.

Infrared beam sensors wouldn't be a bad choice since they have a vast number of advantages along with disadvantages that could be mitigated but the next choice is able to meet all the necessary requirements.

3.1.3.5 Laser Sensors & Photodiode (Selected Option)

Laser sensors work similarly to infrared beam sensors since they both require a receiver component to detect a change in the system. With these laser sensors high focused beams of visible light are used to detect an interruption, this means the wavelength of these beams must fall between 380nm to 700nm (the visible light spectrum). When applied to this project, these laser beams will be mounted above each water string for the light beam to be sent through the water where it would meet the photodiode (receiver) on the opposing end. An advantage to using laser beams over IR beams is how the human eye can see the direction of the beam which aids in aligning the transmitter and the receiver. Another advantage is its instantons response time, reducing latency and meeting the previously outlined project objectives and requirements. A third advantage is its extended maximum distance between transmitter and receiver of 6 feet compared to the 1.5 feet for IR sensors. These sensors are also manufactured to be epoxy or silicone encased essentially waterproofing the component.

One disadvantage to these sensors is their need to precise alignments but since their laser beams have wavelengths that fall between those on the visible light spectrum, the alignment process won't be as difficult. Another disadvantage is how the water element typically absorbs the light beam, reducing the distance the laser

beam can accurately travel. The absorption is least occurring when wavelengths fall between roughly 400nm to 530nm, which are the colors green and blue.

According to the research done on the different applicable sensors, the transmitter beam and photodiode combination has proven to be more efficient and effective. However, the LED laser sensor is the better option compared to the IR sensor due to its larger range, visible beam (for needs involving alignment) and its ability to be waterproof.

Table 3.1.3.5.1: *Sensor Comparison*

Specification	**Capacitive Sensor**	**Ultra-sonic Sensor**	**Piezo-electric Sensor**	**IR Beam Sensor**	**Laser Beam Sensor**
Detection Method	Changes in conductivity (capacitance)	Echo detection, pattern recognition	Mechanical stress conversion, pattern recognition	Transmitter-receiver beam interruption	
Possible False Triggers	High (humidity & temp interference)		High (vibration inter-ference)	Moderate (water absorption)	
Alignment Difficulty	Low (mounting two metal plates)	High (multiple sensors pointing at strings next to one another)	High (mounting issues, water strings are too close together)	High (beam is not visible to human eye)	Moderate (beam is visible)
Effective Range	Limited (effective within centimeters)	Mid-range (~1 to 2 feet)	Less than a few centimeters	~1.5 feet MAX	~6 feet MAX
Cost & Availability	$5 - $20	$10 - $20	$2 - $25	$3 - $25	$3 - $25

3.1.4 Laminar Flow of Water Strings Options

For the Aqua Strings project to work as designed, each usable water string needs to have a laminar flow. A laminar flow is defined as a flow of liquid that follows a linear and parallel pattern which ensures a smooth flow. The need for this type of water flow is because the laser beam requires minimum refraction within the string so that the beam can reach the photodiode (receiver) on the opposing end without alignment issues. To ensure laminar flow can be reached at a consistent flow rate there are multiple techniques or devices that could be used.

3.1.4.1 Perforated Material

Perforated plastic sheets are a fast and effective means of achieving laminar flow by diffusion of the incoming flow and relief of turbulence. Perforated sheets usually have an array of small, evenly spaced holes or slits that allow air or liquid flow at a predetermined rate. The fundamental principle with the application of perforated sheets is that they diffuse or filter, distributing the flow and allowing it to stabilize before it moves into a more sensitive system. The size and pattern of the perforations will have a bearing on laminarization efficiency.

The perforations reduce the velocity of flow at various points in the sheet. If fluid is supplied through the holes, the velocity distribution is made more uniform, promoting smoother downstream flow. The flow becomes less turbulent, a state necessary for laminar flow. In gas or air usage, the sheets are used to prevent high-velocity air from striking or mingling in such a way that it would result in random flow. Perforated sheets have the same role in liquid systems by reducing areas of high-speed that would produce turbulent flow.

Perforated plastic sheets possess various advantages that can be utilized across various applications. They are cost-effective and therefore do not put pressure on the budget for a system that demands a simple method of flow control. They are easy to use and implement since they can easily be cut or molded to suit the specific application needs of the system without requiring major modifications. Apart from this, they are also not fixed; the size and the shape of the holes can be customized to better suit the needs of the airflow or fluid system. However, there are limitations in perforated sheets. They cannot provide the perfect laminar flow in high-velocity systems since their diffusion and reduction of turbulence capability is limited to a certain extent. With time, they also become clogged, when there are particles in the fluid, and thus become useless. The success of perforated sheets largely depends on the precise alignment of the holes; error or improper spacing of the perforations could result in poor laminarization and therefore needs precise pattern design.

3.1.4.2 Honeycomb Flow Straighteners

Honeycomb flow straightening devices are a series of hexagonal open flow channels that are uniformly spaced for flow conditioning, very similar to perforated materials. This setup forces the fluid through small uniform channels along a set-controlled pathway that reduces the random motion caused by turbulence and promotes motion in smooth, parallel lines. The structure does resemble a

honeycomb, in which each cell is a channel for fluid passage. They find widespread applications such as scientific experiments, aerospace engineering, and HVAC systems, where flow has to be controlled and measured accurately.

The honeycomb design works by inducing multiple parallel, small-flow pathways so that the flow is biased toward uniform direction. Bigger-sized and lesser-numbered honeycomb cells can enhance performance; smaller cells will, however, cause high flow resistance; more channeling results in better interference. The fluid enters with random turbulence and gradually loses this through multiple cells, resulting in a significant reduction in turbulence with the increase in distance traveled within the honeycomb material due to flow streamlining. The cells in the honeycomb structure act as a series of simple flow conditioners, each slightly redirecting the flow until the entire fluid stream is aligned into parallel paths. This straightening effect assumes great relevance for applications wherein the fluid must enter sensitive systems or components with minimum disturbance since turbulence could cause measurement errors or operational inefficiency-even damage in precision environments.

Honeycomb flow straighteners are the most efficient means of generating laminar flow. The designs streamline the flow into paths almost parallel to each other, hence directing the fluid uniformly and minimizing turbulence. They are resistant and strong structures for use in high-pressure environments; therefore, honeycomb straighteners are widely used in various industrial applications. Moreover, they have the additional strength of being customized to specific flow conditions by cell sizes, material, and density, enhancing their versatility. These provide good flow uniformity, beneficial for delicate operations, for instance, scientific experimentation or precision fluid systems. Their downfalls entail larger sizes, and therefore space-consuming, unsuitable for compact systems. Their manufactured costs, concerning simpler alternatives, are relatively high. They are efficient, but an optimized design for a honeycomb structure suitable for a specific application requires meticulous engineering to promise the best performance.

3.1.4.3 Funnels and Nozzles

Some of the basic devices that are used to control and direct the flow of fluids in either states that are liquid or gaseous type include funnels and nozzles. A funnel is as such designed to direct fluid from a wide opening to a narrow outlet so that spillages or mess is minimum, such as in laboratories or industrial sites. Unlike a nozzle, a funnel is not designed to direct the shape and direction of fluid then leaving the system. Though both devices shape and direct fluid flow, a nozzle typically gives much more precise shaping of the three flow parameters - pressure, velocity, and the pattern of the fluid stream. Together, these two are basically required wherever efficient management of fluid dynamics is needed. Through the control over speed and direction of flow, they prevent problems, like uncontrolled separation of flow or turbulence, having the potential to arise without any restriction on fluid behavior.

Funnels and nozzles control the flow of fluids by controlling the cross-section of the area through which they pass. The velocity and pressure of flow are regulated

in the same manner. A funnel is a hollowed-out cylindrical item, which when placed in an upright position has a funnel-shaped mouth opening at the top and a smaller outlet at the bottom. The flow of the fluid accelerates as the fluid enters a funnel. The broad opening is then progressively narrowed down to a smaller opening. A narrow stream is produced, which can then be controlled to direct the flow of the liquid or disperse it, as required. It is designed in such a way that the liquid is practically undisturbed, meaning that it tapers gradually rather than having acute bends and sudden velocity changing conditions that could give rise to turbulence. Nozzles do similar things, although they are generally more than just a faster fluid flow in restriction; they are also designed to shape the manner of the discharge through different forms like a finely nebulized jet, a straight jet stream, or a diffused spray. Here fluid will be accelerated on the moving fluid in a converging nozzle, where it moves through the way of narrowed pipes and pressure will be acted upon resulting in kinetic energy. Therefore, this gradual acceleration maintains orderliness in the liquid flow throughout without inducing chaotic movement hence suitable for laminar flow. When nozzles and funnels are paired with diffusers later on to reduce the flow speed, they become effective individuals to minimize turbulence and instigate more even flow patterns with controlled fluid movement. Ideal as they are fast and dependent on each other, appropriate conditions must be ensured so that no hazard occurs when velocity alters because variations will cause flow separations and with that, additional turbulence will occur finally, nullifying benefits of laminar flows.

Funnels and nozzles are rather simplistic yet practical devices when it comes to controlling and directing fluid movement. The implementation is relatively straightforward and finds multiple applications in industrial fluid handling systems. The simplicity of construction and low cost make them a viable option for tasks that do not require sophisticated flow conditioning. Funnels and nozzles are equally applicable to treat the flow of either air or liquid, and the design can be suitably modified to achieve the desired flow rate. Even with these positives such systems come with disadvantages. An abrupt transition from a large to narrow configuration may induce flow separation resulting in turbulence, so a slow gradual shape change is needed. Similarly, if laminar flow has higher velocities, funnels and nozzles may not play an efficient role: they generally serve purposes inclusive of flow control, not turbulence elimination. The functionality of such an aid depends on the flow rate; at higher speeds, turbulence will develop, even with the use of such.

Table 3.1.4.3.1: *Laminar Flow Specification Comparison*

Specification	Perforated Material	Honeycomb Flow Straighteners	Funnels and Nozzles
Efficiency in Achieving Laminar Flow	Moderate (Reduces turbulence for low-velocity flow over high-velocity flow)	High (Structures are more uniform with parallel pathways)	Moderate (Best for directing flow but doesn't ensure laminar flow)
Design Complexity	Simple (can easily be constructed with a drill and spare plastic)	High (requires a uniform pattern, possible 3D print)	Simple (can obtain from store)
Cost	~$1 - $5	~$5 - $10	~$5 - $10

After researching different techniques, the three mentioned approaches stand out the most. To achieve as close to a laminar flow as an affordable price point a combination of all three mentioned methods should be used. See section 6.1 for figure(s) that demonstrate possible configurations. For this reason, the table shows all three methods being highlighted and selected.

3.1.5 Applicable Sensors for Water Level Detection

For the project to run smoothly with playable water strings, the water in both the upper and lower reservoir needs to be monitored and maintained. This can be done by having a water level sensor in the upper reservoir that tracks the level of the water in the tank and adjusts the state of the water pump whether it is on or off. This would be an essential part of the functionality of the project because without it the upper reservoir could run out of water, which would ruin the flow of water falling leaving the user without water to run their hands through. On the other hand, the lower reservoir could also run low on water if the speed of the water pump exceeds the rate at which the water falls from the upper to lower reservoir. To avoid any lower water issues, a water level sensing would need to be implemented to ensure proper functionality of the project.

3.1.5.1 Pressure Sensors

Pressure sensors work by being fully submerged in water and measure the hydrostatic pressure exerted by the water mass above the sensor in a tank. The

pressure throughout the tank is directly proportional to the height/ depth of the tank as it can be represented by the following pressure equation:

$$P = \rho g h$$

"*P*" represents the pressure as seen by the sensor, "*ρ*" represents the density of water which would be about 1000 kg/m^3, "*g*" represents the acceleration due to gravity (9.81 m/s^2), and lastly "*h*" represents the height of the tank/ reservoir in meters. By understanding this height could be calculated based on the other variables with an acceptable threshold for the height being set and if the water level falls outside of said threshold the water pump will act accordingly.

Some advantages with this type of sensor include but are not limited to, are its designed capability to be submerged in water meaning there are no other necessary waterproofing measures need for this sensor along with ability to accurately work regardless of environment changes such as humidity or presence of mist/ vapor. By utilizing a sensor that is specifically made for the environment it would be measuring in reduces the occurrence of inaccurate data being measured and/or calculated.

Even with a list of advantages, there still are some significant disadvantages to this type of sensor. The main downside to pressure sensors is the high cost for these sensors compared to other traditional methods or sensors. When the same outcome could be reached with a different component then the project will tend to veer in that direction. Another disadvantage is how these sensors typically are made for deep, large tanks and in this project, we wouldn't be utilizing reservoirs anywhere near the size at which these sensors are rated for. This would be an insufficient use of funds based on the project needs.

3.1.5.2 Optical Sensors (Infrared)

Optical sensors provide accurate and contactless measurement of water levels based on infrared or laser-based reflection principles (see section 3.1.3.4 for additional information). The mounting configuration is relative to the environment and application, with three primary methods: sidewall, top-mounted, and submersible installations. The sidewall mounting consists of mounting the sensor at a stationary location within or outside the tank and detecting water by the sensor at a predetermined level. This is the simplest form of level sensing and is commonly utilized for on/off type control applications, e.g., the pump start-up once the water reaches a critical level. Fitted externally, there may be a requirement for a transparent window for the sensor to be able to read effectively. This can be applied to high- or low-level water sensing but does not provide continuous level measurement.

To measure level continuously, a top-mounted optical sensor would be the typical selection. They are positioned at the top of the tank, sloping downwards, and measure readings for the water level based on infrared or laser beam reflection. It is appropriate for large or enclosed tanks where water contact is to be avoided. The alignment and detection path should, however, be clear to provide reliable

readings since surface disturbance or obstructions can bend the readings. Alternatively, submersible optical sensors are mounted inside the tank at different levels to facilitate multi-level measurement. These sensors are particularly handy for small tanks where there is a need for continuous level measurement at multiple points.

There are numerous advantages to optical sensors in the field of non-contact water level detection. In contrast to pressure or float sensors, they do not contact the water, which prevents mechanical wear and grime. They are often touted for their accuracy, whereby even slight movements in water levels can be recorded, which may make all the difference in applications that need accurate measurement. In contrast, these sensors are highly suited to sealed and pressurized tanks, where traditional levels of measurement may not work well. Optical sensors are also fast instruments and, hence, used in most real-time monitoring and automation control systems.

At the same time, some shortcomings are associated with optical sensors. For example, turbidity, foam, or strong ambient light can create conditions under which optical sensors may fail to operate effectively. In dirty or contaminated water environments, sensor surfaces may become dirty from easily adhered particulates causing the eventual misreading of the actual parameter. Proper alignment and installation are a must, especially for sensors mounted above since they can undergo reflection distortion when the shapes of tanks are not regular. Furthermore, optical sensors are much more expensive than float switches-probes, for instance, which is another drawback. Optical sensors will require maintenance periodically and calibration based on the operating environment to ensure that they perform as good as new.

3.1.5.3 Float Switch Sensors

The float switch sensors are among the most affordable and reliable options for water level detection, which is why they find the most common application in both residential and industrial functions. The basic operation involves simple mechanical design: the buoyant float travels linked to the height level in the tank and, when it reaches a certain height above the tank floor, activates a switch. Such low-cost, simple operation actuators are best used in applications where basic on/off control is required, protection against overflow, or starting the pump when the water level becomes low. The most common installation is side wall mounting, where the float is set at the desired height inside the tank. If the float sensor is in a vertical position, the set switch will activate and start other devices, like a pump or an alarm. This kind of installation requires minimal calibration work, and provides consistent performance a long way without heavy, complex electronics.

Another mounting option could involve a top-mounted float switch hanging from the very top of the tank, permitting them to move freely in accordance with the liquid level. Being simple mechanically, float switches are very reliable since they do not depend on electronic components that are prone to damage by the environment. Submersible float switches provide an alternative for applications in which external mounting may prove difficult and allow multi-level detection of the

different sensor depths. Unlike the more sophisticated types of sensors, maintenance or specialized calibration are not required by float switches, thus providing a long-term standby service in most cases. Further, since the float switch operates even with very turbid water with a lot of foam and potential contaminants in it, the application is quite practical.

One of the biggest advantages of float switch sensors is that they are inexpensive and mechanically rugged. That their operation is simple, based on buoyancy and a mechanical switch, makes them work reliably with little maintenance. Float switches perform just as well as electronic sensors, which have to be carefully calibrated, for example. They are very versatile and perform well in various water conditions, including turbid, foamy, or dirty water. Their mounting simplicity makes them an ideal choice for small-scale applications other than industrial applications, conserving mounting time and money. Also, because they are devoid of high-technology electronics, they are power-efficient and ideal for low-power or off-grid applications.

While reliable, float switches are not perfect due to their mechanical nature. Over time, mechanical abrasion on moving components, where sediment or mineral buildup might be introducing operation interference, can happen. In addition, in very turbid water, excessive float movement may sometimes introduce spurious triggering or erratic operation. While being extremely capable at simple high/low water level detection, they are not suited for continuous level measurement, which could be a drawback where high accuracy measurements are required in an application. Simple cleaning and inspection can help to maintain function, but under harsh conditions more frequent maintenance will be necessary. Float switch sensors remain an inexpensive and reliable choice for water level control despite these challenges, making them the perfect option for the Aqua Strings project.

3.1.5.4 Resistive Sensors

Resistive sensors are electronic sensors that measure water level through conductive traces that operate based on changes in electrical resistance. These sensors typically have about two to five conductive lines that are separated by some sort of resistive material such as conductive ink or metal traces. When these lines come into contact with liquids such as water, the liquid contact will then adjust the resistance of the material causing the voltage and/ or current to also adjust accordingly throughout the circuit. Either a microcontroller or some sort of measurement circuit would be able to understand these changes read by the sensor to determine whether there was presence of water, at what height the top of the water has reached and/ or the amount of force/ pressure the device experiences. The sensors work by relying on measuring the conductivity of the water to complete the circuit of the traces on the sensor itself. As the sensor becomes for submerged the conductivity will change proportionally so that it can read a numerical value and relatively dictate the level of the water.

Two of the major advantages of these types of sensors are their simplicity and their affordability. These sensors typically can be purchased ready to use or may need some slight additional parts that also don’t cost much. Their components are

manufactured at a very low cost due to their mass production for the consumer market along with their industrial applications. In addition, they also are highly sensitive, allowing for quite accurate results and interpretation of data that can be used to modify other portions of our project, such as turning the water pump on and off as needed. Another benefit would be how intuitively the sensor can integrate with microcontrollers such as Arduino, which would be perfect for our needs since it only requires an analog to digital converter (ADC) to read the changes in resistance. There is no need for over complexity since the sensor fundamentally works based on the basic electrical properties on conductivity and can be easily implemented for applications involving touch screens, water level detection or force measurement of a system.

Even though this option has vast advantages in its application to our project, it is important to note the disadvantages as well in the hope of mitigating them. One of the major downsides is its susceptibility to degrading over time, especially when it is left within a liquid for an extended period of time, causing its functionality to eventually stop working and require the sensor to be replaced. To mitigate this, we would just have to order a batch of these sensors and since they are already so cheap that would not be a major issue. Another disadvantage would be how temperature of the surrounding environment affects the values the sensor reads. Our team would just need to make sure these water level sensors are placed near the laser beam transmitters since those will increase in temperature so there should be no other components that would alter this read data. This would be an excellent option for our Aqua Strings project due to its low cost and simple detection design.

Table 3.1.5.4.1: *Water Level Sensor Comparison Table*

Specification	**Pressure Sensor**	**Infrared (IR) Sensor**	**Float Switch Sensor**	**Resistive Sensor**
Functionality	Measures hydrostatic pressure (water depth and pressure have a direct relationship)	Measures water level based on beam reflection	Float moves with top of water, activates switch	Measures resistance based on changes in water conductivity
Advantages	Accurate in humid environment, no alignment	Contactless, no mechanical wear	Simple, cheap, reliable	Simple, cheap, easily integrated

	or calibration needed			with microcontrollers
Disadvantages	Costly, best for large tanks, high maintenance cost	Interference from turbidity, lighting	Mechanical wear	Possible corrosion of electrical components
Cost	~$30+	~$20 - $40	$3 - $15	$3 - $15

3.1.6 Applicable Laser Beam Technology for Harp Strings

For the magic of Aqua Strings to work, we need a means for the instrument to properly “strum” whenever the user interacts with a column of water. This relies on the interaction between the sensor, the water column, and the laser beam. All three of these factors need to work together to properly pull off this effect. As such, it’s important that we also consider what laser technology we will use that will pass through the falling columns of water. This beam needs to be safe (since players will interact with the beams), reliable in water environments (such that they won’t scatter too much or be absorbed), and easily detectable by the sensor at the bottom of the water column. An additional factor that we can consider for the laser beam is whether or not we want to make the laser beam visible. Having it visible may help the instrument be more intuitive. If the user can see the beam they are strumming through the water column, it may be easier for the user to play the harp. Additionally, we can coordinate the visible beams of light to be different colors. This would add a layer of interest that would synergize with the water columns. Each note could potentially be represented by a different color. Or, if the project is set to “Fountain Mode,” the harp will function like a water installation and be lit up by the different color lasers. A potential downside of having different color lasers is that their interactions with the sensors may differ. In a project where our main priority is consistency and latency, having an optional feature that would inhibit reaction time would be a poor decision. There’s a chance this wouldn’t be the case, thus it’s worth prototyping the error to see if this additional feature is worth implementing. The alternative is to choose a laser beam that is invisible. This would hide the laser beam’s reflection when the user passes their hand through the column of water. In this case, it appears as if the water column has become a functioning harp string. If the beam were invisible, the only interaction the user could see would be with the water. Both cases are tempting and will be considered at a further stage in development. At minimum, the laser beam’s safety, reliability, and detectability will be considered at the research stage.

3.1.6.1 Diode Lasers (Point Laser)

A diode laser consists of a p-n junction semiconductor that emits coherent light when electrons and holes are stimulated by an electrical current. This light is then amplified by an optical cavity where the light would shoot out from. The wavelength

(and corresponding color) of the emitted light will change depending on the semiconductor material used. One key feature of diode lasers is that they are compact. This is important due to the potential waterproofing of these electronics. If the laser beams are stored in an enclave below the upper reservoir, then they would need to be compact to fit with the design. If they are implemented within the reservoir, it would be ideal to be compact to minimize the size of the electronic needed for waterproofing. Another key feature is that they are efficient. They consume a low amount of power during operation. Additionally, there exists a range of wavelengths for diode lasers. Red (650-670nm) is the most common and easy to source. Green (520-532nm) and Blue (405-450nm) are brighter but may be more expensive. There are also infrared (780-1550nm) wavelengths that are invisible to the naked eye. We will touch on this application in a later section. One advantage of this technology for our project is that they are safe for eye exposure under normal conditions. For instance, a class 2 diode laser, like a laser pointer, is safe for eye exposure assuming it was not directly aimed at the eye. Since the output of a diode laser is a single point at the aimed surface, there shouldn't be any safety concern as long as it is not aimed at a person. For the user of the harp, this certainly will not be a problem. The laser beam will be shining directly down and be enclosed within a water column. A potential safety consideration is for our team during testing and assembly of the project. We will have to ensure that the laser beam is never directly aimed at another person or ourselves while testing these optical systems. Another advantage is that they produce continuous and coherent beams. This will ensure reliable detection at the base of the water column. Additionally, these lasers are affordable and often readily available. Their power levels are offered at a wide range (roughly 10mW-5W), making them accessible for this project depending on the intensity of the beam. This will be important when considering how lower power lasers may scatter in water and require a more intense beam. The biggest downside of a diode laser is that the beam itself is invisible. The initial output of light at the base of the laser might be seen, but the beam as it shoots through the water column won't be. This isn't a major fault, as the visibility of the beam is an optional component. However, it is worth noting that a diode laser will only produce visible light when it hits its corresponding sensor at the bottom of the harp. Ultimately, a diode laser is an excellent choice for a laser harp due to their visibility to the sensor and their affordability.

3.1.6.1 Line Lasers

A line laser is a specialize laser module that projects a straight, thin laser beam. It utilizes the same technology as a diode laser to produce light: a semiconductor. The difference lies in how the light is altered before it is emitted. A basic diode laser produces a single dot onto whatever surface the laser is aiming at. This makes it ideal for applications like a laser pointer where the user needs to highlight a specific physical location. The size of this dot is incredibly small but will expand over distance due to beam divergence. A line laser starts with the same source as a diode but uses an optical element, like a lens or diffraction grating, to spread the beam into a linear shape. This converts the single, focused dot produced from a diode laser into a visible and thin, elongated line. Ultimately, a line laser is essentially a diode laser that has been optically manipulated to have a different

beam shape, spreading into a line instead of a single point. Thus, it's able to produce different wavelengths just like a diode laser. The most common colors are red, green, and blue. This matters more as the entire line of the laser is visible. If we wanted to add some aesthetic appeal using colors, we would likely pick line lasers as their colors would be more visible. That's one of the technologies' greatest advantages for the context of this project. It would provide a greater level of interaction between the laser beam and the water column. This may provide a greater detection area, making the optical system more reliable. In a similar vein, the broader beam will be easier for users to see. This would lean more into the optional requirement of making the beams visible and more like a harp string. This feature of the visible light scattering through the water could create a more immersive visual experience. A potential concern of this technology is that the beam may unintentionally spread when passing through the water column. Since the technology is already manipulating the beam away from a point and into a line, there's a greater chance that the beam might not stay uniform. This might be caused by the water's refraction (assuming it is not exhibiting laminar flow). Additionally, this technology is generally more expensive than its diode laser counterpart. Aside from this, the line laser exhibits similar qualities to that of the diode laser. Regarding its safety, the same considerations would be applied to ensure no one is hurt. Line lasers could be a great alternative to diode lasers for this project, especially if we're aiming for visibility of the harp's strings and a greater aesthetic.

3.1.6.3 Infrared Lasers

Infrared lasers emit light that is in the infrared spectrum and invisible to the human eye. Despite this, specific sensors can read and detect infrared light. They operate at wavelengths greater than 700nm. Common wavelengths that are available for infrared lasers are at lengths of 808nm, 980nm, and 1550nm. Similar to line laser technology, infrared lasers are produced from the same semiconductor base that diode lasers operate on. The primary difference is in the frequency of the lasers produced. While diode lasers produce a visible colored dot at the end of its beam, infrared will produce the same effect, but the dot will not be visible. The entire effect from an infrared laser would be outside the range of human vision. This offers a key advantage in making Aqua Strings a more immersive project. By having completely concealed laser beams, the effect of strumming water columns would be in full force. The water interaction would become the focal point of the instrument as the electronic portion becomes literally invisible. This could elevate the experience of the project and make the experience of strumming the harp more magical. Unfortunately, concealing the laser beams is a double-edged sword. If implemented properly, it can add to the experience. However, if something went wrong during testing, it could be incredibly difficult to troubleshoot a misalignment because the beam is invisible. We would have no immediate means of determining where something is going wrong (i.e., if the laser is malfunctioning, if the laser is misaligned, if the infrared sensor is malfunctioning, if the laser is not being properly detected by the infrared sensor, etc.). Additionally, infrared lasers might not coincide with the water element as smoothly as its alternatives. Shorter infrared wavelengths, like 808nm) penetrate water better, but as these wavelengths get

longer, they tend to get absorbed. Wavelengths at longer values like 1550nm are often heavily absorbed, which would disrupt the detection from the sensors. This could be avoided by working on a shorter wavelength of infrared. If we were to leverage this technology, this is an important consideration that we would keep in mind. Infrared lasers could provide a stealthy means of achieving the optical element of the harp. The water interaction and sensor selection need to be carefully considered for this technology. Specifically, an invisible beam with a wavelength around 808nm paired with an infrared photodiode detector could be a viable option. Utilizing this combination would allow for the water interaction to take full focus and obscure the electronics.

3.1.6.4 Vertical-Cavity Surface-Emitting Lasers (VCSELs)

Vertical-Cavity Surface-Emitting Lasers (VCSELs) are a specialized type of semiconductor diode laser that produces a different beam. They are similar in this vein to the line laser: both generate a beam using a semiconductor but manipulate the beam optically before emitting. VCSELs use a resonant optical cavity formed by layering two mirrors on either side of the semiconductor region. This difference results in emitted light is projected perpendicular to the surface of the semiconductor wafer. This means that the laser beam is emitted vertically as compared to traditional edge-emitting lasers. When using this technology, a highly collimated (parallel), low-divergence beam is produced. This is one key advantage of the VCSEL. The emitted light is a nearly circular beam with minimal spread over distance. This would result in improved detection accuracy, an aspect especially important when transmitting through water. Additionally, they are both energy efficient, in that they consume less power than traditional laser diodes, and both durable and reliable. Like their diode counterpart, they also can produce beams at varying wavelengths. These options span the visible light spectrum and even include infrared. To summarize, VCSELs boast several advantages like improved beam consistency, more accurate laser-sensor interaction, and lower power consumption. The largest drawback of this technology is that their availability is typically lower than a standard diode laser. They are common in industrial applications, but fewer are available on the consumer or hobbyist level. As such, they are often more expensive for an individual laser than all of the previous technologies. Unfortunately, this makes VCSEL a less likely pick as a laser technology, despite all its advantages. Its inaccessible for the scale of the project is too large a drawback to overcome this. However, they are a worthwhile consideration if this project were to be scaled up in the future.

Table 3.1.6.4.1: *Laser Technology Comparison Table*

Feature	Diode Laser	Line Laser	Infrared Laser	VCSEL
Cost	$5 - $20	$10 - $50	$10 - $20	~$20
Wavelength Range	400 – 980 nm (generally)	400 – 980 nm (generally)	780 – 1550 nm	400 – 940 nm
Power Output	1 – 5mW	1 – 5mW	1 – 100mW	0.1 – 10mW
Beam Shape	Circular Dot (on surface)	Thin line	Circular Dot	Circular Dot
Functionality	Emits a focused beam on the aimed surface	Projects a thin, visible line to emulate a string	Invisible laser detection for more subtle applications	Similar to Diode, utilizes a more efficient technology
Advantages	Cheapest, easy to integrate, has many colors	Visible laser allows for aesthetic applications	No visible beam allows for seamless applications	High-speed and very efficient, lowest power output
Disadvantages	More susceptible to divergence	Reflections from water can refract the beam itself	Needs appropriate IR-sensors	Higher cost than diode, less accessible

3.2 Part Comparison and Selection

3.2.1 Microcontroller Selection

The microcontroller will be at the heart of this project; thus, it's important that the most optimal one is chosen. Microcontrollers contain many essential components housed all in one chip, each with their own specs. One must consider the central processing unit (CPU), memory, RAM, I/O pins, serial communication interfaces, and more. For our case, we also need to consider the Digital-to-Analog Converter (DAC) and Pulse Width Modulation (PWM) capabilities of the chips due to the audio generation aspect. Microcontrollers are designed to execute specifics tasks using all these aspects; and as they are used, community support for these specific tasks grows. Another primary consideration is how available the libraries and community support are for each microcontroller. Additionally, the programming language to operate the system is important. Ideally, support for a higher-level language such as C or C++ would be preferable. While there is an abundance of microcontroller families to choose from, we decided to limit our selection to a few. For the scope of this project, we are considering options amongst ATmega328, ATmega2560, MSP430FR, ESP32, and Teensy4.0.

3.2.1.1 ATmega (Arduino)

The ATmega family of microcontrollers is a series of low power, CMOS 8-bit processors developed by Atmel, which was later acquired by Microchip Technologies in 2016. They run on AVR enhanced RISC (Reduced Instruction Set Computer) architecture. These chips are generally used most often in embedded systems due to pros like simplicity, low power consumption, and extensive library support. Specifically, we will consider both the ATmega328 and ATmega2560 microcontrollers. The ATmega328 is a simpler microcontroller with a fraction of the latter's flash memory, SRAM, GPIO pins, and PWM channels. However, it also operates on a much lower power consumption rate, even beating the most efficient ultra-low power consumption settings of its competitor. ATmega2560 boasts a larger capacity of memory and pins, allowing it to handle a greater level of scalability with the project (i.e., adding more strings to the harp). For our application, we will consider both microcontrollers' presence on Arduino boards. ATmega328 is featured on select models of the Arduino Uno and Arduino Nano. ATmega2560 is featured exclusively Arduino Mega 2560. Both of these boards offer the potential to meet the demands and constraints of our microcontroller.

3.2.1.2 MSP430

The MSP430 family of microcontrollers is a series of ultra-low power, 16-bit RISC processors developed by Texas Instruments (TI). These chips are generally used for similar applications to the ATmega, such as low-power embedded system applications, but especially ideal for battery-operated devices due to low power consumption. And unlike the 8-bit microcontrollers from the ATmega series, the MSP430's 16-bit architecture allows for a greater degree of processing whilst also reducing power consumption. One defining feature of the MSP430 is the ultra-low power consumption with operating currents ranging at 100uA in active mode. When in standby mode, currents can go as low as 0.4uA typically. This is a significant advantage if our power source is from a battery. When the system will have to power multiple lasers, sensors, and a water pump, having a low power cost to the microcontroller is a plus. Additionally, the MSP430 features a built-in DAC which can generate simple waveforms or synthesize audio directly. This produces a sound that's much higher quality than sound generated from PWM, as the ATmega microprocessor will have to do. For this project, we will consider the MSP430FR6869 specifically due to experience with the hardware.

3.2.1.3 ESP32

The ESP32 microcontroller is a powerful and versatile option for embedded applications, especially those involving Wi-Fi and Bluetooth connectivity. Developed by Espressif systems, it is the successor to the ESP8266, improving upon its predecessor with dual-core processing, added RAM, and integrated Bluetooth capabilities. These chips are generally used in Internet of Things (IoT) applications, but its versatility applies to a wide range of robotics and automation. The ESP32's wireless connectivity offers an interesting opportunity for this project. Using its built-in Wi-Fi and Bluetooth, the ESP32 could be used to transmit MIDI signals wirelessly. This would allow for a cable-free system when interfacing with

potential MIDI modules, a wireless speaker, or a computer. Considering our need to isolate any electronics from the risk of water, having a wireless connection would be beneficial. However, there is a risk of latency when sending audio signals over wireless connections. If selected, this feature is something that we will consider. Another key feature is the ESP32's much higher clock speed. With its 240 MHz processor, multiple laser interruptions could be processed much faster and sound outputs could be triggered with minimal latency. This offers a significant advantage over both ATmega models and the MSP430, especially considering our demonstrable specifications. We are aiming for a detection and a response time that is within tens of milliseconds. The microcontroller that we have chosen for this project may have a significant influence on this factor. If the difference between the ESP32 and its competitors were smaller, maybe this would not be worth highlighting. However, it is operating at a clock speed that is at least twelve times faster than its fastest competitor so far. Because of this, the ESP32 has a noticeable edge over the other options, alongside its higher flash memory and SRAM. One downside to note is that it has a much smaller pool of general purpose I/O pins to operate with. This shouldn't be a problem with the scale of the project and the number of strings we want the harp to have. However, it is worth noting if we decide to scale up the size of the project. Considering all of ESP32's features, it will be regarded as a solid pick for the microcontroller.

3.2.1.4 Teensy 4.0

Teensy4.0 is a high-performance microcontroller board designed by PJRC. Using a NXP i.MX RT1062 processor, it is one of the fastest microcontrollers available for hobbyist and embedded projects. It boasts powerful processing specifications, a wide range of I/O applications, and native USB support. If the advantage that the ESP32 has over the ATmega models and MSP430 is impressive, then the Teensy4.0 is leaps and bounds over the competition. It operates at a clock speed almost forty times faster than ATmega and MSP430 and almost three times faster than the ESP32. Another great feature is the native USB support that allows for a seamless connection to a computer for audio tweaking. Unlike the ESP32, it has native USB MIDI support, allowing it to be recognized as a MIDI device by a computer without extra components (i.e., external modules) or software workarounds. Perhaps its biggest proponent is the Teensy4.0's community and software support. It has a custom audio library that would be particularly useful for building a sound font for the harp. Additionally, the extension, Teensyduino, for Arduino IDE software provides support for MIDI, audio synthesis, and sensor interfacing. This would greatly reduce development time in this sector of the project. Overall, Teensy4.0 rivals ESP32's processing power while also providing an interesting advantage in the MIDI application. This makes it a solid option amongst the other microcontrollers.

3.2.1.5 Microcontroller Sound Generation Methods

Before selecting a microcontroller, it is important to consider how that system will synthesize the sound. Based on each of their own limitations, they will generate sound differently. Some of these microcontroller-specific differences lie in their processing power, complexity of digital-to-analog converters, and ease of audio implementation. There are several common approaches to producing audio, each with different hardware requirements and varying levels of complexity. The three primary methods are Digital-to-Analog Converters (DACs), Pulse-Width Modulation (PWM) audio, and MIDI (Musical Instrument Digital Interface) output. Each approach holds its own strengths and weaknesses, impacting what specific microcontroller is most optimal for the project.

3.2.1.5.1 Digital-to-Analog Converter (DAC)

Digital-to-Analog Converter (DACs) is the most direct method for sound generation. This method allows the microcontroller to output smooth, continuous waveform by converting digital values into corresponding voltage levels. The result is high-quality audio without requiring any additional external components for processing analog signals. Microcontrollers like the ESP32 and Teensy4.0 include native DACs, meaning they can generate sound signal directly. These signals can then be immediately fed into an amplifier or a speaker. Having a built-in DAC is a huge benefit because it allows these systems to use a method of sound generation that is almost universally applied. Internal DACs have reached a point technologically where they are the best they have ever been. Only small differences lie between DAC to DAC, but generally every modern DAC is entirely transparent and doesn't introduce noise when converting signals. DACs are particularly useful for playing pre-recorded audio samples, synthesizing waveforms (such as sine or square waves), and implementing real-time sound effects. These are all incredibly applicable for producing sound from an instrument. This gives us the freedom to choose between pre-recording notes or synthesizing the waveforms and playing in real-time. One downside, however, is DAC-based audio is often demanding on processing power and memory. Especially if it involves generating complex waveforms or handling multiple audio channels (such as having multiple different strings on a harp). For our application, DACs provide high-quality sound output with minimal external circuitry, making the ESP32 and Teensy4.0 stronger options as a microcontroller.

3.2.1.5.2 Pulse-Width Modulation (PWM)

Pulse-Width Modulation (PWM) is a widely used technique to generate sound when a native DAC isn't available. This method approximates an analog waveform by pulsing an output signal. Specifically, the microcontroller rapidly switches an output pin between a high and low voltage state at a specific frequency. By switching between the two states, the microcontroller creates a duty cycle, or a ratio when the output is "on" compared to when it is "off." And to create an analog waveform, the microcontroller rapidly alters this duty cycle. The result is a

waveform that, once passed through a low-pass filter, can produce sound. Microcontrollers like the ATmega328, ATmega2560, and some models of the MSP430 (MSP430FR6869 includes a native DAC however) rely on PWM for sound generation. This allows them to produce basic tones and simple waveforms. However, this means that the audio typically is of lower sound quality compared to DAC output. This is a result of unwanted high-frequency noise and requires additional filtering and external circuitry to resolve the issue. Despite these downsides, PWM is still an effective method where audio fidelity isn't critical. For Aqua Strings purpose, producing the highest quality sound is not a priority as long as the intended notes are played at the right frequencies. Thus, it is a potential consideration for those microcontrollers that rely on PWM to generate sound, like the ATmega328 and ATmega2560.

3.2.1.5.3 Musical Instrument Digital Interface (MIDI)

Musical Instrument Digital Interface (MIDI) is a method that, rather than directly generating sound, the microcontroller sends digital control signal to an external synthesizer or similar software-based sound engine. Using this digital signal, the MIDI will generate any necessary data regarding the noise, such as the pitch, velocity, and instrument data. It is important to note that MIDI software does not produce the audio directly. MIDI will then transmit this data to an external system, such as an amplifier or a speaker, to generate that sound accordingly. A microcontroller like Teensy4.0 is particularly suited for MIDI for this method. Since utilizing MIDI requires connecting to an external synthesizer, Teensy4.0's native USB-MIDI support provides it a significant edge. This allows it to function as a plug-and-play MIDI controller for external audio synthesis. The ESP32 also benefits from the MIDI approach. Using its Bluetooth capabilities, the microcontroller could connect to a computer and synthesize audio from a software-based sound engine. MIDI-based sound generation has several advantages. Since the sound generation occurs externally to the microcontroller, it allows for high-quality instrument playback without any additional processing power. It essentially offloads the audio synthesis and relieves the microcontroller of generating the soundwaves itself. Additionally, synthesizers boast greater versatility and can produce a wider range of instrument-specific sounds. If accurate, instrument-like audio is what we are aiming for, then MIDI-based audio generation is the method we should aim for. Consequently, ESP32 and Teensy4.0 are both microcontrollers we should consider.

3.2.1.5.4 Comparison of Microcontroller Sound Generation

Table 3.2.1.5.4.1: *Comparing Microcontroller Audio Systems*

Feature	ATmega (Both models)	MSP430	ESP32	Teensy 4.0
DAC (native)	No	Some models (FR6869)	Yes	Yes
PWM sound	Yes	Yes	Yes	Yes
External DAC needed?	Yes	Maybe	No	No
MIDI support	No (requires module)	No (requires module)	Yes (USB or Bluetooth)	Yes (native USB-MIDI)

The table above shows a comparison in the microcontroller's ability to utilize each of the sound generation methods. Their differences are reliant on the level of processing each microcontroller has access to. Additionally, the comparison is affected by their ability to communicate with external hardware. This is situational in cases where an external DAC is required or when an external synthesizer is needed for the MIDI method. It's important to note that all options have the ability to generate sound via Pulse Width Modulation. This method is the most primitive but is a potential backup in case the audio generation needs to be simplified. Both the ATmega328 microcontroller and the ATmega2560 microcontroller are the weakest in this department. They both lack a Native DAC and intuitive MIDI support. To leverage a synthesizer, they would need an external module to communicate with the microcontroller. The MSP430 is a little better than the ATmega. Specifically, the MSP430FR6869 features a native DAC while some of the other MSP430 models don't. Unfortunately, it would also need an external module to synthesize audio via MIDI. However, this might not be too big a downside considering its incredibly low processing power. It would have no need to outsource audio generation when it already takes such little power to operate. From its base specifications, the ESP32 is already in a stronger situation. It can leverage native DAC and MIDI support via USB or Bluetooth. This wireless potential is incredibly strong for the purpose of minimizing cables in a wet environment. The Teensy4.0 is also in a strong position. Like the ESP32, it can use native DAC and MIDI. Additionally, its USB capabilities can directly communicate with a synthesizer to the port on its board. This would minimize development time by having a direct connection to an external synthesizer. Ultimately, under the context of sound generation, the ESP32 and Teensy4.0 feature the best capabilities.

3.2.1.6 Comparison of Microcontrollers

Table 3.2.1.6.1: *Comparing Microcontrollers*

Feature	ATmega 328 (Uno)	ATmega 2560 (Mega)	MSP430 FR6869	ESP32	Teensy 4.0
Cost	20.00 (Mouser)	48.40 (Mouser)	24.00 (DigiKey)	8.00 (Mouser)	23.80 (Adafruit)
Size (board)	68.6 x 53.4 mm	101.52 x 53.3 mm	89 x 58 mm	51 x 25.5 mm	35.56 x 17.78 mm
Manufacturer	Microchip	Microchip	Texas Instruments	Espressif Systems	PJRC
ADC Resolution	10-bit 8 inputs	10-bit 16 inputs	12-bit 16 inputs	12-bit 18 inputs	2 x 12-bit 10 inputs
PWM Channels	6	15	8	16	31
Number of GPIO Pins	23	86	83	34	40
Communi-cation Interfaces	1 x UART 1 x I2C 1 x SPI	4 x UART 1 x I2C 1 x SPI	2 x UART 2 x I2C 2 x SPI	3 x UART 2 x I2C 4 x SPI	3 x UART 3 x I2C 3 x SPI
Power Consumption, Active Mode (per 1 MHz)	200uA	500uA	100uA	~200mA	100mA
Best Feature	Simple, low power	Many I/O and PWM	Ultra-low power, DAC	High memory, DAC	Fastest, USB MIDI

3.2.1.7 Selection of Microcontroller

To best suit the project's requirements, the microcontroller we have chosen is the ESP32-WROOM Development Board. The ESP32 seemed to be the strongest contender amongst the microcontroller options. It is fast, has a good amount of I/O pins, features a native DAC and MIDI potential, and is incredibly affordable. This specific model was partly chosen due to experience with this board and similar

models. Members of the team have used this microcontroller in previous projects and its familiarity will aid in the development of Aqua Strings. And this isn't even considering its extensive capabilities. This board features dual Tensilica LX6 cores clocked at up to 240MHz, creating a powerful processor for its size. Additionally, it features 512 kilobytes of SRAM. This board has access to 30 pins, 15 on each side. For specific information on each of these pins, consult Appendix C for figures on the pin layout diagrams for this microcontroller. Its many pins will be utilized to interface with the photodiodes and water level sensors. It supports essential peripherals like ADCs and DACs alongside potential communication protocols like UART and I2C. This will be important when it comes to the audio system. The DAC signal output will be vital for driving the different tones of the music. Ultimately, it balanced the constraints of the project the best while providing the most features.

3.2.2 Programming Language Selection

Table 3.2.2.1: *Comparison of Programming Languages*

Language	Dev Difficulty	Power Eff.	Exec Speed	Libraries	Docs Available	Learn Curve
ESP-IDF	High	Great	Great	Good	Great	Difficult
Arduino Framework	Medium	Good	Good	Great	Great	Moderate
MicroPython	Low	Poor	Fair	Good	Good	Easy
JavaScript	Low	Poor	Poor	Fair	Fair	Medium
Rust	High	Great	Great	Fair	Good	Moderate

In developing Aqua Strings, our interactive waterfall display project, the selection of a programming language is just as critical a decision as the selection of hardware. With the ESP32-D0WDQ6 microcontroller already selected due to its high processing capability, on-board wireless capability, and extensive feature set, the next task is to select the language that will unlock its full potential. This decision influences not just the ease of development but also determines the responsiveness of the system, stability, and ability to handle the complex interaction between sensor input, real-time sound synthesis, and dynamic UI.

Considering the seriousness of our project, where every pause of the flow of a water stream must produce a musical note with zero delay, the programming language used must allow for quick processing, precise timing, and intelligent resource utilization. Our evaluation examines some top contenders: the sprawling Espressif IoT Development Framework (ESP-IDF), the Arduino platform, MicroPython for rapid prototyping, JavaScript with Espruino for event-driven style, and Rust with high performance and good memory safety guarantees. There are

trade-offs for each language's benefits. For example, ESP-IDF has extensive hardware control and performance optimization but with the requirement of intimate knowledge of C, whereas Arduino is easy to use and has a large user base at the potential cost of utilizing the chip's advanced features to their highest capability.

Also, MicroPython excels in usability and rapid iterations but may not be able to cope with the real-time demands of dealing with ongoing sensor inputs. JavaScript through Espruino provides an event-driven, natural syntax, but the performance may not be adequate to meet the strict demands of our system. Rust, however, provides efficiency and safety, although its relatively new ecosystem for embedded systems may introduce integration challenges.

In the following pages, we discuss these alternatives in more detail, as they relate to development complexity, availability of community resources, and potential for levels of hardware control. By comparing the strengths and weaknesses of each contender, we aim to choose a programming language that not only meets our technical needs but also allows for scalable, sustainable development. This single design ensures that the finished system will deliver an immersed, real-time interactive experience.

3.2.2.1 C/C++ with ESP-IDF

C was created in the early 1970s when Dennis Ritchie developed it at Bell Labs to be used as an extension of the B language. C was originally developed in Bell Labs to allow the user to have low-level memory access while giving some of the benefits of a high-level language as well, making it ideal for embedded systems programming. The C programming language was standardized in 1989, also known as ANSI C.

In 1979, C++ was created by Bjarne Stroustrup, introducing object-oriented features while keeping the programming language's compatibility with native C. It was originally known as "C with Classes" until being standardized in 1998 and given the name C++, with the "++" representing an upgrade from C.

ESP-IDF (Espressif IoT Development Framework) was created for the ESP32 microcontroller in 2016. It was the official software development framework for ESP32 products using the RTOS operating system, which was created in 2003. Since its initial release, the Espressif team has built ESP-IDF out with more helpful libraries, tools, and APIs, allowing for more capabilities on the ESP32.

The Framework is built on FreeRTOS, allowing for real-time task scheduling, enabling the user to have more precise control over its prioritization and the timing of execution. A benefit of using the official ESP32 development framework is that we will be able to achieve better performance than many alternative frameworks, as we would be directly accessing our hardware without unnecessary abstraction layers interfering.

Using this framework would also allow for the best use of memory allocation, maximizing our available RAM for our project. The Espressif IoT Development Framework has exceptionally good documentation and online resources to help us through development.

ESP-IDF provides extensive access to all the needed ESP32 hardware features we would need for this project. This also gives us fine control over hardware timers, sensor polling, and interrupt handling, which will come in handy for low latency for the audio response after a string of water is broken.

ESP-IDF also allows for more advanced power management by enabling us to optimize battery life through selectively choosing what components are necessary and accessing deep sleep modes when needed.

However, ESP-IDF does have some glaring downsides, one being how steep the learning curve is. ESP-IDF is a verbose framework requiring more boilerplate code to get basic features and operations to work. Being a lower-level programming framework, it requires more manual management of hardware resources and memory. This can lead to more complexities, bugs, and a much longer development time.

Programming in the ESP-IDF framework requires proficiency with the C/C++ coding language, a deep understanding of embedded systems concepts, along with understanding the principles of using RTOS. In the case of our project, this would mean allocating more time for initial development and any testing and calibration, potentially prolonging a working prototype.

3.2.2.2 C++ with Arduino Framework

Arduino was created in 2005 by Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis. The idea behind Arduino was to create a basic tool that was accessible to students with no previous electronics or programming background. The first Arduino board was released in 2005 alongside the Official Arduino IDE that was forked from the wiring project. Being made for Arduino, the Arduino IDE was introduced to lower the hardware complexities and make C/C++ easier to program for microcontrollers by less experienced designers.

Official Arduino Support for the ESP32 began in 2016. This project attempted to bring the ease of programming on the Arduino boards to the more complex ESP32 line of microcontrollers, allowing for two separate microcontroller platforms to share the same platform. Arduino for the ESP32 is actively updated with stability features and library support. This makes the Arduino framework a more accessible entry way for programming on the ESP32.

The Arduino framework adds an additional compatibility layer on top of the original ESP-IDF framework mentioned in the section above. This compatibility layer allows us to strike a balance of having easier to understand code but at the same time having a lot of the benefits of using the original ESP-IDF framework. Another

large benefit of using the Arduino Framework is the access to numerous prebuilt libraries for popular sensors, LED control, and pumps. These libraries would mean that we wouldn't have to create these functions from scratch, and we could add to our project to significantly reduce development time. Most of these libraries are fairly well documented and have many online resources to reference.

The Arduino IDE also provides a quite easy to use development environment with simple compilation and uploading, making it easy for team members with varying amounts of programming knowledge to get involved and understand what's going on. Another large advantage of this framework is the large community behind it, which means that many niche problems we find during development are highly likely to already have published solutions somewhere on the internet, which will also help to reduce debugging time.

The Arduino Framework also has high cross-platform compatibility, meaning that our code will have high portability if we intend to switch to another Arduino Framework supported microcontroller in future versions of the project.

The Arduino Framework has its downsides. Due to the compatibility layer added on top of the original ESP-IDF development framework, we will incur small performance penalties; however, it should be noted this is nowhere near the inefficiency of using something like MicroPython. Another issue we'd face using this framework is the limited access we would have to more advanced features that Arduino simplifies and less control over certain timings. These features may be detrimental to achieving precise pump timings needed for this project.

Due to the simplification of the ESP32-IDF framework, we will have limited ability to optimize aspects of the system. Additionally, this simplified programming model may not be ideal for running operations in parallel with one another, which would be used for running multiple water strings or having the pumps activating at different speeds to balance out the water reservoirs.

3.2.2.3 MicroPython

MicroPython was created by Damien George with a Kickstarter campaign going up in 2013. MicroPython was originally written as a lighter version of Python 3 to be used in the realm of embedded devices and, more importantly, microcontrollers. Its first implementation was on a specialized MicroPython dedicated board. Going back to the source of MicroPython, Python was created by Guido Van Rossum in the late 1980s and was published in 1991. Python is famous for its readability and has quickly become one of the world's most versatile and used languages.

Official MicroPython support for the ESP32 began in 2017, with the primary port being worked on by the Espressif and MicroPython team. Over time, the ESP32 port has improved with support for additional chip features and performance and stability improvements. A big use of MicroPython is educational use for its very easy readability and its simplicity, which will greatly help in lowering development

time and making it more accessible to team members that may not have as much coding experience or more experience with Python.

Python's simple syntax and dynamic typing eliminate a significant amount of boilerplate code that is used in C/C++. The implementation of MicroPython introduces a Python interpreter that allows the user to control the hardware through Python. Another plus of using the Python development environment is using the quick development tool such as the built-in REPL (Read-Eval-Print Loop), which allows the user to utilize immediate testing on their hardware without having to worry about compilation cycles. This allows the user to adjust their parameters and see real-time results, which comes in handy for things like calibrating laser sensors or tuning the water pump and flow system.

MicroPython also includes many built-in modules for embedded use, such as modules for communicating with popular communication methods such as I2C, SPI, and UART and hardware peripherals. The language also includes automatic memory management with garbage collection, which eliminates common memory-related bugs such as memory leakage or dangling pointers that are common occurrences when coding in C/C++. For our use case, this could mean more stable long-term operation without memory-related crashes. MicroPython also features a modular approach, which will allow us to make clean separation within our code base for our various subsystems such as Audio, water flow, and sensor detection, making maintenance easier.

However, MicroPython doesn't come without its tradeoffs that could make a big impact, particularly for our Aqua Strings project. As we are using an interpreter language such as Python, this leads to significantly slower execution speed compared to compiled C/C++ code. This could present a big issue when it comes to time-sensitive tasks such as playing the audio response in reaction to a water string being broken. Having too heavy of a latency on the sound after a string is played could greatly affect the user experience.

Along with this latency, another downside of using an interpreter on the microcontroller is the memory resources needed to hold the interpreter. In MicroPython's case, this would be around 356KB of flash memory and at least 16KB of RAM out of the ESP32's 520KBs of RAM to hold just the interpreter. One of the biggest concerns of using MicroPython for our project is that due to the implementation of the automatic garbage collection system, timings can become quite unpredictable. When trying to create a responsive system giving the user near-instant feedback of playing a string, possible random pauses, and jittering audio due to delay in execution timing would not be ideal.

3.2.2.4 JavaScript (Espruino or Moddable SDK)

Brendan Eich developed JavaScript in 1995 at Netscape to enhance interactivity in web browsers. Initially called Mocha and later LiveScript, the language was renamed JavaScript to align with Java’s rising prominence, though the two share

no technical ties. Standardized as ECMAScript in 1997, JavaScript evolved through iterative updates, becoming a staple of web development. Its expansion into embedded systems began years later. In 2013, Gordon Williams introduced Espruino via a crowdfunding campaign, enabling JavaScript execution on microcontrollers by embedding an interpreter directly into firmware. By 2017, Espruino extended compatibility to ESP32 chips, allowing developers to prototype IoT applications using familiar web syntax. Around the same period, Moddable SDK emerged, prioritizing modern JavaScript standards (ES6+) and modular toolchains to merge web development workflows with embedded hardware design. Both frameworks aim to simplify microcontroller programming but can introduce trade-offs when applied to time-sensitive projects like Aqua Strings.

For teams with web development experience, JavaScript reduces initial barriers to firmware coding. The language's event-driven model aligns with interactive systems, where actions like interrupting a water stream must instantly trigger audio feedback. Espruino's interactive coding environment REPL (Read-Eval-Print Loop) supports real-time adjustments, such as tweaking sensor thresholds or refining sound delays without recompiling code. This agility accelerates prototyping, particularly for calibrating laser sensitivity or pump response times. Moddable SDK complements this with tools like xsbug, which monitors memory allocation and execution flow, aiding optimization. Native JSON handling also simplifies data management, enabling future expansions like user preference storage or remote control via Wi-Fi.

Despite these strengths, JavaScript struggles with real-time constraints. Interpreted execution introduces latency: tests show sensor interrupts on ESP32 take 2–3 milliseconds longer to process in Espruino than in C, risking audible lag during rapid interactions. Memory consumption poses another hurdle. Dynamic data types and automated garbage collection strain the ESP32's 520 KB RAM, risking instability during prolonged use, especially with multiple sensors and audio buffers active. Hardware integration demands extra effort, as most drivers for components like lasers or pumps are written in C/C++, requiring hybrid codebases that blend JavaScript with low-level modules.

Community support for embedded JavaScript remains limited compared to Arduino or MicroPython. Few pre-built libraries exist for peripherals such as I²S DACs or solenoid valves, forcing teams to adapt C libraries or develop custom solutions. Debugging timing conflicts, like overlapping interrupts between water-level sensors and pump controllers, becomes cumbersome without advanced tools like cycle-accurate simulators. Power efficiency also suffers: Espruino's interpreter draws 15–20% more current than equivalent C code during idle states, a critical issue for battery-dependent setups.

While JavaScript frameworks excel in rapid prototyping and lowering entry barriers, Aqua Strings' demand for precision and efficiency prioritizes languages like C or MicroPython. These alternatives offer tighter hardware control and predictable

performance, essential for maintaining seamless user interactions despite their steeper learning curves. JavaScript's role may shine in early-stage experimentation but falters under the rigorous demands of a polished, real-time system.

3.2.2.4 Rust

Rust was created as a personal project in 2006 by Mozilla programmer Graydon Hoare, and Mozilla officially picked up support for the project in 2009. The first stable release was released on May 15, 2015. The main goal in the development of Rust was to remove intense memory management while retaining the performance of C and C++.

Rust first started being integrated into the embedded world around 2016–2017 when the Embedded Devices Working Group was established within the Rust project. Seeing as it was meant to be high-performance, it makes sense to make it usable in an application with resource-limited microcontrollers. Official Rust embedded support was started in 2019–2020, with the initial releases only allowing limited hardware access. Since the initial release, however, the ecosystem has expanded its tooling, documentation, and library support. In 2021, Espressif, the company behind the ESP32, officially released Rust support on their chips. Rust has begun to gain traction in the embedded field for its reliability in applications where safety redundancy is important.

With how relatively new Rust is as a modern programming language, it doesn't have the development toolchain for the ESP32 that we see in other options such as Arduino and ESP-IDF. Using Rust could provide many advantages to our project, such as guaranteeing memory safety at compile time without having to worry about garbage collection like you would in C/C++. We could get comparable performance while eliminating the need to worry about common bugs and issues using the Rust ownership model and it borrow checker.

Another feature built into Rust that will benefit us is its advanced type system, which helps to prevent data races and similar concurrency bugs. This could be very helpful when implementing multiple tasks running in parallel. For example, in our project, we could have polling running for the laser sensors while simultaneously managing the reservoir water level management and pump flow control. Rust also offers modern programming language features such as pattern matching, zero-cost abstractions, and a macro system that we could utilize to simplify the development of complex functionality. The language is known for its robust error management, which could lead to more reliable operation for our very water-sensitive electronic system.

With safe Rust code, entire classes of undefined behavior that C/C++ programming is rampant with are eliminated at compile time, making debugging quicker and the system more stable. With all these advantages, using Rust in our ESP32 development comes with some issues. The ESP32 in the Rust community

is still developing compared to well-established alternatives such as Arduino and ESP-IDF. Although operational, the available frameworks and libraries are less mature and will feature more workarounds for ESP32-specific features.

Rust has a highly vertical learning curve, with hard-to-grasp concepts like ownership and borrowing that will require significant investment in learning. This will push the initial development time frame as members become accustomed to the language. The Rust ecosystem, although growing rapidly, still falls behind Arduino or ESP-IDF in ESP32-specific examples, tutorials, and community support.

On the practical front, Rust's expressive type system and safety checks translate into longer compilation times, which slows the development process. Rust on ESP32 embedded libraries is less mature than Arduino or ESP-IDF, with fewer high-level abstractions for mundane tasks. Library support for a single sensor or peripheral we wish to utilize in our project is similarly weaker, with additional custom implementation effort potentially required. Finally, installation and configuration of the Rust toolchain for ESP32 development is more complicated than with other options like Arduino, introducing overhead to project Initial Development.

3.2.2.6 Programming Option Selection for Project

Following close consideration of all the programming options for the ESP32, we have decided to use C++ with the Arduino Framework for our Aqua Strings application. This was chosen for a variety of key factors relevant to our specific application requirements and team expertise.

The Arduino framework provides the optimal combination of development speed and performance for our interactive waterfall display. Though ESP-IDF would provide marginally improved performance, the Arduino ecosystem's streamlined API and rich library environment will save us a huge amount of development time, letting us concentrate on the specific challenges of our water-based instrument instead of low-level implementation details. The stringent timing requirements of our sensor detection and audio response system require a compiled language over interpreted alternatives such as MicroPython or JavaScript. Arduino's C++ implementation provides us with nearly native performance for these time-sensitive operations but still offers helpful abstractions for common tasks.

While Rust also offers comparable performance with additional safety features, its greater learning curve and less mature ESP32 support would introduce unnecessary risk to our development timeline. A second reason is that several of our members are already familiar with the Arduino environment, reducing the learning curve and allowing for faster progress. The extensive Arduino community and the vast number of sensor interfacing, audio output control, and water flow system control libraries provide handy resources that will help us avoid typical pitfalls.

In our specific scenario, Arduino offers the best compromise between control and abstraction. It gives us sufficient access to the hardware capabilities of the ESP32 while simultaneously providing us with high-level functions that allow us to quickly develop our code. The small performance difference between Arduino and ESP-IDF will not impact user experience.

As we proceed with our project, we can opt to use direct ESP-IDF calls in the Arduino environment for performance-critical sections of code if necessary. This hybrid approach would give us the best of both worlds, Arduino's ease of development for most functionality and ESP-IDF's performance for more custom solutions.

3.2.3 Speaker Selection

Fundamentally, Aqua Strings' main objective is to be a playable instrument. There is a simple relationship one expects to occur when they play an instrument: When you interact with it, sound comes out. This can happen from a variety of different means: the contact between a mallet and wooden bar, the air traveling through a pipe, or the vibrations from a string. Regardless of how, this core relationship exists with all instruments. Aqua Strings will be no different. However, laser beams and water columns don't exhibit the same musical properties as a taut and tuned harp string. We've already discussed the potential ways to generate audio, but we need a means to play this noise. A normal instrument would use the physical properties of its materials to produce vibrations. To achieve the same effect, we will need to interface with an audio output device to play specific audio signals. When the sensor triggers a detection from a "string," it will communicate with the microcontroller to send an audio signal to an external device. This device needs to be able to produce an acceptable sound quality. The notes that the harp will play should be of distinguishable tone and volume. If necessary, the speaker can leverage the power of the external amplifier if it does not have one built-in. The speaker must also be within a certain size to allow the device to be concealed within the harp. From these factors, the following speakers will be considered: Passive Speaker (3W), Active Speaker (3W-10W), Exciter Speaker, Piezo Speaker, and High-Fidelity Audio Speaker (10W-20W).

3.2.3.1 Passive Speaker (8Ω, 3W)

Passive Speakers are the simplest option of all sound generation options. It consists of a diaphragm, voice coil, and magnet, which work in tandem to produce sound. When an audio signal is passed through the coil, it creates a varying magnetic field around the permanent magnet. The interaction from the magnet on the field oscillates the coil, which drives the diaphragm to vibrate and produce sound waves. And unlike active speakers, they do not possess a built-in amplifier. An external amplifier will be required to drive a passive speaker because a microcontroller, like an ESP32, will not be able to supply enough power by itself. These are low cost and commonly available in a variety of sizes and power ratings. For the sake of this comparison, we will look at passive speakers rated at 3W. However, there are plenty of alternatives at 5W and 10W ratings. One key

advantage is that the lack of an amplifier means we have complete control over the external amplifier we can use. As such, we have more flexibility in tuning the speaker's performance. There is also the possible of incorporating an amplifier into our PCB design, as an amplifier for a speaker in this application should not be too complex. The double-edged nature of this trait is that the speaker lacks the ability to function on its own. Overall, the advantages of passive speakers are apparent: low cost, availability, compatibility, and versatility. For such reasons, a small 8Ω, 3W passive speaker would be a great choice for this project.

3.2.3.2 Active Speaker (8Ω, 3W)

Active Speakers are the next progression of speakers when considering sound generation options. Unlike passive speakers, they feature a built-in amplifier. This internal amplifier circuit boosts low-power audio signals given by the microcontroller. This provides a number of advantages over its predecessor. For one, they can be powered by the ESP32 through a voltage regulator (which would be required anyhow). Additionally, they can be interfaced directly with the microcontroller's Digital-to-Analog Converter or Pulse Width Modulation output. Passive speakers have this capability as well, but Active Speakers can do this via 3.5mm audio jack or USB. As such, this device could be as simple as a plug-and-play with greater integration. Some models even include extraneous features like volume control knobs or equalization settings that could provide increased sound customization options. Another advantage is that active speakers can deliver clearer and louder audio. This would improve the playback of the music of the harp. As a result, these models typically require more power to operate and may not be ideal for battery power. A key disadvantage of these speakers is that, in addition to a voltage regulator required to power the device, an external module will be required to interface these speakers. Having the ability to connect via 3.5mm audio jack or USB is a nice option if our microcontroller had those ports. However, a small ESP32 Devkit only has one port: a micro-USB connection to power and program. Unfortunately, in our case, our microcontroller cannot leverage this advantage. We will need an external module regardless of whether the speaker has an internal amplifier or not. For this reason, a passive speaker seems to be the better option. If both products require external circuitry to properly operate, it only makes sense to choose the one that is more affordable and more available. However, there is one option of connection that we have not considered. The ESP32 can leverage the power of Bluetooth to wirelessly connect with devices. If we were to connect an active speaker via Bluetooth, we would surpass the need to connect via audio jack or USB, reducing the additional circuitry required to operate. This method would also allow us to place the speaker anywhere within the harp (or outside it) with little concern for its connection. However, two major considerations would arise for utilizing Bluetooth in this way. As previously mentioned, latency is a serious concern for this project. Our specification goals require quick processing of signals and playback. Wireless connection via Bluetooth risks latency between the microcontroller and speaker. While this would likely be in the milliseconds, the longer the delay, the more the experience is diminished. When an instrument is played, the soundwaves are near instant. We

want Aqua Strings to be on this level. Additionally, a wireless complication would complicate the software end of development, as the audio transmission now needs to occur via Bluetooth. Before, the speaker would simply need to connect to a few specific analog outputs from the ESP32. Now, the programming would need to utilize Bluetooth libraries to interface with the device. And this doesn't even consider that the speaker would still require a power source. Ultimately, an active speaker is still a solid potential option, but its complexity is to be considered alongside its greater audio fidelity.

3.2.3.3 Exciter Speaker

Exciter Speakers are a special type of speakers that conduct their sound waves through solid surfaces. Instead of traditionally emitting sound through the medium of air, this speaker turns the surface into a sound-producing element. Fundamentally, an exciter speaker is very similar to a passive speaker. It consists of similar elements, like a voice coil and a magnet, and the physical reaction that produces audio is the same. However, unlike a traditional speaker, the oscillating coil doesn't drive a speaker cone. Instead, exciter speakers possess a mounting plate that transfers the vibrations from the coil to a surface. The surface itself then acts as the speaker's diaphragm, producing sound. This would be the most immersive of the speaker's options, as the harp's sound would resonate from its frame. That is one of its biggest advantages: the unique and immersive sound that would change characteristics and resonance based on the shape of the frame. However, that is one of its biggest potential concerns. At this stage in the design, the frame of the harp is yet to be finalized. Prototypes of its general shape exist, but we have no idea how the surface will conduct sound in such a configuration. There is also the possibility that the large amount of water from each reservoir would absorb the soundwaves and dampen the volume. Additionally, vibrations produced in the water could disrupt the flow of the water columns. It is tempting to consider an option as interesting as this, but we have no idea how it will sound until we finalize and begin testing the frame. At this stage in the design, there is unfortunately too much we do not know about the frame to understand how well this application would function. However, this consideration is worth returning to because of how experimentation-dependent it is. Not only does the material and size of the mounting surface impact clarity, as does the mounting spot within the frame. Determining its application would best occur later in development. As a result, we will move forward with another option in the meantime. However, provided there is room in development, the exciter speaker will be considered for its unique and organic sound.

3.2.3.4 Piezoelectric Speaker

Piezoelectric (piezo) speakers are a type of speaker that generates audio using the piezoelectric effect. Unlike traditional speakers that rely on a voice coil and a magnet, they utilize a thin piezoelectric ceramic disc that produces vibrations when subject to an electrical signal. This occurs according to the piezoelectric effect, which is when a voltage is applied to a piezoelectric material (like quartz), causing that material to rapidly deform. This deformation causes mechanical vibrations that

generate sound. These vibrations tend to produce high-pitched tones and beeping sounds, such as those commonly present in alarms and watches. A piezo speaker is an incredibly simple device, and as such, has a host of benefits. For one, it is extremely power-efficient, requiring little current to operate and is ideal for battery-powered situations. It is compact and requires little room to function, meaning it can be placed throughout the harp's frame. While limited it frequency, the tones these speakers produce are sharp and clear. This could be useful for ensuring the audio is heard despite the sound from the other moving components (i.e., rushing water, water pump operation). And many piezo speakers can run directly off the microcontroller's Pulse Width Modulation output, meaning it doesn't generally require an external amplifier to operate. However, by being such a simple and efficient device, it possesses several limitations too. One of its greatest disadvantages is its limited sound quality. While it is effective at simple tones, it cannot achieve a fuller range of audio due to its means of generating sound. Specifically, it would struggle at producing lower-pitched tones and any level of bass. Additionally, because their audio is so biased towards high frequency, they are great for beeps and chimes, but not as applicable towards more complex musical tones. If we wanted the sound of a string being plucked to have a complex, vibrating tone, we are unsure if such a speaker could produce that note. Another disadvantage is that some piezo speakers, despite operating very efficiently with a low current, may require higher voltages to boost their circuit for proper volume levels. This voltage range could reach 12V or above, which would certainly require a voltage regulator. For the applications of this project, if we wanted the output of the harp to be short, percussive, and high-pitched, we would choose a piezoelectric speaker. However, Aqua Strings at its core is resemblant to a harp and its audio output should reflect that. Confining the music to high-pitched beeps would limit the project's effect and stifle the experience. For this reason, piezoelectric speakers will most likely not be considered as the means of generating the tones and audio for this project.

3.2.3.5 High Fidelity Speaker

High fidelity (hi-fi) speakers are the most advanced option to produce the highest level of quality. They feature much more complex electronics than the previous speakers to produce greater audio quality. Specifically, they consist of multiple driver components such as woofers for lower frequencies, tweeters for higher frequencies, and midrange for the main body of the sound. When combined with the microcontroller's Digital-to-Analog Converter and an amplifier, the hi-fi speakers could most faithfully produce musical tones. This option is similar to what one might purchase for their computer or television audio system. Their advantages mainly come from their complex hardware. They can produce a rich and superior sound with support across the full range of frequency. They would also be the most ideal for music playback, as this higher quality could sound as close to a genuine instrument as possible. It also could be expanded upon and upgraded with additional units such as subwoofers or external amplifiers. If this project was entirely audio focused, then this would be a definite pick. However, that is not our only focus. We must balance the project's other features and

consider its limitations. And that is where hi-fi speakers' disadvantages begin to come in. For one, they consume a very large amount of power compared to their alternatives, so much so that they might need to run on a different power source. Additionally, they are large and bulky. They could not be placed within the harp's frame and would simply need to sit outside the instrument. This would take away from the harp's effect as the audio playback would occur away from the strings. These speakers also tend to be rather expensive, some ranging almost ten times as much as some passive speakers. This is outside the range of our budget for the speaker. Ultimately, hi-fi speakers are overkill for the level of audio quality we are expecting for this project. While we do want the instrument to play with a reasonable level of richness and clarity, we do not want to sacrifice the rest of the project to achieve it. Unfortunately, high fidelity speakers will most likely not be considered for this project because of their high-power consumption and unnecessary level of audio quality.

Table 3.2.3.5: *Comparison of Speakers*

Feature	Passive Speaker	Active Speaker	Exciter Speaker	Piezo Speaker	Hi-Fi Speaker
Cost (general)	~ $3 - $8	~ $10 - $20	~$3 - $20	$0.5 - $3	$30 - $100
Sound Quality	Good	Good - Excellent	Surface-dependent	Poor - Good	Excellent
Size	Small - Medium	Medium - Large	Small	Very Small	Very Large
Complexity to Integrate	Moderate (need amplifier)	Moderate (need 3.5mm or Bluetooth)	Moderate (surface/ frame dependent)	Very Low (near plug and play)	High (need amplifier/ external power)
Power Consumption	Moderate (3 – 10W)	High (10 – 30W)	Moderate (1 – 10W)	Very Low (0.1 – 1W)	Very High (20 – 100W)
ESP32 Compatibility	Yes (with amplifier + module)	Yes (with amplifier + module/BT)	Yes (with amplifier)	Yes (PWM output)	Yes (with DAC + amplifier)

Best Feature	Simple application, average sound quality	Higher-than-average quality, Bluetooth	Unique sound profile, immersive	Incredibly simple, low power	Excellent audio quality, great range

3.2.4 Water String Detection Laser & Sensor Selection

For the Aqua Strings project to become fully functional the sensors used to detect an interruption in each water string need to be reliable, affordable, practical, and compatible with the other chosen electronics. In section 3.1 the different types of sensors that could be considered were researched and our findings were compared leading to visible LED lasers being selected due to their numerous advantages, including the added visibility when aligning the beam and the receiver. Now that the type of sensor has been selected, the next step would be to compare specific parts and choose the proper part based on a multitude of factors such as cost, delivery time, wavelength (color of beam), performance, device specs. Similarly, the photodiodes also need to be compared and selected based on a similar list of qualities. They must have a sensitivity range that the laser beam falls into, which is roughly between 400-580nm since green and blue laser have been chosen due to their reduced absorption in water. The majority of these options are not inherently waterproof, for both laser transmitters and receivers, it is important to note the costs that come along with waterproofing these materials verse paying the price of these already waterproofed components. This section will lay out each component and the advantages and disadvantages it'll bring to our project.

3.2.4.1 Laser Beam Component Selection

This project will have a handful of electronic components that will need to be powered so finding a component with a low power demand while also maintaining a strong output at an affordable price is a main priority. Even though green and blue lasers have slight differences in visibility and their interactive properties it is still important to note and compare those discrepancies. Green lasers are typically a bit more visible to the naked human eye, which gives these lasers a small advantage when aligning and implementing the system. Blue lasers are still sufficient in terms of maintaining the functionality of the project and are typically a bit cheaper compared to green lasers. Both disadvantages are basically negligible, but they are still factors that exist.

3.2.4.1.1 Green Laser – GH05210H2K

The GH05210H2k is a laser diode that produces a laser beam at a wavelength of 520nm meaning it falls on the visible light spectrum as a green color. Due to this green color, the beam will have great visibility which will help with alignment (as mentioned previously) and will add a color to the water string enhancing user experience. This specific is a suitable option for our project. It was found online for

$8.00 on the website Laser Tree however delivery added a few more dollars depending on how fast the product will be shipped, bring the overall cost to about $12.00 roughly. At this cost, this laser would operate as expected with its performance meeting expectations and needs. Its cost is slightly high, but it does fully out-rule this beam from the equation.

3.2.4.1.2 Green Laser – GM532-50B

The GM532-50B is a green dot laser diode module with slightly long dimensions when compared to the size of the GH05210H2K laser diode. This particular model is a dot laser meaning the laser beam isn't as visible as other options but the photodiode on the receiving end would still be able to read the input as the same since the green dot appears on the end. An advantage to this would be that the user wouldn't see the laser until they swipe their hand across the water string, and it would appear on their hand, giving them an illusion of how the water being there. This, however, doesn't affect the functionality of the project, so it is more so just a cosmetic design the team could make. These laser beam modules were found on Amazon for $10.00 per unit and delivery cost is free since our team members have an Amazon Prime subscription which includes free shipping. This puts the overall cost per unit still at $10.00, which is less than the GH05210H2K green laser diode making it a more affordable option. Performance measures are about the same with similar power requirements (see the table in this section) but this module is slightly more appealing due to the small difference in cost.

3.2.4.1.3 Blue Laser – B0CYL53LVG

The B0CYL53LVG is fairly similar to the previously mentioned laser diodes but instead it produces a laser beam at a wavelength of 405nm, which is a blue color on the visible light spectrum. This specific model can be purchased as either a line beam, dot beam or cross beam. These different line types only serve as cosmetic choices and do not affect the functionality of the project. Blue beams act similarly to green beams since they have a reduced absorption rate within water but there still is a slight reduction in visibility to the user. Even with this slight difference there is no change in the performance of the overall project. These laser modules were found for $8.80 per unit on Amazon and delivery costs are already included since our team members are prime members, leaving the total per unit price at $8.80. Since this product is also on Amazon its delivery time is only two days, beating out other websites. This option is a solid choice for our blue laser beams since they are available in multiple different line types along with its lowered cost.

3.2.4.1.4 Blue Laser - B07M9DTSF7

The B07M9DTSF7 is another blue laser diode module at a wavelength of 405nm. This model comes as a blue dot beam meaning the color is mainly seen at the end of the beam when it hits the receiver (photodiode) on the opposing end, there's no functionality issues. This module is quite identical to all the other laser modules listed in this section since it has relatively the same dimensions and performance.

This transmitter was found on Amazon for a price of $11.99 and since it is ordered from Amazon it would have shipping fees already included. Its reliable power output and high stream stability make it a contender for actually being implemented into our Aqua Strings project. This higher cost, however, is a drawback so other options are likely to be chosen first before our team chooses the B07M9DTSF7.

3.2.4.1.5 Red Laser – 1528-1391-ND

The 1528-1391-ND is another red laser beam which means the mean has a wavelength of 650nm on the visible light spectrum. In previous sections it was mentioned that green and blue lasers would most likely be our best option since there is a reduction in the absorption rate of the beam into the water. Red beams do have a higher absorption rate, but visible light beams are also rated for a long distance which is about 6 feet (dependent on the specific laser transmitter) compared to the 1.5 feet for IR lasers. The draw factor to red laser diodes is how much cheaper they are compared to other colored beams since they are more widely produced. Our team has decided to purchase a red beam along with our green and blue laser beams to run our own experiment to see how much more the red beams are absorbed and if it's worth using 650nm beams instead. These specific transmitters were found for $5.95 on DigiKey, not including shipping. This is only one option of the many that were found and there are packs of red beams that can lower the unit price if purchased for all 8 water strings.

Table 3.2.4.1.5.1: *Laser Comparison*

Specification	**Sensor 3.2.4.1.1**	**Sensor 3.2.4.1.2**	**Sensor 3.2.4.1.3**	**Sensor 3.2.4.1.4**	**Sensor 3.2.4.1.5**
Color (Wavelength)	Green (520nm)	Green (532nm)	Blue (405nm)	Blue (405nm)	Red (650nm)
Power Output	15 mW	5 mW	5 mW	20 mW	5 mW
Voltage	5.8 V	3 - 3.7 V	3 – 5 V	3 – 5 V	2.8 - 5.2 V
Current	35 mA	200 mA	--	< 120mA	25 mA
Laser Type	Dot	Dot	Line, Cross, Dot	Dot	Dot

Dimensions	--	1.3” L x 1.3” W	12mm x 30mm	14mm x 45mm	10 mm x 31 mm
Operating Temperature	-10°C to 60°C	0°C to 50°C	15°C to 35°C	10°C to 40°C	-10°C to 40°C
Cost	$8.00 + shipping	$10.00	$8.00	$11.99	$5.95 + shipping

*See each data sheet for more information

3.2.4.2 Photodiode Component Selection

For an accurate detection system, a reliable photodiode is needed on the opposing end of the laser beam that is traveling through the water string. Photodiodes are made up of semiconductors which convert light into a form of electrical current using the photoelectric effect. When a beam comes into contact with the photodiodes p-n junction, it will then generate electron hole pairs causing current to flow. These diodes are highly effective for light detection because the amount of current that will be produced from these sensors is directly proportional to the intensity of the light beam it is reading. For applications where photodiodes are used to detect interruptions between a laser beam and a receiver, these photodiodes play a crucial role since they record whether they are in contact with the beam or not. When there is no interruption, the photodiode maintains a constant current value with no pattern change. When there is an interruption there will then be a fluctuation in the amount of current that is flowing causing the system to note there was an interruption. The microcontroller will read this occurrence and act accordingly, in this case it will trigger a note to play out of the speaker system. Due to the instantaneous reaction time and sensitivity, photodiodes remain a great choice when needing to read input based on the visibility of a laser beam.

3.2.4.2.1 Part 475-BPW34-ND

The 475-BPW34-ND is a mass-produced silicone PIN photodiode that is widely known for its fast detection time and sensitivity. This specific photodiode has an operation wavelength range that includes the visible light to the near sided inferred spectrum, which includes all the listed laser diode parts in section 3.2.4.1. One of the main advantages is its relatively low cost, which was found at $1.14 per unit on DigiKey, making it an affordable choice in relation to the needs of our project. Another advantage is its small dimensions and high efficiency which allows for a straightforward integration into the system. However, one disadvantage is how the component isn’t inherently waterproof which would require extra waterproofing measures to be taken on our end or having to isolate the component from the water completely. This photodiode has a sensitivity to a wide range of wavelengths on

the spectrum, incorporating the project's needs, but may require extra steps when working with water.

3.2.4.2.2 Part 475-SFH213-ND

The 475-SFH213-ND is another photodiode that has a sensitivity range within the visible to inferred spectrum, but it does have a peak sensitivity in the inferred range, meaning it is more suited for IR beam rather than visible light. This doesn't mean it wouldn't work but it may lead our team towards other components. An advantage to this diode is its low cost, which was found at $0.85 per unit on DigiKey, making it one of the cheapest listed items here. A function of this diode to note is its function with low dark current in low light areas but our team doesn't have to worry about low light areas when assembling our project. Just like the previous photodiode, this component is not manufactured to be waterproof so it would need additional waterproof measures for it to be submerged in water. The 475-SFH213-ND most likely won't be the option chosen for out water string detection since it seems not to meet our needs as well as the other options that were researched.

3.2.4.2.3 Part 751-1013-ND

The 751-1013-ND is considered to a be a premium photodiode since it offers a higher sensitivity and accuracy when compared to other options. It serves as a strong component for this type of interruption detection due to its stable and high-performance readings. One of these photodiodes will cost $11.38 per unit from DigiKey making it the most expensive option by far out of our listed options. Our project only consists of sending a laser beam about a foot distance to which it will be then read so this type of diode may be overkill and an unnecessary purchase. Also at this price point, this photodiode does not come manufactured to be waterproof so there would be additional costs needed to make this option capable of being submerged.

3.2.4.2.4 Part B07VS3FLLF

The B07VS3FLLF is the fourth option of photodiodes, and it is a high sensitivity diode that is designed for general light detection, perfect for the needs of Aqua Strings. These photodiodes are the cheapest out of all the options listed and can be purchased at $7.29 for 20 diodes, making the unit cost be about $0.36. These packages can be purchased with Amazon which includes fast free shipping unlike DigiKey (where the other three options would be purchased from). The downside to these photodiodes is their lack of documentation compared to the other options so by purchasing them our group would be taking a gamble at the reliability since there is much information out there about the uses of this model. It also lacks waterproofing options when being manufactured so it would also require additional costs and steps if the component were used in or around water. These photodiodes are very cheap and affordable but due to their lack of testing with this type of application it may not be a top choice for our project when we can find a more documented component that is relatively the same price.

Table 3.2.4.2.4.1: *Photodiode Comparison*

Specification	Photodiode 3.2.4.2.1	Photodiode 3.2.4.2.2	Photodiode 3.2.4.2.3	Photodiode 3.2.4.2.4
Wavelength Range	430nm – 1100nm	400 nm – 1100 nm	420 nm – 675 nm	400 nm – 1100 nm
Peak Wavelength	850 nm	850 nm	565 nm	940 nm
Reverse Voltage	32 V	50 V	10 V	0.5 V
Current	Dark, 2nA	Dark, 1nA	Dark, 2nA	5 mA
Viewing Angle	120°	20°	100°	40°
Dimensions	5.4 mm	37 mm x 5.9 mm	--	26 mm x 0.5 mm
Temperature	-40°C to 85°C	40°C to 100°C	-40°C to 125°C	--
Response Time	20ns	5 ns	--	--
Active Area	7 mm^2	1 mm^2	--	--
Cost	$1.14 + shipping	$0.85 + shipping	$11.38 + shipping	$7.29 ($0.36 pu)

3.2.5 Water Level Sensor Selection

Water level sensors play a crucial role in the functionality of the Aqua Strings project since they ensure both the upper and lower reservoirs have at least the minimum amount of water needed for smooth operation. This project is focused on the ability to swipe a hand through strings of water and for that water to be flowing the whole system needs to have water balanced throughout without any water

pooling in one specific location. This requires the water pump, that will be implemented, to work when needed and not overwork itself. To guarantee the water pump won't be over pumping water from the lower to upper reservoir by the utilization of water level sensors. These sensors will control the on or off state of the water pump depending on the amount of water in each reservoir, ensuring proper use. There are various types of water level sensors including resistive, capacitive and float designs, all have different types of uses and applications but will manage the flow of water. Choosing the right sensor type will ensure the water flow for the water strings works as expected aiding to the user's experience.

3.2.5.1 Part 1528-4965-ND

The 1528-4965-ND is a digital water level sensor that works by reading the resistance in the water. The sensor itself consists of exposed interlocking traces that are considered an open circuit when not in contact with water but when the sensor is submerged the water completes the circuit, allowing current to flow. When the current starts flowing the microcontroller attached will take note of this and flag this detection allowing for the water pump to be properly controlled. It is a simple and direct detection method that could be easily implemented for our needs. However, if these sensors remain in the water consistent, they become prone to corrosion and degrading over time which reducing the accuracy of the device. This type of sensor is on the cheaper side making it a great option for the Aqua Strigs project but due to the possibility of corrosion over time our team would need to make sure to take these sensors out of water when not in use.

3.2.5.2 Part 42AK5700

The 42AK5700 is another digital water level sensor that operates by measuring capacitance or resistance to determine the level of a liquid. Digital sensors do not rely on moving mechanical parts unlike float sensors, which makes it more intuitive. This specific model operates at a 5-volt DC and outputs either a high or low value depending on the input the sensor reads. The microcontroller then reads then output and can act accordingly, such as adjusting the state of a water pump. However, this specific component is documented as thoroughly as out other digital sensor option which can cause issues when implementing the sensor comes around. With this lack of documentation, it is uncertain to tell if it is fully waterproof or if only a portion is, which is another factor to consider. While this option produces a reliable output that the microcontroller would easily be able to read, other factors such as the documentation and accreditation of this sensor may not make it the top chose our Aqua Strings.

3.2.5.3 Part 2010-1043-ND

The 2010-1043-ND is a float sensor that works by mechanically monitoring the position of a floating object and switch the flow of water either on or off according to its position. This model does not have any electronic or digital component to it and operated solely on mechanical pieces which can make the design of the system more simplistic. This straightforward mechanical approach it very reliable and resistant to electrical malfunctions, unlike the digital sensors. These electrical

malfunctions could include any false triggering due to water temperature, water impurities, or corrosion of electrical parts. A major advantage to these sensors is it waterproof design since it does not use electronics, making an ideal option for in water uses. However, due to its reliance on the movement of different mechanical parts this affects the response time slightly, which may not hurt the functionality of the system but still a factor to consider.

3.2.5.4 Part B07G13JB57

The B07G13JB57 combines both digital and float technology into one sensor. This sensor works by using a mechanical float switch to detect water level but instead of using a physical switch to block off water flow it uses a digital signal the microcontroller can read to turn the water pump on or off. This design is intended for water use meaning it is fully waterproof since the sensor itself is mechanically based and the connecting wires are properly protected. Resistive and capacitive sensors take input through the completion of a circuit where with this combination of design its input is mechanical, and output is electrical. One of the main advantages of this sensor is its durability but overtime it is still subject to normal wear and tear. This would be a great option for the Aqua Strings project since it takes the best parts of both resistance/ capacitance sensors and mechanical float switches and combines them into one reliable sensor at a reasonable price.

Table 3.2.5.5: *Water Level Comparison*

Specification	**Sensor 3.2.5.4**	**Sensor 3.2.5.4**	**Sensor 3.2.5.4**	**Sensor 3.2.5.4**
Detection Type	Digital	Digital	Switch	Digital Float
Output Signal	Digital (High/Low)	Digital (High/Low)	Open/ Closed Circuit	Digital (High/Low)
Voltage	3V – 5V DC	5V DC	--	5V DC
Fully Waterproof	No	--	Yes	Yes
Dimensions	64mm x 20mm x 7mm	20mm x 133mm	53mm x 24mm	19mm x 45mm

Temperature	--	-40°C to 105°C	-20°C to 80°C	-20°C to 80°C
Cost	$1.95 + shipping	$8.95 + shipping	$3.65 + shipping	$9.90 ($3.30 pu)

3.2.6 Water Pump Selection

A key element for our Aqua Strings project is incorporating continuous water movement. The water pump in Aqua Strings serves the purpose of providing consistent transportation of the water from the bottom reservoir back to the top reservoir. This water flow continuity is very important in creating stable water streams not only to generate stable water streams for a fountain visual, but also to maintain a high accuracy of water detection to successfully execute the water's interactive feature. The design of our project includes an upper reservoir that has water piping for facilitating the eight vertical streams of water. Our water pumps will be located on the bottom portion of the project, and it will be constantly recycling the water that has just been released from the waterfall back to the top portion to be utilized again.

During the selection process of an appropriate water pump, there are a variety of technical specifications that need to be evaluated. One of the primary considerations is the water pump’s flow rate, which is responsible for the speed and quantity of water moving through the system. The approximate height for the project is about two feet tall, with around two gallons of circulating water across 8 small streams. A typical residential home sink can be found with a flow rate of 1.5 to 2 gallons per minute (GPM). With this taken into account, an appropriate flow rate to maintain a steady stream for our project should be a slightly higher amount. As the quantity of water requirement is not a large quantity, but for 8 separate streams we expect to utilize a less powerful water pump than for high volumes of water, such as fountains. Having an estimated flow rate of about 2.3-2.5 gallons per minute (GPM) is an accurate and also useful specification for selecting an optimized water pump. In addition, adjustability of the flow rate is a crucial component. Our project's stream pressure for all 8 waterfalls requires minor adjustments and changes to create just the right stream pattern for sensor detection. Having such flexibility to make small adjustments is important to create the visual effect and sensor performance.

Another very crucial parameter is the head height of the water pump, which is the ability for the water to push up vertically against gravity. Since Aqua Strings is about 2 feet tall, the head height of the pump should be able to support at least three feet. The reason for this is that the pump has to take into account the loss in pressure from the tubing of the frame. An adjustable head height for a water pump

will also be useful as it will allow easy calibration throughout our testing process of water pressure.

As the main function of Aqua Strings is to produce a musical tone from hand movement, it is important to consider that the water pump operates at a quiet noise level, to prevent the background noise from interfering this the sound output of the harp. Even though the noise of the musical output tone can be adjusted with selection of the audio system, it is overall extremely important to minimize the operational noise whenever possible to create the optimal auditory experience for users. Additionally, noise of a water pump correlates with the size and GPH capacity, so selecting a smaller water pump has a positive correlation with noise and cost.

3.2.6.1 Condensate Water Pump

A condensate water pump is a specialized type of pump designed primarily for handling the collection and movement of condensate water, often found in heating, ventilation, and air conditioning (HVAC) systems, dehumidifiers, and refrigeration units. Condensate pumps are optimized for low-flow applications where water is collected intermittently and needs to be transported to a designated drainage or recirculation point. For our water harp project that involves continuously moving water from the lower reservoir to the upper reservoir to maintain consistent water streams, a condensate pump has qualities that would allow it to be a viable option.

The operation of a condensate pump involves an internal reservoir or sump where water accumulates. Once the water reaches a certain level, a float switch or sensor is triggered, activating the pump motor. The pump uses an impeller mechanism, similar to other centrifugal pumps, to generate the required pressure to move the water through a discharge line. However, condensate pumps have a piston-style mechanism, which is designed to manage smaller volumes of water at lower flow rates. This is achieved by creating suction that draws water into the pump chamber, where the impeller then pressurizes the water and pushes it through the outlet. Many condensate pumps also include a check valve to prevent backflow, which is critical in maintaining the directionality of the water flow within a closed-loop system.

When considering a condensate pump for Aqua Strings, the critical specifications include flow rate, head height, and activation mechanism. Condensate pumps are generally designed for flow rates ranging from 0.5 to 2 GPM and head heights up to 20 feet. For Aqua Strings which requires a pump with a flow rate around 2.5 GPM and a head height of approximately 3 feet, a properly sized condensate pump would be able to meet these needs. The intermittent operation mode of most condensate pumps that is triggered by the float switch could be adapted for continuous operation if the pump allows for manual override or if the reservoir is designed to keep the pump running steadily. This change would ensure that the water streams remain consistent and uninterrupted, which is crucial for the accurate functioning of the motion sensors and the overall musical experience of the water harp.

One of the strengths of a condensate pump is its ability to handle small volumes of water efficiently while maintaining a compact form factor, which would be beneficial as the pump must fit within the lower reservoir without occupying excessive space. Additionally, condensate pumps are typically designed for quiet operation, which is also a viable option considering our project is sound based. The built-in float switch also adds a level of automation and safety as it prevents the pump from running dry and potentially overheating.

However, there are some limitations to using a condensate pump for our project. The primary concern is that condensate pumps are not inherently designed for continuous operation. They are intended to activate sporadically when water accumulates to a specific level, and prolonged use could lead to premature wear on the motor and internal components. Additionally, the flow rate control on condensate pumps is often limited, as they are engineered for specific drainage applications rather than dynamic water features. This could pose a challenge in fine-tuning the water streams of Aqua Strings, where adjustability is needed to balance the flow for optimal sensor performance and visual effect. Another consideration is the pump's sensitivity to small pieces of debris. Since condensate pumps often deal with relatively clean condensate water, they might not be adaptable with less clean water. In Aqua Strings, where water is being recycled continuously, the risk of small particles or impurities entering the system is higher, potentially leading to clogs or reduced performance is using a condensate pump.

While a condensate pump could technically fulfill the requirements of Aqua Strings in terms of flow rate and head height, its limitations in continuous operation and flow adjustability does make it a less ideal choice compared to a traditional submersible fountain pump. If a condensate pump were to be used, there would be additional design considerations that would be necessary like implementing a bypass to maintain constant water flow or integrating filtration to prevent debris buildup. Ultimately, the decision would hinge on balancing the pump's compact size and quiet operation against the need for reliability and precise flow control in a dynamic water-based musical instrument.

3.2.6.2 Submersible Water Pump

A submersible water pump is a specific type of centrifugal pump that is designed to operate in fully submerged water. Traditional water pumps rely on suction of water, but submersible pumps operate by pushing the water to the surface by converting the rotational energy into pressure energy. This process is achieved through the operation of an internal impeller, which is connected to a sealed electric motor. When this motor is activated, the impeller rotates and creates a low-pressure area at the pump's inlet. Water is then drawn into the pump and leaves through a volute, where the water is then directed through the outlet and into the connected tubing system. This mechanism allows the pump to efficiently move water vertically, overcoming gravity and reaching the top reservoir of our project.

Specifically for Aqua Strings, a water pump with a flow rate of approximately 2.3 gallons per minute (GPM) and a head height of about three feet was identified as

optimal for achieving the required performance criteria for the consistent flow of eight strings of water. The 2.3 GPM flow rate translates to approximately 138 gallons per hour (GPH), providing enough volume to create visible and even water streams without causing inconsistencies that could interfere with the motion sensors' accuracy. The three-foot head height capacity is just as important, as it defines the maximum vertical distance the pump can move water while maintaining the necessary pressure. Given that our project stands about two feet tall, the three-foot head height offers a margin of safety in order to account for pressure losses due to friction within the PVC tubing or potential resistance from flow control precautions. This head height specification also allows for reliable delivery of water to the upper reservoir, ensuring all eight streams remain uniform and responsive to user interaction.

One of the potential water pumps under consideration for Aqua Strings is the "Smartpond" brand pump, which meets these precise specifications. The Smartpond pump is designed for small water features and is both compact and efficient, aligning well with the space and power constraints of the Aqua Strings design. The compact size of the pump allows it to fit comfortably within the bottom reservoir without obstructing water collection or requiring a complex installation. In addition, the Smartpond pump features adjustable flow control, which is crucial for fine-tuning the water output to achieve the desired stream stability. This adjustability allows our team to calibrate the water flow during testing and setup, ensuring the water streams are both visually appealing for the user and effective in triggering the sensors accurately. Submersible pumps like the Smartpond model offer the advantage of quiet operation. Since the pump is fully submerged, the surrounding water acts as a natural sound dampener, which is perfect for minimizing the mechanical noise that could interfere with the musical output of Aqua Strings. As detailed previously, creating a quiet operation is particularly beneficial for a musical instrument, where maintaining an unobstructed auditory experience is important for user experience.

Beyond the performance, submersible pumps are also advantageous for their efficiency and ease of maintenance. The Smartpond pump, like many modern submersible pumps, is designed to be energy efficient, and is beneficial for our project because we would require operation with limited power capacity. The pump's design typically includes a sealed housing to prevent water intrusion into the motor which is beneficial to safety and durability. Routine maintenance for submersible pumps is minimal, compared to other pumps in the market. Cleaning for submersible pumps is limited to small maintenance of the intake screen, to prevent potential blockages from debris. This low-maintenance requirement will allow our team to focus on the project's performance with hardware and sensor accuracy, rather than frequent servicing of the water pump.

Chapter 4: Standards and Design Constraints

Definitionally, a design standard refers to a certain set of guidelines or specifications that ensure a project meets a certain level of quality, safety, and performance. For our Aqua Strings project, these standards are crucial for ensuring that our design stays as reliable and functional as possible. Design constraints, on the other hand, represent limitations that then affect the design process of our project. Constraints can come in many forms, such as time, budget, resources, or technical limitations, and need to be managed to ensure the success of the project. In the case of our project that is primarily electrical-based, our group must adhere to standards set by organizations like IEEE (Institute of Electrical and Electronics Engineers) for our PCB to ensure that our design complies with industry regulations. Additionally, safety and manufacturability play a significant role, as our design must be safe, practical, and cost-effective to produce. The main constraints in our project are time and finances, both of which add limits on how quickly we can complete the design and how much we can invest in materials, components, and testing.

4.1 Design Standards

4.1.1 PCB Design Standards

In order to have a well fabricated PCB, there are a few design standards set by our selected manufacturer, JLCPCB, that our group must adhere to. These key specifications have an impact on our design choices and must be taken into account when designing our PCB to send out for manufacturing. IPC-2221 standard is the guideline that dictates the general PCB design requirements that include the material selection, trace routing, spacing, and hole sizes. More specific standards, such as IPC-2222, focus on only rigid PCBs, which we will not be utilizing. Another critical standard is IPC-6012, which defines the performance and qualification requirements for high-reliability PCBs. This standard makes sure that all of the boards meet quality durability expectations, which is especially important for our senior design project, for the quality of our hardware integration.

Overall electrical performance standards are also crucial in PCB design to maintain signal integrity and power efficiency. IPC-2152 provides guidelines on appropriate trace widths and thicknesses for handling different current levels, preventing overheating and unexpected voltage drops across the board. For manufacturability, the IPC-7351 standard provides rules for surface-mount component footprints, and IPC-A-600 defines the acceptance criteria for PCB fabrication. Additionally, ISO 9001 and ISO 13485 certifications ensure that PCB manufacturers follow quality management practices to produce consistent, high-reliability boards. JLCPCB is an optimal choice for our group because it is a company that has consistency in creating reliable, manufacturable, and high-performance circuit boards that meet all of the PCB standards discussed above.

In addition to these standards, there are also general physical and dimensional constraints that guide our PCB design. JLCPCB imposes minimum and maximum board size limits, minimum trace width and spacing, which is typically around 6 mil, and hole size requirements that we must meet for successful fabrication. These constraints do influence how densely we can place components and how we route connections, especially in limited board real estate. Drill-to-copper distances also play a role in ensuring that the board meets mechanical and manufacturing tolerances. Proper silkscreen labeling, mounting hole placement, and thermal relief strategies need to be incorporated to make the board both readable and robust, and most importantly adhering to manufacturability standards.

4.1.1 Soldering Standards

Proper soldering in assembling both PCB's and general electrical work is labeled by established industries. A key standard in this area is IPC J-STD-001, which list the Requirements for Soldered Electrical and Electronic Assemblies, and this sets comprehensive guidelines for achieving reliable solder joints. It outlines the proper soldering processes, acceptable joint appearance, cleanliness levels, and inspection criteria. For wire and cable assemblies, IPC/WHMA-A-620, has detailed criteria for proper soldering of wire terminations, insulation stripping, and mechanical integrity, ensuring that connections remain durable. Additionally, to protect sensitive components during the soldering process, ANSI/ESD S20.20 offers requirements for electrostatic discharge (ESD) control in manufacturing environments.

4.1.2 IEEE Standards

IEEE (Institute of Electrical and Electronics Engineers) standards are a globally recognized standard of regulation for electrical and electronic systems. These standards cover many industries, however for our project we will be focusing on hardware design, power systems, signal processing embedded systems. By adhering to IEEE standards, our project is benefitting from reducing the risk of malfunctions and improving its long-term performance.

IEEE 1451, which standardizes the smart sensors and transducers, is very relevant to our project's laser sensor system as it gives protocols for sensor communication. This ensures that the motion detection data is consistently being transmitted to the microcontroller. Once the sensors detect movement, the data must be processed, and it makes this standard essential for floating-point arithmetic in digital signal processing. IEEE 1149.1 standard provides a set structure for diagnosing faults within digital components. Through this, it ensures that important hardware components, such as the microcontroller, are consistently being monitored and working as expected. In addition to this, IEEE 1687 (IJTAG) facilitates the testing of the embedded components within the PCB for accurate mentoring and detecting specific faults,1241 defines standards for ADC's, ensuring that sensor signals are accurately converted into digital data for processing. This standard helps maintain precision in detecting and interpreting interactions with the lasers, which is better overall responsiveness. Lastly, IEEE

1620 ensures safe and efficient power conversion within the PCB. This standard helps regulate voltage levels and maintain power integrity, preventing fluctuations that could impact the microcontroller performance and sensor accuracy.

4.1.2.1 IEEE Validations Standards

IEEE validation standards detail structured methods for testing hardware and software components in engineering systems. One of the key standards in this area is IEEE 829 which focused on test documentation for software validation, providing guidelines for test plans, test design specifications, and test summary reports. Additionally, IEEE 12207**,** defines processes for software lifecycle management, including validation activities, and is widely used in systems engineering. These validation standards play a crucial role in IEEE standards to make sure that all systems are meeting functional and performance expectations.

4.1.3 Safety Standards

Safety is a critical consideration for our senior design project, as there are a few hazardous components that our team must be mindful of. Aqua Strings involves the simultaneous use of water, electronics, and lasers, all of which pose potential dangers if our group is not being mindful. The integration of water and electrical components requires strict adherence to IEEE and IEC electrical safety standards. IEC 60529 is the standard for water resistance, which ensures proper insulation and protection against short circuits. Encasing electrical components in waterproof housings and using sealed connectors will be essential to prevent water from coming into contact with the PCB, microcontroller, and power supply. Additionally, proper grounding and the use of low-voltage DC power sources will minimize the risk of electrical shock. Circuit protection mechanisms, such as fuses and surge protectors, will further enhance safety by preventing overcurrent conditions that could damage components or create fire hazards.

Beyond electrical safety, the project must also account for laser safety, as the system utilizes lasers to detect motion within the water strings. Following the laser safety standard, IEEE 60825-1, will ensure that the chosen laser modules operate within safe power limits, reducing the risk of a potential eye injury. The lasers used will be Class 1 or Class 2, which are considered safe under normal operation and do not pose significant harm unless directly stared into for extended periods. Additionally, proper alignment and shielding will prevent accidental exposure to laser beams, ensuring that they are directed only down at the intended point of the bottom of the harp. Clearly marking laser areas and providing safety instructions for users interacting with the system will further mitigate risks.

4.1.4 IEC Standards

In the context of power supply systems, IEC standards provide essential guidelines for both design and operation. One of the key standards is IEC 60950-1 which outlines safety requirements for information technology and communication equipment, including power supplies. These standard address protection against electrical shock, fire hazards, and mechanical dangers. IEC 61010-1 is also relevant, particularly for power supplies used in laboratory and industrial control equipment, specifying safety requirements related to insulation, grounding, and overvoltage protection. For power supply efficiency and performance, IEC 62040 applies to uninterruptible power systems, also known as UPS, which detail classifications and testing procedures to ensure proper backup functionality and energy performance. Furthermore, IEC 61000 series standards are crucial for ensuring electromagnetic compatibility, and help power supplies operate without emitting excessive electromagnetic interference or being disrupted by external sources.

4.1.5 Audio Standards

Audio standards from both electrical performance and ear safety perspectives are essential in guiding the development, testing, and usage of audio equipment to ensure both optimal functionality and user protection. From an electrical standpoint, audio standards define parameters such as signal voltage levels, impedance matching, distortion limits, and frequency response to maintain consistency across devices and ensure compatibility. One of the most recognized organizations in this field is the Audio Engineering Society (AES), which has developed numerous standards, such as AES48, which addresses grounding and shielding practices to minimize noise and interference in audio systems, and AES17, which specifies methods for measuring audio equipment performance, including total harmonic distortion, noise levels, and dynamic range. These standards ensure that audio equipment can operate reliably and with high fidelity across various applications, from professional studios to consumer electronics.

In addition to performance standards, there are critical guidelines that address the safe use of audio equipment to prevent hearing damage. Organizations like the Occupational Safety and Health Administration and the World Health Organization have established exposure limits for sound levels, particularly in occupational and recreational environments. WHO recommends lower thresholds for prolonged listening, especially through personal audio devices. Standards such as IEC 62368-1, which covers safety requirements for audio, video, and ICT equipment, include guidelines on limiting audio output to prevent accidental hearing damage. Similarly, EN 50332, a European standard, specifically sets testing and safety requirements for headphones and personal music players, mandating that the output should not exceed safe listening levels unless explicitly overridden by the user with warnings in place.

Moreover, emerging technologies have prompted updates in standards to address modern concerns. For example, wireless audio transmission standards like

Bluetooth include specifications for latency, compression, and volume control that indirectly affect both audio quality and user safety. As audio devices increasingly integrate with smart systems, there is growing attention to ensuring that automated volume adjustments, noise cancellation, and environmental awareness features adhere to safety norms. Together, these audio standards form a comprehensive regulatory framework that supports high-performance audio experiences while prioritizing health.

4.1.6 ANSI/H1 Submersible Water Pump Standards

Submersible Water Pump Standards focus on providing guidelines and requirements for the design, construction, and performance of submersible pumps, particularly those involving the pumping of water or other fluids in submerged environments. These standards are crucial for ensuring that these pumps operate well under challenging conditions such as high-water pressures, electrical submersion, and potential exposure to contaminants. The ANSI (American National Standards Institute) and H1 refer to specific regulations that dictate mechanical, electrical, and performance specifications that manufacturers must follow to ensure both functionality and safety.

One of the most important aspects of the ANSI/H1 standards is the safety requirements, which cover both electrical and mechanical components to prevent hazards like electrical shock, fire, and damage from submersion. The standards outline requirements for electrical insulation, motor protection, and temperature control to ensure that pumps can operate safely underwater for extended periods. Additionally, the materials used for construction, such as corrosion-resistant metals and seals, are specified to ensure that pumps withstand the harsh environmental conditions they are exposed to in submersible applications.

Performance criteria under the ANSI/H1 standards typically include guidelines on pump efficiency, capacity, and durability. These specifications ensure that submersible water pumps can handle the required flow rates, head pressures, and operational durations without failure. Furthermore, the standards address factors like noise level restrictions and vibration limits to enhance operational longevity and reduce environmental disruption. Additionally, ANSI/H1 standards may also require rigorous testing methods, including performance testing under simulated operational conditions, to ensure that pumps meet the established criteria for reliability and efficiency.

4.2 Constraints

4.2.1 Time Constraint

The largest constraint that our team has to consider with our senior design project is the limited time. Our Senior Design semester takes place in the Spring and Summer term, and our group only has twelve weeks to bring our water harp fountain concept to run successfully and ready for demo. Unlike a traditional timeline with 14 weeks for Senior Design 2, this compressed schedule offers no

possibility for extensions and has a shorter fixed deadline that we must meet to complete the project and pass the class.

Our team has already taken a few proactive steps in the beginning of the semester by forming our group, creating a project design, and gathering research. These early actions and overall accountability from each member will help reduce some of the time pressure as the semester moves along. Our project involves multiple complex and time-consuming tasks, including the design of electronics and water reservoirs, setting up motion sensors, and integrating sound output when the water is touched. Each of these tasks presents its own challenges and building the water reservoirs requires careful planning and thought to ensure water flow and safety. Sensor testing and calibration will take work and time to make sure the system accurately detects interactions, as well as the sound system needed to be properly integrated. These elements all need thorough testing, debugging, and troubleshooting, that are all heavily taken into consideration with the timeframe.

Some unexpected problems such as component shipping, testing failures, or overall delays in trouble shooting efforts could complicate our planned schedule. A specific concern is the potential delay in receiving critical components, like the PCB, due to supply chain issues or shipping delays. If parts arrive late, it could slow down our progress on certain project aspects, and this would consume valuable time that we cannot afford to lose. The risk of these time constraints is that our project might become rushed, leading to compromises in quality or overall success in the course. Our group will continue to be proactive in organization, time management, and communication to overcome the time constraint of our project.

4.2.2 Economic Constraint

Our project operates under an economic constraint due to the lack of an external sponsorship or funding. Unlike some projects that receive financial backing from companies or university grants, our group is responsible for covering all costs for our design. In order to manage this, each group member has agreed to split the total estimated cost of around $400 equally among the four of us. While this approach ensures a fair and equal distribution of expenses, it also means that all of us are aware of personal financial limitations when it comes to purchasing items. This makes financial decisions a priority throughout our Senior Design project.

One of the key challenges we face is balancing cost efficiency with the need for backup materials. While it is important to minimize any unnecessary expenses, we also recognize that there are risks associated with having no extra components in case of unplanned damage or failure at any point of the semester. However, since we are working with limited funding, we must carefully determine which materials are essential to purchase in surplus and which can be limited to the exact quantity needed. Our group has held multiple discussions on our financial plan with how to optimize our budget while maintaining enough flexibility to account for unforeseen issues.

Additionally, our budget impacts our ability to test as freely. Ideally, extensive testing with multiple components and materials would allow us to refine our design and troubleshoot issues more effectively, but this would mean additional material costs. Since we want to avoid unnecessary purchases, we must be selective in our testing approach, prioritizing the most critical aspects of our design while minimizing unused materials. This economic limitation pushes us to be more precise in our initial design and creating a more budget-friendly project.

Furthermore, rising costs due to inflation and international tariffs have added an extra layer of complexity to our budgeting process. Since many of our components are sourced from overseas, especially from China, unexpected increases in part prices or shipping fees could strain our already limited budget. Tariffs and fluctuating exchange rates can significantly raise the final cost of key components, making it harder to stay within our $400 target. This forces us to closely monitor pricing trends and potentially delay or revise orders based on market changes, which introduces uncertainty in both our timeline and budget forecasting.

4.2.3 Environmental and Sustainability Constraints

Several environmental and sustainability constraints were considered within our hardware. The selection of components such as the analog-to-digital converter, amplifier, and photodiodes emphasized low power consumption to minimize energy usage during operation. The PCB design aimed to reduce material waste by optimizing trace routing and board size. Where possible, components were chosen from manufacturers with a restriction of hazardous substances to limit the use of environmentally harmful materials like lead and mercury. Additionally, considerations are being made with disposal, ensuring that the system can be disassembled and recycled with minimal environmental impact. The design also seeks to minimize electromagnetic interference and thermal output, reducing its ecological footprint in both active and passive states.

4.2.4 Political Constraints

One of the primary political constraints affecting our project is the ongoing trade relationship between the United States and China. Since most of our components are being sourced from Chinese manufacturers, our budget and timeline are potentially impacted by import tariffs. These tariffs, which have fluctuated due to geopolitical tensions and trade negotiations, can increase the overall cost of materials beyond initial projections. Additionally, customs delays or stricter import regulations may affect shipping timelines, risking disruptions in the prototyping and testing phases of our project.

Beyond tariffs, there are broader political risks associated with sourcing key electronic components from abroad. Export controls, bans on specific Chinese suppliers, and supply chain restrictions could limit our access to critical parts or require last-minute substitutions. These issues also raise concerns about long term availability and compatibility, particularly if future policy changes restrict access to

certain technologies. As a result, these ever-changing political dynamics have led our group to consider identifying alternate suppliers or domestic alternatives whenever possible.

4.3.5 Safety Constraints

Given that our project involves the interaction of electrical components and water, safety is a critical consideration throughout the entire design and construction process. The core concept of the water harp includes a reservoir or flow of water near active electronics, including a speaker, amplifier, PCB, and microcontroller. To reduce the risk of short circuits, corrosion, or electric shock, we are taking precautions such as waterproofing exposed components, isolating high-voltage elements from water zones, and using non-conductive housing materials in splash-prone areas. Ground fault protection and secure insulation will be implemented where appropriate, and the design ensures that no users come into direct contact with live circuits, even in the event of a water leak.

Another key concern is the integration of lasers and photodiodes for sensing interactions. Since the system uses visible lasers to simulate strings, we must comply with laser safety standards to protect both users and team members from potential eye damage. Only low-power, Class I or II lasers will be used, and laser beams will be aimed in a fixed, downward, or controlled direction to minimize accidental eye exposure. Proper labeling and safety instructions will also be part of the final display if the project is demonstrated publicly. In addition, all materials used in the construction of the harp's frame must be structurally stable and free of sharp edges or splintering surfaces. We will avoid toxic adhesives or finishes and ensure the frame can safely support all mounted components without tipping or collapsing.

The assembly and testing process also involve several safety constraints. Soldering components onto the PCB requires proper ventilation, eye protection, and heat-safe work areas to prevent burns or inhalation of fumes. We will follow standard ESD (electrostatic discharge) safety practices when handling sensitive components. The water pump, while low-voltage, introduces mechanical and electrical risks as well—improper pressure or leaks could create unsafe conditions, so we'll ensure proper sealing and containment during operation. Finally, since the project will be handled by both the design team and potentially demoed to others, user safety is a top priority. The system will be designed to operate safely and predictably, even if interacted with by someone unfamiliar with its inner workings.

4.3.6 Manufacturability Constraints

Our project includes the design and fabrication of a custom printed circuit board (PCB), which introduces several manufacturability constraints. We are utilizing the University of Central Florida's JCB (Joint College of Business and Engineering Maker Space) resources for PCB milling, which limits us to certain board sizes, routing capabilities, and the types of materials that can be processed in-house.

Unlike outsourced fabrication services, the JCB tools have specific constraints on minimum trace width, spacing, and layer count—typically supporting only single- or double-layer boards. This restricts the complexity of our circuit layout and requires careful planning to avoid routing conflicts or overcrowding components. Additionally, since in-house milling does not support plated through-holes, we must manually solder both sides or adjust our layout to avoid via reliance.

We are also using Fusion 360 for modeling and designing the physical frame and housing of the project, which presents its own set of constraints. While Fusion 360 is a powerful tool for CAD and simulation, the manufacturability of the designs it produces depends on access to tools such as 3D printers, laser cutters, or CNC machines available at UCF's fabrication labs. This means that certain design choices must account for machine tolerances, material availability, and structural simplicity to ensure parts can be built efficiently and reliably. Designs must be constrained by the build volume of 3D printers and the types of filaments or materials stocked at the lab. All components must also be designed for ease of assembly and integration with the electronics, keeping in mind the alignment, mounting, and potential need for access to internal parts during troubleshooting.

Chapter 5: Comparison of ChatGPT with other Similar Platforms

Artificial Intelligence (AI) is becoming a useful tool in engineering design, offering instant access to technical information, assistance in creative problem-solving, and computer-aided support for coding through documentation of design. Large Language Models (LLMs) such as ChatGPT, Gemini, Claude, and DeepSeek are the latest generation of AI platforms that can comprehend natural language inputs and respond with contextually relevant, technically correct output. Their use in engineering applications is transforming the way students, researchers, and experts resolve complex design problems.

In the instance of the Aqua Strings Interactive Waterfall Display, a senior design project that is cross-disciplinary with hardware, embedded systems, fluid dynamics, and user interface AI platforms were important as an enabler of accelerated development and improved decision-making. These platforms were used in supporting software development, sensor calibration, component selection, and documentation composition. Besides individual use, AI assistants also increasingly become collaboration tools, supporting team communication, idea organization, and iteration design.

This chapter explores the technical potential, benefits, and limitations of four leading AI platforms used in engineering design, namely embedded systems development, and collaborative learning application. By comparing their performance, integration capabilities, and real use in the Aqua Strings project, we create a framework for selecting the most appropriate AI tools for future engineering tasks.

5.1 Introduction to AI Platforms in Engineering Design

In the rapidly evolving field of engineering design, Large Language Model platforms have been a goldmine of tools for conceptualization, research, and problem-solving. In the creation of the Aqua Strings interactive waterfall display, our team made use of several LLM platforms to enhance our design process. This chapter explores the relative strengths and weaknesses of ChatGPT, Claude, Gemini, and Deep Seek, focusing on their specific applications in engineering design uses.

GPT-o1 was created released by OpenAI on December 5th, 2024 as a first in their new o series which was an advancement on OpenAI's previous model GPT-4 this newer model represented a shift for OpenAI in implementing their new chain of thought reasoning which allowed them to solve more complex problems by using a reinforcement learning structure to help refine and recognize its mistakes to give a better response. You can access this model through either the OpenAI API or through the ChatGPT web or mobile app. Currently GPT-o1 and GPT-4 are only available to ChatGPT pro users while free members have access to GPT-3.

Claude 3.7 Sonnet was released by Anthropic in February 2025, as a successor to Anthropic's previous 3 model Claude 3.5 Haiku. Sonnet introduces enhanced reasoning through an extended reasoning mode. This model can be accessed through the Claude mobile or web app along with an API. Only Pro members have access to Claude 3.7 with free members only having access to Claude 3.5 Haiku.

Gemini 2.5 was released by Google on March 12th, 2025, as a successor to the previous Gemini 1.5 and is Google's currently most advanced model featuring a chain of thought methods like Claudes extended thinking and OpenAI's feature by the same name. Gemini also includes some other features such as a temperature slider to determine how much creativity you want to give the LLM along with code execution allowing the LLM to generate code and run code to solve problems and access to google search to gather up to date reference information. Gemini 2.5 pro has an experimental version available for free using google AI studio along with an available API.

DeepSeek R1 was created by the Hangzhou DeepSeek Artificial Intelligence Basic Technology Research Company and was released On January 2025. Unlike the other models, DeepSeek is unique in that it is open source. DeepSeek features include the standard of being able to input both texts pictures and documents and features a deepthink mode available for free on the DeepSeek website. DeepSeek has seen major criticism

5.2 Comparative Analysis of AI Platforms

This section introduces the four AI platforms evaluated for their relevance to engineering design, embedded systems development, and educational collaboration: ChatGPT, Gemini, Claude, and DeepSeek. Each platform represents a different company's philosophy on how a large language model should be deployed, accessed, and optimized for technical problem-solving.

Table 5.2.1: *Comparison of LLMs*

Model	Intelligence Index	Speed (Tokens/sec)	Price (per 1M Tokens)	Context Window	Knowledge Cutoff	Special Features
ChatGPT o1	62	70	$26.3	200,000	October 2023	Multimodal supports audio and vision, built in code interpreter, Web access, DALLE image generation
DeepSeek R1	60	23	$1	128,000	July 2024	Open Source, Good Reasoning, uses reinforcement learning, efficient
Gemini 2.5 pro	68	206	$3.4	1,000,000	January 2025	relatively high context, code interpreter, high rate limit
Claude 3.7 Sonnet	48	78	$6	200,000	October 2024	chart engine, recent knowledge cutoff, code interpreter

ChatGPT (OpenAI): Based on OpenAI's GPT-4 architecture, ChatGPT is widely recognized for its balanced performance in code generation, technical reasoning, and explanatory clarity (OpenAI, 2024; Wired, 2024). It is available through a web interface and API, with both consumer and enterprise-grade offerings. ChatGPT is commonly used in engineering for tasks such as debugging, firmware development, and documentation assistance (Khattar et al., 2023; NASA, 2023). It features strong language modeling and an expanding plugin ecosystem, although it relies on cloud access.

Gemini (Google): Gemini is Google DeepMind’s flagship multimodal AI model, designed for reasoning across text, code, and visual data. Integrated into Google’s ecosystem through tools like Vertex AI and AI Studio, Gemini stands out for its massive context window up to one million tokens and its ability to analyze diagrams, codebases, and even sensor outputs in a single session (Google DeepMind, 2025; Blog.Google, 2025; AI.Google.Dev, 2025; Omdia, 2025). This makes it especially useful for large, complex engineering projects requiring simultaneous analysis of documentation, code, and schematics.

Claude (Anthropic): Claude emphasizes interpretability, ethical design, and long-context reasoning, with context limits up to 200,000 tokens (Anthropic, 2025; CoinReporter, 2025). Its strength lies in structured responses, technical accuracy, and collaborative features, such as its seamless Slack integration. Claude is designed to act like a knowledgeable and safety-aware teammate, making it ideal for summarizing technical documents, offering design advice, and supporting group discussions during engineering development (Anthropics Interpretability Research, 2025).

DeepSeek: Developed as an open-source alternative to proprietary models, DeepSeek offers advanced reasoning and coding capabilities while allowing full offline deployment through applications such as LM Studio and Ollama (). It is highly customizable and ideal for privacy-conscious or resource-constrained environments. With models like DeepSeek-Coder and DeepSeek-R1, it supports efficient, low-cost integration into local workflows, including embedded systems development, without reliance on external servers (Omdia, 2025).

Case Study 1: "What type of sensor should be used for accurate and waterproof hand detection through a laminar water stream?"

In deciding the most suitable type of sensor to use for hand interruption detection by using a laminar water stream under the Aqua Strings project, some AI models including ChatGPT 4o, Claude 3.7, DeepSeek R1, and Gemini 2.5 Pro were considered to compare how they would work in determining component selection. A sensor that would be waterproof as well as provide accurate readings was sought, together with a sensor that could give good performance where lighting is undergoing changes.

ChatGPT 4o highly advocated the utilization of a modulated infrared (IR) through-beam sensor, including a 940nm IR laser diode and receiver, either a phototransistor or TSOP-type receiver. It contended that laminar water, being optically transparent, permits IR beams to pass through with little interference. When interrupted by a hand, the system is sure to detect the signal loss. 38kHz modulation adoption, as well as waterproof acrylic enclosures, were also suggested for false triggering evasion due to indoor or ambient sunlight. ChatGPT rationalized its recommendation by explaining why the other sensors fail, as for instance, ultrasonic sensors produce variable readings due to water flow noise and reflections, and capacitive sensors produce false positives when they meet water.

Claude 3.7 offered a different perspective by choosing waterproof capacitive proximity sensors, pointing out that they are able to detect objects without contact. It further noted that these sensors can function well even if housed in a casing of plastic, and that they are water-resistant in design. But Claude did recognize that the sensors would have to be properly adjusted so that water and a human hand could be differentiated, and this would have to be done in a manner inconsistent

with flow conditions. Its conclusions were across-the-board and were primarily composed of web search results, not naming vendors for off-site sensors but providing minimal concrete circuit-level implementation guidance.

Both DeepSeek R1 and Gemini 2.5 Pro independently came to the same conclusion as ChatGPT 4o, that the optimal solution was modulated IR through-beam sensors. DeepSeek outlined the way in which this solution fulfilled all the significant requirements: waterproofing (via IP67/IP68 housing), immunity to ambient light (via modulation), and proper detection of hand interruption. Gemini went further by evaluating a few optical and ultrasonic setups. It ruled out ultrasonic sensors due to the irregularity of sound waves along the water–air interface and criticized capacitive sensors for excessive false triggering under damp conditions. Gemini discovered that an IR beam-break arrangement provides maximum accuracy and environmental resilience, especially since near-infrared light propagates well in laminar water, and it is appropriate for Aqua Strings' interactive performing aims.

Overall, while Claude 3.7 provided a competitive alternative and DeepSeek R1 and Gemini 2.5 guaranteed the method works, ChatGPT 4o exceeded by providing not only the best component selection but also an implementation work plan for the method, including modulation frequency, lens casing, and ESP32-compatible circuit design. The engineering realizability and level of detail suggested from ChatGPT to be most feasible and complies with Aqua Strings project requirements.

Case Study 2: "How can we implement real-time audio mixing to overlay two or more WAV files on an ESP32 from an SD card with minimal latency?"

In seeking how to conduct real-time mixing of audio for two or more WAV files on an ESP32 from an SD card with low latency, several language models were tested for technical insight, clarity in implementation, and ability to fulfill performance expectations with practical viability. The core issue was enabling the capability to combine and play several audio streams in real-time without audio degradation or lag critical to the Aqua Strings instrument, in which multiple "strings" may be triggered rapidly in succession.

ChatGPT 4o provided the most straightforward yet complete response, outlining a clean architectural order for real-time playback of audio. It suggested using the ESP32's native I2S peripheral for low-latency audio output, along with ring buffers per audio track and FreeRTOS tasks to read files and mix samples. Mixing logic was described as simply mixed_sample = (sample1 + sample2) / 2, requiring sample values to be clamped to audio limits (such as -32768 to 32767 for 16-bit signed PCM). This answer was significant for balancing feasibility and clarity, using examples that are easy to realize. The model also proposed double buffering and DMA to reduce lag and went as far as to propose a stretch goal: dynamically adjusting playback volume and length based on user interaction speed precisely

in accordance with Aqua Strings' stretch goal of "harder strums play louder or longer".

Claude 3.7 repeated the same hardware-level strategies but went about them with a more software-oriented mind. It gave a lot of weight to buffer size tuning, recommending small 1024-sample buffers to reduce latency, and included multi-core processing by assigning the mixer task to one ESP32 core and leaving the other available for various Arduino operations. Claude's implementation plan was good but lacking in some low-level code readability or performance optimization recommendations for SD card I/O, which is a typical hotspot in embedded audio systems.

DeepSeek R1 had a very structured and detailed plan of implementation, dividing the system into entities like I2S audio output, SD card double buffering, and real-time mixing using FreeRTOS. It proposed concepts like contiguous storage of files, buffer underrun detection, and summation in 32-bit downscaling to 16-bit to avoid overflow during mixing. DeepSeek was distinctive in that it made a high-level split of two FreeRTOS tasks, one for Core 0 for SD file reads and another for Core 1 for mixing and I2S streaming. It even showed good C++ code with interrupt-safe buffer handling and task scheduling. This made DeepSeek's answer particularly appealing to engineering teams that need precise timing control and high-grained resource allocation

Gemini 2.5 Pro provided the most comprehensive system-level design. It emphasized synchronization using semaphores, utilization of double mix buffers, and comprehensive WAV header parsing to enforce consistent sample rates, bit depths, and channel formats on all input files. Gemini also discussed the topic of clipping versus attenuation, explaining how to avoid distortion when summing audio streams with integer scaling or floating-point arithmetic. Its level of detail such as DMA underrun handling, buffer alignment, and SD card optimization using SDMMC matched reading professional-level embedded audio manuals.

In short, all models provided helpful insights, but ChatGPT 4o excelled at providing a fast, practical solution with extensibility for Aqua Strings' creative goals, while DeepSeek and Gemini provided advanced engineering solutions for groups that were testing the boundaries of the hardware's performance. Claude provided a good middle ground but lacked some of the low-level optimization details necessary for time-critical audio performance. Together, these AI-based responses formed the basis for both the MVP release and the leading-edge planning of low-latency audio playback using the Aqua Strings instrument.

Case Study 3: "What are the trade-offs between using mineral oil vs. water for a liquid-based instrument that uses lasers and photodiodes?"

In weighing the merits of using mineral oil as opposed to water as the fluid medium for a laser-photodiode device like Aqua Strings, a number of AI models laid out

detailed comparisons on optical, thermal, safety, and maintenance-related factors. The goal was to determine which fluid is more appropriate to clear beam transmission in a laminar stream, minimal interference, and long-term resistance in interactive environments.

ChatGPT 4o began with a brief, tabular contrast on the basis of optical characteristics like refractive index and scattering, as well as practical factors like viscosity, evaporation rate, and material compatibility. It explained that water has a lower refractive index (~1.33) and minimal distortion of light and is therefore appropriate for laser alignment and visibility in the visible spectrum. Yet, the lower viscosity of water can contribute to a more turbulent flow and higher evaporation, necessitating frequent refilling and maintenance. Mineral oil, on the other hand, provides smoother, more stable laminar flow because of its higher viscosity and zero evaporation, which is preferable for stable beam paths. Yet, it is more expensive, more difficult to clean up, and may be incompatible with certain plastics.

Claude 3.7 focused more on thermal and optical characteristics, quoting that mineral oil is clearer in the infrared spectrum and has a higher refractive index (~1.47), which can be beneficial for optical design and index matching. It also mentioned that mineral oil is thermally more stable and less prone to microbial growth. Water was preferred, though, because it has a higher cooling capability and is non-flammable. Claude noted that water is less expensive and safer but would need to be treated with biocides or frequently changed to avoid biofouling, especially in open systems such as Aqua Strings.

DeepSeek R1 gave a comprehensive technical review, citing specific heat, flow efficiency, dielectric strength, and chemical reactivity characteristics. It briefly described why water is more suitable for thermal management due to its high heat capacity and low viscosity for faster flow and heat removal. However, the electrical conductivity of water is hazardous in the presence of electronic equipment unless caution sealing is implemented. Mineral oil, being non-conductive, provides electrical safety and eliminates corrosion problems but has a higher viscosity that necessitates more pump power and can lead to pressure-induced difficulties with small nozzles. Further, DeepSeek raised the issue of oil oxidation and long-term filtering to ensure optical clarity.

Gemini 2.5 Pro presented the broadest comparison, splitting the choice into ten detailed engineering categories like refractive stability with temperature, Fresnel reflection losses, biological contamination, and environmental disposal. It argued that water is best where cost, ease of handling, and clarity in visible light are the major concerns, especially if the temperature conditions are moderate and routine maintenance can be accepted. For sealed systems, however, it recommended mineral oil where electrical insulation and long-term stability outweigh cost and cleanup difficulties. Gemini concluded with a comparative summary matrix that helped to distill the trade-off logic into applicable selection metrics for real-world engineering decisions.

In all, while all models revealed simple trade-offs, ChatGPT 4o provided the most balanced summary, which is suitable for initial prototyping and decision-making. Claude and DeepSeek provided insight into system safety, thermal performance, and fluid dynamics more profoundly, while Gemini provided a very structured analysis for final system configuration. Cumulatively, the answers guided a design strategy in which water is favored for initial testing and visible laser interaction, with mineral oil as a viable alternative for commercial versions of Aqua Strings requiring less maintenance and more stable optics.

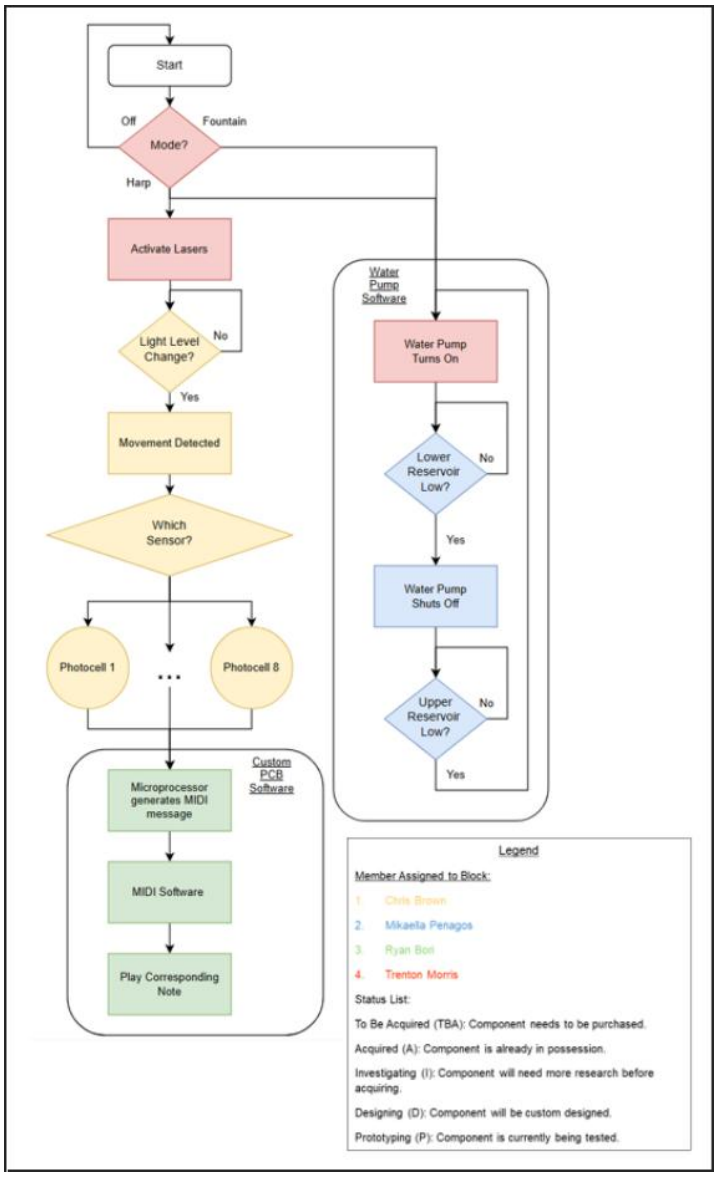

Figure 5.2.2: *Example Software Diagram*

Case Study 4: "Based on the software block diagram, what happens when the user interrupts a laser beam in HARP mode?"

In Case Study 4, the intention was to check how well different AI models were able to read a software block diagram of the Aqua Strings system in HARP mode specifically to observe what happens when a user interrupts a laser beam. The block diagram specifies the basic logic: the lasers are activated when HARP mode is active, and the smallest change in light level, say, a hand passing through the beam is followed by a sequence of detection and sound generation processes. This experiment measured how accurately the models could infer logic flow and describe system behavior from a drawn programming diagram.

ChatGPT 4o demonstrated a clear and accurate reading of the diagram's flow. It explained how when a user breaks a laser beam, the system first detects a light level change, then motion, and finally determines which photocell sensor (1 to 8) was triggered. From there, the microprocessor outputs a corresponding MIDI message, which the MIDI software decodes to sound the corresponding musical note. The solution correctly divided the left-hand music interaction logic from the right-hand water pump software branch, claiming the water logic to be independent apart from important reservoir conditions. This proved to be good diagram reading and proper domain interpretation.

Claude 3.7 also produced an equally accurate list of activities, listing all decision points and action steps involved—from laser firing to playing back MIDI notes. But it occasionally paraphrased block names or broke decisions up higher-level like "movement detection" and "sensor check," which, although correct, were not as technically accurate as those of ChatGPT. But Claude correctly identified that each beam is mapped to a distinct photocell sensor, and that this mapping triggers a MIDI signal to the audio playback system.

DeepSeek R1 and Gemini 2.5 Pro both performed well, correctly identifying the beam-break detection logic and the MIDI output pathway. Gemini’s explanation was particularly methodical, matching each diagram block with its functional outcome. It also pointed out that the interface of the laser harp employs beam interruptions as input signals, with each laser being treated as a "string." Gemini also referred to the autonomy of the water pump subsystem, and how it is only called upon under certain reservoir conditions. DeepSeek's reply was concise but emphasized the transition from sensor feedback to note generation, describing it as a "gesture-based MIDI control system" emulating physical harp strings, emphasizing the music metaphor of the instrument.

Among all the models, ChatGPT 4o gave the most comprehensive and well-defined explanation with the addition of both hardware components (like lasers and photocells) and software activities (like generation of MIDI, playing of notes). Along with this, ChatGPT 4o observed the importance of mode selection step as the system startup of the flowchart in determining whether it is to act as a harp or going to fountain mode. This level of accuracy depicts the significance of using AI to

elucidate and document embedded systems diagrams for technical writing or instructional writing.

5.3 Integration with Engineering Workflow

AI platforms are quickly becoming indispensable tools in engineering design workflows, particularly for firmware, embedded systems, and hardware-software co-design. All the platforms discussed here, ChatGPT, Gemini, Claude, and DeepSeek, have their respective strengths and limitations in offering support for real-world integration among these processes.

ChatGPT is typically employed to offer support for firmware and embedded development due to its reliable code generation as well as solid training on technical documents. It was employed to write microcontroller code, understand compiler errors, and propose architectural changes in real-time (OpenAI, 2024; Khattar et al., 2023; NASA, 2023). To illustrate, for Aqua Strings prototyping, ChatGPT was utilized to describe the logic of sensing laser beam interruptions and controlling note playing on the ESP32 microcontroller. In terms of integration, ChatGPT can be consumed through a secure API and implemented within build pipelines or testing frameworks. Its integration with GitHub Copilot and other IDEs allows it to assist engineers within their IDEs, reducing development friction. However, because it runs in the cloud, ChatGPT requires an internet connection and is not suitable for offline or field deployments.

Gemini, Google's cutting-edge LLM, provides high functionality for multimodal data integration, which proves useful in embedded projects where schematics, sensor plots, or visual data analysis is required alongside code. Gemini's big context window (1 million tokens) enables it to process and reason about whole firmware projects or long technical logs in one session. For Aqua Strings, Gemini could be used to analyze both schematic and the related microcontroller firmware to identify wiring or logic errors, giving a full debugging experience (Google DeepMind, 2025; AI.Google.Dev, 2025). While Gemini excels at handling complex technical workflows, it is tightly integrated with Google Cloud services. Meaning Gemini teams must have experience with cloud-based infrastructure Google's cloud services.

Claude excels in collaborative workflows and long-context design reviews. With 100k–200k tokens of context support, Claude can ingest full engineering specifications, firmware documentation, or versioned changelogs and assist in debugging or summarizing changes. For example, while calibrating Aqua Strings firmware, Claude could retain all the laser sensor calibration logic and progressively get better with more interactions without forgetting the context. Even though Claude lacks native IDE or plugins, Its API makes asynchronous integration into test frameworks or documentation pipelines for projects possible. Claude is less hardware-oriented and workflow-focused but skilled at software architecture, documentation, and collaborative decision-making processes.

DeepSeek stands out through the support of offline and self-hosted integration, which embedded systems operating securely or in isolated environments require.

Because DeepSeek-R1 and its iterations are open source, engineering teams can execute the model locally and integrate into bespoke toolchains. This is especially convenient in laboratory environments where hardware may not be connected to online networks. In Aqua Strings, for instance, a local version of DeepSeek could be used to generate code or perform real-time sensor log analysis during field tests without accessing external servers. DeepSeek also allows for fine-tuning, where a team can train a domain-specific AI helper that is tailored to their firmware layout, naming scheme, or hardware modules. Although its baseline performance will fall short of proprietary alternatives like GPT-4 or Gemini in raw capability, its adaptability and command of the data pipeline position it to thrive in deep integration into embedded development pipelines.

ChatGPT and Gemini are generally best integrated at the development level due to their ease of use and depth of developer communities. Claude maximizes collaborative workflows and is hence ideally suited to documentation-rich or team-based work. DeepSeek stands out for its offline-friendly and extensible architecture, making it suitable for private integration into workflows with embedded connectivity or limited data privacy. For the Aqua Strings project, a hybrid approach, using ChatGPT for firmware code generation, Claude for collaborative documentation, and DeepSeek for on-site support, offered the best-balanced method of integration.

5.4 Limitations and Potential Drawbacks

In contrast to the many benefits that AI platforms introduced to the Aqua Strings development process, a number of limitations were encountered when using these tools for real-world engineering workflows. Among the most recurring problems across all platforms was confident technical inaccuracy or oversimplification (Wired, 2024; ZDNet, 2023). For example, while Claude consistently provided well-structured and safety-conscious advice, it once suggested a broad "silicone spray coating" for underwater electronics, a very misguided remedy lacking the necessary inside protection and long-term reliability for underwater sensor housings. The same issues arose when platforms like ChatGPT or Gemini proposed components (voltage regulators or photodiodes) that were not matched to the voltage or current needs of the microcontroller, requiring manual cross-validation against datasheets. These incidents highlighted the importance of checking AI recommendations against engineering judgment and extrinsic documentation.

Another significant limitation was the context window size, particularly when working with large codebases or lengthy debugging discussions. Although Gemini’s expansive 1 million-token context capacity alleviated this issue to a degree, platforms such as Claude and ChatGPT (particularly the o1 model) struggled to maintain coherence over longer sessions involving firmware logic, hardware pin mappings, and real-time sensor feedback loops. This occasionally led the models to fixate on surface-level bugs or produce incomplete patches while missing more fundamental design flaws that had scrolled out of scope. This resulted in a higher likelihood of brittle or edge-case-prone solutions, with additional manual validation and test coverage needed.

Design fixation, a problem where the team tended to take the very first seemingly feasible AI-generated suggestion without considering other architectures or materials, was another challenge faced while executing the first project phases. This occurred, for example, with waterproofing system research, where initial proposals headed in one direction offered by ChatGPT before further testing revealed more modular and robust options. While AI tools accelerated concept development, they sometimes created a premature sense of finality, reducing the inclination for more thorough brainstorming unless prompted explicitly.

For DeepSeek R1, its primary weaknesses were not in model quality, but in usability and reliability. Although the open-source model had demonstrated good reasoning ability and excellent cost-efficiency, the hosted web interface consistently suffered from response timeouts, interface delays, and intermittent availability (DeepSeek, 2025; Theori, 2025). That made it unviable for time-critical use cases such as iterative code polishing or real-time debugging. While DeepSeek would have been more stable in a self-hosted environment where it could be integrated into a local toolchain and tuned for performance, this route was outside the scope of the present development workflow.

Finally, while AI tools accelerated many tasks such as documentation drafting, schematic planning, and firmware prototyping, they also introduced inefficiencies in the workflow in some cases. A few members of the team, in pursuit of the "ideal" prompt or reply, spent an excessive amount of time revising AI-inspired interactions instead of transitioning into physical prototyping or manual validation. This was the cause of minor delays in decision-making, particularly when AI responses had become the de facto standard for determining next steps. Blending AI support with engineering instinct and iterative testing remained essential for advancing the project.

5.5 Best Practices for AI Integration in Engineering Design

During the Aqua Strings project, our team gained insight into how AI platforms may be strategically employed in an engineering process. Our most significant best practice that we developed was strict verification of AI-generated outputs with authoritative sources like component datasheets, microcontroller technical reference manuals, and standard electrical specifications (Cheng et al., 2023; Vinuesa et al., 2020). While models like ChatGPT and Claude offered immediate circuit design recommendations or waterproofing strategies, they would occasionally simplify constraints into over-simplifications or give out-of-date or contextually irrelevant advice. Validation of parameters like input voltage ranges, tolerances, and interface compatibility through manufacturer documentation was still required to offer safe and functional designs.

We also found that prompt engineering was very important to the quality and accuracy of results. Prompt specificity that included direct constraints (e.g., "provide a pinout for ESP32-WROOM-32 with analog input mappings" or "compare MCP602 vs. TLV2372 for low-noise op-amp application") resulted in much higher-quality technical output. Gemini particularly did well when presented with well-

formatted questions comparing large datasheets or part specs cross-referencing, while ChatGPT did better when requested to reformat firmware code or explain low-level I2C protocols. This practice relieved much of the necessity for iterative clarification and allowed faster convergence to implementable solutions.

Another key finding was the advantage of preserving iterative, back-and-forth workflows rather than answering AI questions as one-time requests. Claude's long-term context retention enabled long-form debugging of firmware logic over several sessions, allowing optimizations in water flow control algorithms and coordination tasks on the ESP32 (Anthropic, 2025; Anthropics Interpretability Research, 2025). Gemini's multimodal and extended context capabilities further enabled simultaneous comparison of multiple datasheets or code snippets, which was invaluable during sensor selection and hardware-software interface design. This iterative use case aligned well with real-world engineering, where problems are rarely solved in a single step and often benefit from layered refinement.

Though these were positives, we always found AI tools to work best as support design tools rather than making autonomous decisions. They were most welcome in suggestions made along with proven engineering tools such as scholarly papers, field-proven guidebooks, or expert opinions, which provided depth and experiential knowledge AI models may not. For example, waterproofing was guaranteed in the case of AI-generated coatings by following up on the outputs with smarts from marine electronics standards to prevent premature failures in underwater conditions. This guaranteed that innovative ideas from AI were grounded in real-world reliability and safety.

To prevent inefficiencies, we imposed stringent usage limits on AI platforms within our workflow. Assigning tasks to individual models, like templating firmware using ChatGPT, writing documentation using Claude, interpreting data sheets using Gemini, and offline troubleshooting using DeepSeek, helped to limit tool overlap and avoid wasteful cycles. Without these restrictions, the responsiveness of these tools at times led to prolonged prompting sessions with diminishing returns, particularly where users were seeking perfection rather than functional sufficiency.

In conclusion, our experience confirms the application of LLMs as collaborative co-engineers, capable of generating ideas, accelerating research, and simplifying embedded system design. However, their potential is realized when integrated with human engineering expertise, continuous verification, and in-the-field validation. With the combination of AI speed and classical rigor, we achieved an innovative yet grounded workflow that provided better design options for Aqua Strings.

5.6 Conclusion

The use of AI platforms throughout the Aqua Strings development cycle transformed our engineering process in terms of concrete improvements in speed, creativity, and technical depth. Whether interpreting datasheets and debugging firmware, performing component trade-off analysis, or writing documentation, each platform realized its own unique technical strengths. ChatGPT's code explanations

with structured text and template generation accelerated firmware creation. Claude's long-context conversational memory made it a fit for iterative refining logic and long-document writing. Gemini’s ability to process extensive technical documents enabled cross-referencing tasks. DeepSeek, while less stable in hosted form, showed promise as an adaptable, low-cost, offline reasoning engine for embedded systems. Yet, the impact of AI was most evident when used in conjunction with traditional engineering principles. Our experience reaffirmed that no model can fully replace the need for hands-on validation, constraint-driven design, and critical cross-referencing with authoritative sources. Context window limitations, occasional hallucinations, and lack of real-world awareness indicated the need for cautious human oversight (Bubeck et al., 2023; Wired, 2024; Blog.Google, 2025). As a collaborative tool, not a standalone authority, AI systems expanded our design space and forced us to iterate and think creatively. In brief, successful use of AI for engineering isn't merely automation—it's amplification. These tools amplified our technical capacities and reduced friction in research, prototyping, and debugging. By balancing LLM support with human-centered design practice, the Aqua Strings project demonstrated how modern engineering teams can harness the potential of AI to innovate faster, validate more comprehensively, and solve complex problems with greater agility.

Chapter 6: Hardware Design

Hardware design plays a crucial role in the overall success of any engineering project because it involves the physical components and structures that make the system function as it is intended. This chapter of our project includes the organization and integration of all necessary parts, such as sensors, pumps, frames, and wiring, ensuring that each component works in harmony within the system. The hardware design must consider factors like durability, manufacturability, and compatibility with other parts of the project, all while meeting the performance and safety standards. The hardware design is also essential for enabling the system's functionality for the recirculation of water, the interaction between the laser sensors and water strings, and the safe operation of all electrical components. A well-thought-out hardware design achieves having all components are properly housed, protected, and positioned, and considers the constraints such as size, cost, and manufacturability. Ultimately, hardware design is key to ensuring that the system operates efficiently, reliably, and safely, contributing significantly to Aqua String’s overall success.

In developing the hardware design and mitigating the technical aspects of a project, it is paramount to precisely document the schematics and generate component selection lists that guarantee proper operational conditions for all subsystems. The custom PCB identified as the heart of the system serves as a central location for measuring, controlling measurements, and powering all the lasers as well as sensors. The central microcontroller is an ESP32 and uses GPIO and SPI protocols to manage all of the inputs and outputs of the proprietary hardware system including several laser diodes, photodiodes, and a microSD card reader. The laser modules are powered by the LM2576-5.0 switching regulator that

provides the regulated 5V rails. The 3.3V powered elements (i.e., ESP32, photodiodes, and microSD card module) are powered by another LM2576-3.3 power regulator. For reliability both regulators utilize a diode, inductor, and capacitor network to remove voltage ripples and improve their reliability. The circuit employs dedicated MOSFETs to turn the laser modules on and off, series resistors and capacitors to shape the audio signal to the PAM8403 amplifier, which drives the speaker. The amplifier is located on a removable sub-board so it can easily be replaced in the event of failure. Peripheral headers are connected to a laser receiver and transmitter boards, and careful complication of signal routing and ground planes are used to minimize electromagnetic interference and any other unwanted disturbances. The system layout allowed for the incorporation of plexiglass shielding from the environment and respect for a physical separation of the wet and dry zones to promote longevity and safety for sensitive electronics. Through a complete and well detailed schematic, we have ensured that every hardware element is integrated properly, resulting in having all hardware components operate efficiently and robustly.

6.1 Design Overview of Aqua Strings Hardware System

One of our initial designs consisted of including a ramp runoff with dividers that would separate the flowing water into eight different sections, one for each water string and laser beam. In the below figure, the design is laid out to show the side profile of the system and how the recirculation of the water would occur. These hardware components, including the laser beam transmitter and receiver are also displayed showing a portion of each component either being exposed to water or fully submerged in the water.

As seen by the figure below, the hardware components of Aqua Strings are mostly located within the top half of the frame. The only exception is the submersible water pump that will be located on the bottom left of the lower reservoir to pump the water back up for the closed loop system. At the top of the frame, there is a fully dry zone to hold the laser sensors, with the transmitters and receivers aligned across the water streams. This dry compartment also contains the PCB, where all sensor wiring is routed for centralized processing. Next to the PCB, the speaker is mounted within the same dry area, positioned in a way that is close to water sensitive electronics. The mode switch is located on the exterior of this compartment next to the speaker to once again provide safety and proximity to the PCB.

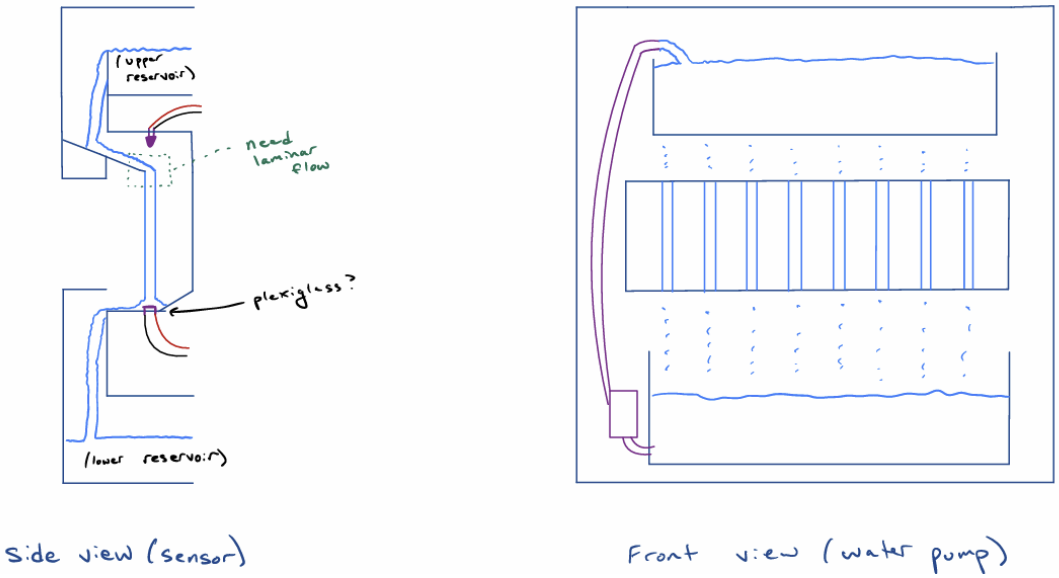

Figure 6.1.1: *Aqua String Side and Front Sketch*

The water pump is positioned at the bottom of the frame, fully submerged in the lower reservoir. A tube runs vertically along the structure, directing water from the lower reservoir back up to the upper reservoir. The ramp for water flow is situated beneath the dry compartment, guiding the water downward in distinct sections. Support beams on either side provide structural stability, maintaining separation between the wet and dry zones.

As this project progresses new adjustments are made altering the overall design of our project. Our most recent design is demonstrated in Figure 6.1.1 and Figure 6.1.2 where the flow of water is demonstrated. These designs may vary slightly from now on, but the flow and overall placement of components should relatively remain constant.

In Aqua Strings, the water is designed to flow smoothly and deliberately, starting from a standard PVC pipe or flexible tubing that takes the water from the top reservoir, through individual funnels above each water string, see Figure 6.1.2. To help eliminate turbulence and promote a stream that is clean and undistorted, each funnel has flow straighteners placed in with or before it. Flow straighteners will straighten the water molecule flow, as they help minimize swirl and splashing, so the water can exit the funnels the same way in a laminar flow. Each funnel has been aligned to a laser diode above it in the dry zone. The laser light goes down the center of each water stream; however, there is a transparent plexiglass window that protects the laser and other dry electronics from the water, by isolating them from direct contact. At the bottom of each stream, there is another layer of plexiglass that separates the wet zone from the dry housing that contains the photodiode sensor below the funnel, which protects both the electrical part and visually detects the optical marker on the top of the water stream. Not only does the vertical flow allow for both visual and functional clarity for each string, but it protects the integrity of the electronics because they are very close to the water.

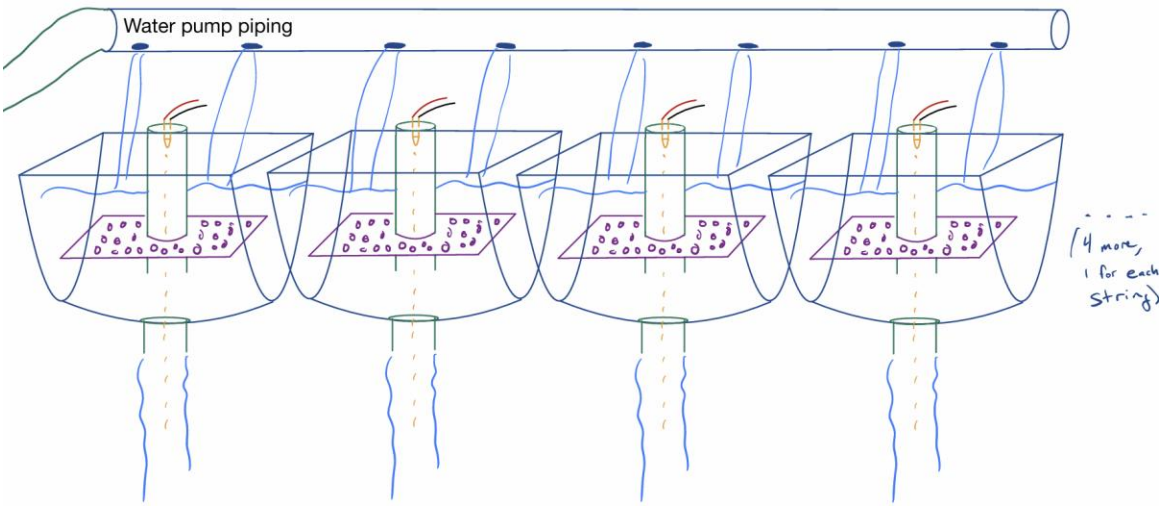

Figure 6.1.2: *Aqua Strings Top Sketch*

On a more technical level, to control these components a custom PCB is required but to get to a physical PCB a verified schematic is needed to ensure all connections are going to their proper spots. This main board will consist of managing components and power levels to ensure safe operation of the project. The main board is the hub of the entire system, where the microcontroller (MCU) is integrated, and all of the main inputs and outputs connect. It has several connectors for various laser modules and lasers as well as intermediate connectors for sensors, all interfaced through GPIO pins of the MCU, where pins can be utilized to switch devices on or off, or read signals usually assisted by supporting components e.g., MOSFETs. The board also has provisions for plug-in audio amplifier, only providing sound as an avenue for interactivity. The schematic layout is thorough and organized. It keeps both the digital and analog sections well organized and separates power components for clean routing and efficient operation.

The board has distinct headers for other support modules, like power regulation or storage, for modular expansion. A microSD card reader can be attached through dedicated SPI lines on the board. The placement of the laser modules and sensors is to receive power and control signals from the MCU as well. The power board also uniquely routes the 12V and other voltages to the corresponding connectors, and every component used will be supplied voltage without more complexity in the design. The schematic represents emphasis on modularity, clean signal routing, and close placement of all the necessary hardware.

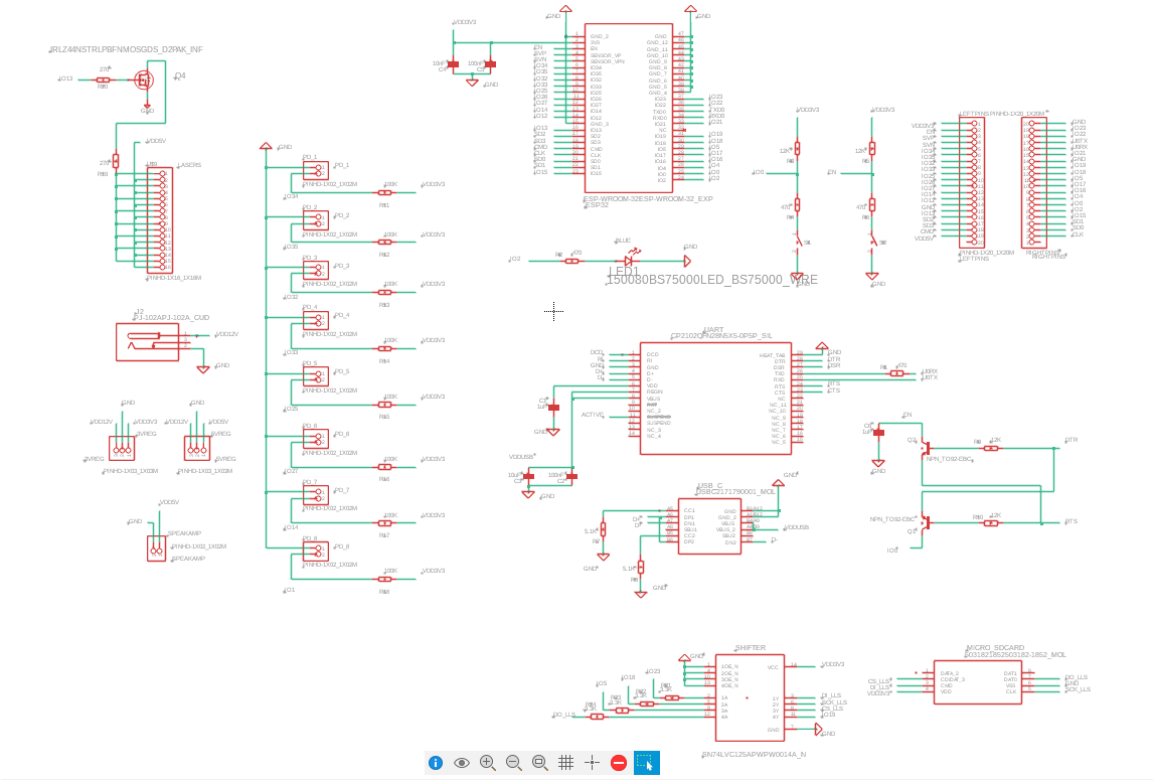

Figure 6.1.1: *The overall schematic (main schematic)*
**Note: See respective sections for a more detailed schematic*

A crucial consideration in our system design was separating the wet from the dry zones, not only in mechanical isolation but in the design of the electrical layout. All PCB mounted components sensitive to moisture, like the ESP32, SD card reader, and amplifier, are secured in a dry compartment on the top of the frame. This compartment is the control compartment where we isolate the critical electronics from the flowing water environment that adds unpredictable effects. The water-tight enclosures of the control compartment and elevated mounted dry components protect from splash zones, condensation, or pump vibrations, which improves reliability long-term.

To bolster this distinction, we route connections to components in the wet zone - laser receivers, transmitters & sensor wiring, through sealed cable glands, or waterproof connectors. This strategy is both safe and serviceable and again allows for change out of individual modules without compromising the integrity of the overall system. Of course, the laser and photodiodes will all need to align across the flowing water, and their function relies entirely on perfect positioning and clean wiring, which can be bolstered by the PCB with labelled headers and modular interfaces.

In addition to protecting the environment, the geometry of the board serves to minimize what could harm the signal routing and power delivery being stable and reliable to meet design performance goals. Every regulator on the board takes the 12V wall adapter input and converts it to 5V and 3.3V rails and delivers it all around the PCB, to power the components: ESP32, sensors, SD card module, etc. Each

voltage rail received capacitors and restrictions at the critical ICs to filter out noise from the motor and audio subsystems. This attention to electrical design provides faith that the control system will remain `true' to form, even amidst a very noisy platform due to water flow or pump activity.

6.2 Power Delivery & Management

The power supply circuit of this project relies on a 12V DC wall adapter, which delivers the main power supply to the entire circuit. This 12V power is input to the PCB through a barrel jack connector, i.e., the PJ-102A, and then delivered to multiple voltage regulators. The 12V rail directly powers high-current devices like the submersible water pump, in addition to being the input voltage for the two onboard voltage regulators that step it down to the lower voltages required by other components.

The LM2576-3.3 and LM2576-5.0 are regulators with fixed output voltages that take a higher voltage input (like 12V from a wall adapter) and efficiently step it down to 3.3V and 5V respectively, with very little margin for error. Within our research we found that the LM2576 series of regulators are the most used switching regulators available today meaning there is a ton of resources to aid in troubleshooting. They are known for their reliability, have a maximum output current of 3A, a thermal shutdown process, internal current limiting, while also remaining an affordable option. The unique feature of the LM2576-buck regulators is that they are not linear like the LDO regulator increasing its application to the project. An internal switch toggles the device from on to off, while using an external inductor to minimize the loss of energy while lowering the voltage. Thes mentioned features are shown below in Figure 6.2.1. As is commonly known, lower voltage is almost always necessary when powering microcontrollers, sensors, and any digital component, and switching regulators can be an effective and appropriate solution.

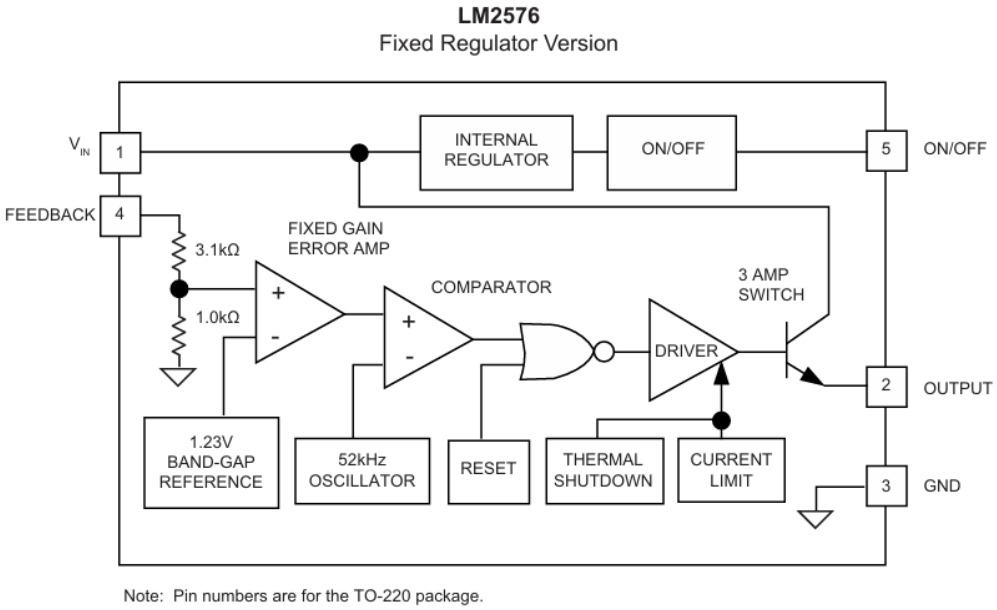

Figure 6.2.1: *Internal schematic of the LM2576 regulator*

For this project, we will use the LM2576-5.0 regulator to create a dedicated 5V line for peripherals including, but not limited to, laser modules and, while the LM2576-3.3 will power components that need a lower voltage, such as, the ESP32 if it was from that module, the microSD card module and the photodiode sensors. Each regulator includes the typical application circuit as described in its datasheet, containing an input capacitor, output capacitor, fast-recovery diode (something like the 1N5822), an inductor, and an LED to aid in troubleshooting. These components will hopefully output as smooth, regulated power with minimal ripple. The two LM2576 regulators will form the backbone of our power distribution system, taking in the 12V input and converting it to stable rails that the whole board can rely on.

Each of the voltage rails were routed to their own respective modules on the PCB with decoupling capacitors assigned close to some of the selected ICs to maintain functionality and avoid noise interference from one component to another during usage. The 5V rail was routed towards higher current peripherals like the laser modules and amplifier board; the 3.3V rail was separated to channel power to more sensitive digital components including the ESP32 microcontroller, microSD card interface and photodiode circuits. By separating aspects of the analog power and digital power where appropriate and adequately choosing the placement of the ground planes connected to the regulators, we will limit the potential for signal integrity issues or brownouts altogether. This intentional power layout resolution is beneficial to ensure the whole board, and all zones have satisfactory operational requirements, and helpful for troubleshooting and swapping components or sections if required if current draw varies unexpectedly.

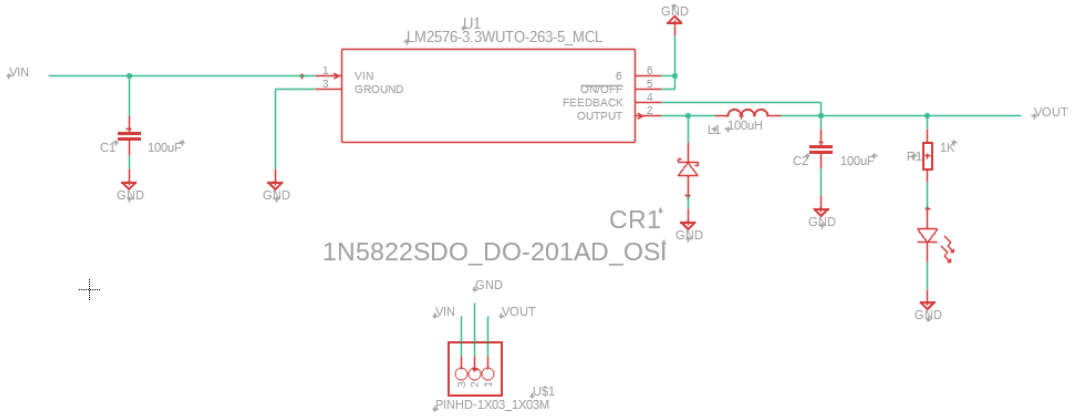

Figure 6.2.2: *The schematics of 3.3V and 5V regulator (same design)*

The PJ-102A barrel jack connector (labeled J2 in the schematic) is the power section of the main PCB and serves to supply 12V input from a simple wall plug power supply. In the schematic, Pin 1 and Pin 3 are tied to each other, and connect to VDD12V. Pin 2 connects to GND. The 12V line is then supplied to two pin headers for 3VREG and 5VREG, to which modular step-down voltage regulator boards are attached. Each header connects VDD12V, GND, and either VDD3V3 or VDD5V. The lower voltages are the output from either the LM2576-3.3 or LM2576-5.0 modules. The two headers also provide a way to bring the regulated

outputs back onto the main PCB cleanly and to supply it to components that only require lower voltages. A modular setup like this will be a simple task to debug and replace voltage regulators in the future, while keeping the 12V path organized and separate from the high-power DC paths.

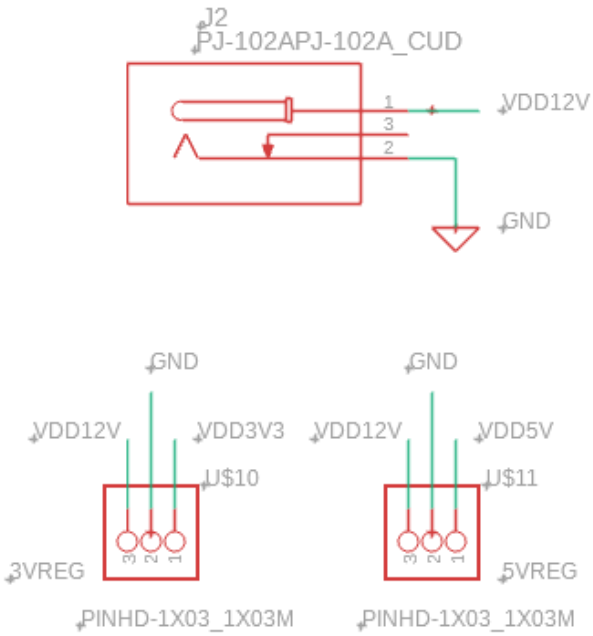

Figure 6.2.3: *The schematic of 3.3V and 5V regulator plug ins and 12V barrel plug on main board*

6.3 Laser Diodes and Photodiodes

The laser diode part of the master PCB takes multiple laser components and allows you to work with those components for beam-based detection and interaction. Each laser diode receives power from the supplied regulated 5V power. The control of the laser diodes is simple using N-channel MOSFETs wired to GPIO pins on the MCU. The positive terminal of each laser is directly wired to the 5V rail, and the negative terminal is wired to the drain of the MOSFET while the source of the MOSFET is wired to ground. Once the MCU sets the MOSFET's gate pin high, the transistor turns on and closes the circuit allowing current to flow through the laser diode. The main benefit of using MOSFETs to turn on/off each laser diode is to provide precise digital control of each laser diode on/off state and keeping a completely isolated system with as little load as possible on the microcontroller. By integrating this MOSFET, the lasers can be turned off allowing for the photodiodes to sense the ambient lighting in the room so that the detection threshold of the lasers becomes more accurate.

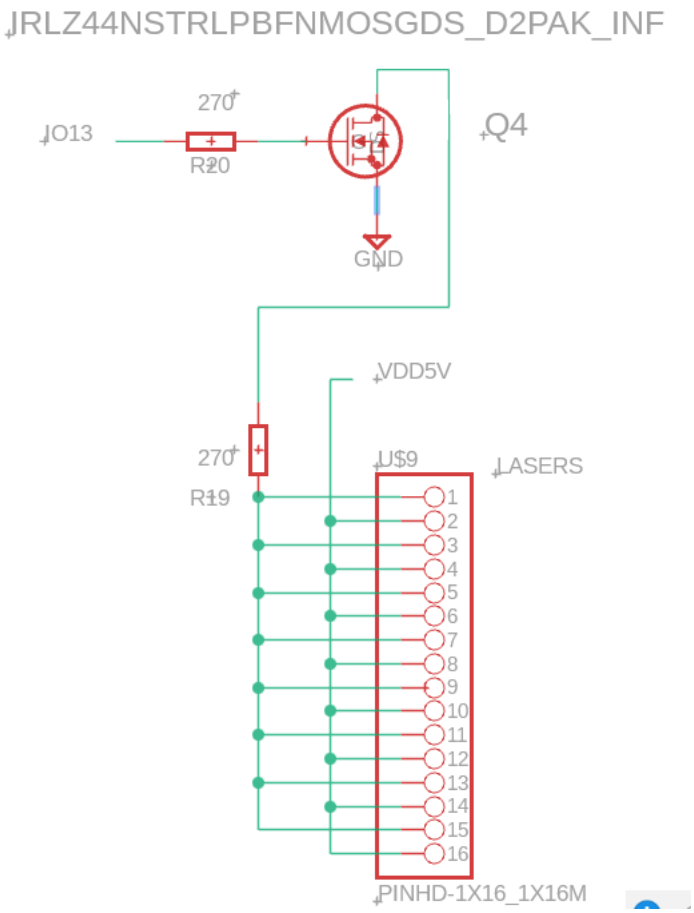

Figure 6.3.1: *Laser diode schematic with MOSFET on main board*

The photodiode portion is located near the laser dies to sense reflected or interrupted laser beams for capture as input. Each photodiode is connected in reverse bias, then the anode connected to ground, and then the cathode is routed through a current limiting resistor to a reading voltage, either 3.3V or an analog pin depending on the type of sensing in practice. When the photodiode detects the laser beam, it conducts a small photocurrent across the diode causes a voltage drop across the resistor and that the microcontroller can interpret as either a digital signal or an analog level. This section allows the main board to detect interrupted or reflected laser path and detect gestures or object sensing in the system's interactive capabilities. In the design, proper alignments and coverings are also planned to mitigate noise and provide better accuracy to detect signals.

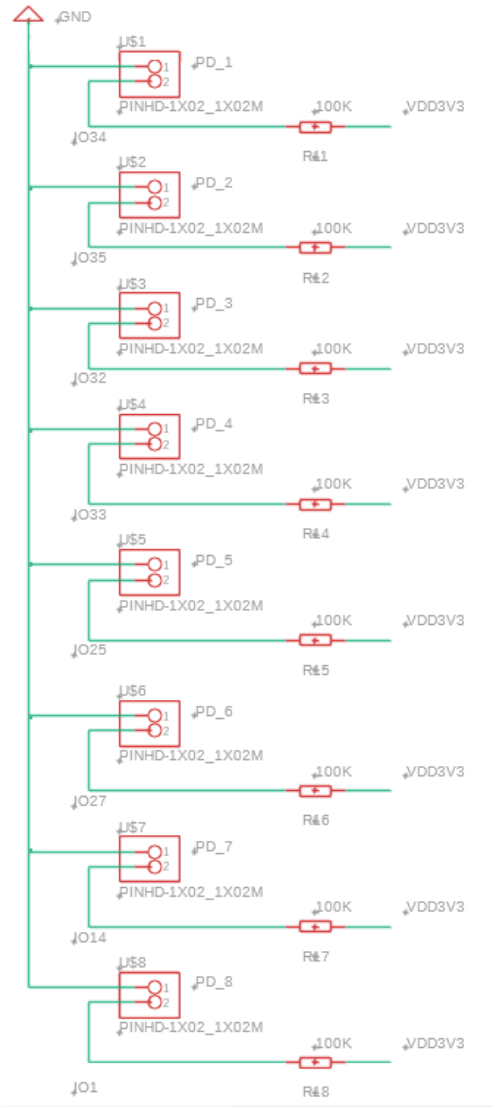

Figure 6.3.2: *Photodiode schematic on main board*

The schematic shown for the laser diode and photodiode section depicts how each laser is powered by the regulated 5V when it is controlled with the appropriate MOSFET tied to a microcontroller I/O. This allows for each individual laser to be input and output through software so that it can be dynamic during operation. The photodiodes are encased to detect the light from the laser beams and provide some input back to the microcontroller for when the beam of light is interrupted. The photodiodes connected to the board were assembled in a reverse-biased series, which allow for direct feeding of the signal through either a pull-up resistor or an analog input based on the desired level of sensing. In this way, the laser and photodiode circuitry provide a fast-sensing way of optical interaction, that is tightly integrated with the logic and power distribution of the main board.

The future versions of the laser and photodiode subsystem will also depend upon the outcome of our prototype evaluations. This aspect of the design will be particularly influenced by the reliability of the beam line alignment, incidence of

ambient light on the detection process and timing accuracy of responding to a collision, obstacle, or interruption detection. In the case that the first prototype tests gave inconclusive detections because of the incidence of ambient lighting or reflections, we may explore IR filters or covering the laser, or use signal modulation techniques in the design, to improve signal around the detection process to enhance reliability in detecting the beamline in the presence of ambient light. In addition, depending on how closely the photodiode response spectrum matched real time gestures or interruption, we might also add some amplifying circuit along with analog or digital filtering, to improve sensitivity and reduce noise. In terms of the laser control unit, it may be possible to expand the system to PWM brightness control or laser multiplexing, if there are opportunities or benefits to using even finer control or multiple detection layers. Overall, these future developments will depend on the operation of the system, with the objective of enhancing robustness and accuracy of interaction between all components.

6.4 ESP-32 MCU Design

The ESP32 microcontroller section on the main board is intended to mimic the features of the NodeMCU ESP-32S development board. In doing so the development board includes all of the features necessary for standalone operation including power supply regulation, boot and reset circuitry and USB-to-UART programming. The ESP32 chip has decoupling capacitors as well as a crystal oscillator and passive components designed to be duplicated from the original dev board configuration for viability purposes. This made it possible for our team to transition from the NodeMCU ESP-32S prototyping environment to the custom PCB environment without modifying firmware or changing workflow.

The board contains a CP2102 USB-to-UART bridge processor which allows for USB programming and USB Serial communication. It works by converting the computer's USB signals into the two standard UART serial signals (TX and RX) and then connects them to the ESP32's serial interface pins. The CP2102 also provides the DTR and RTS signals that are connected through discrete components, where the component circuitry provides the ability to control the EN and IO0 lines of the ESP32 for automatic reset during firmware upload and can select the boot mode via the CF0 line. The complete system provides the same features as seen on the NodeMCU board and provides firmware uploading and serial communication over a single USB-C connection.

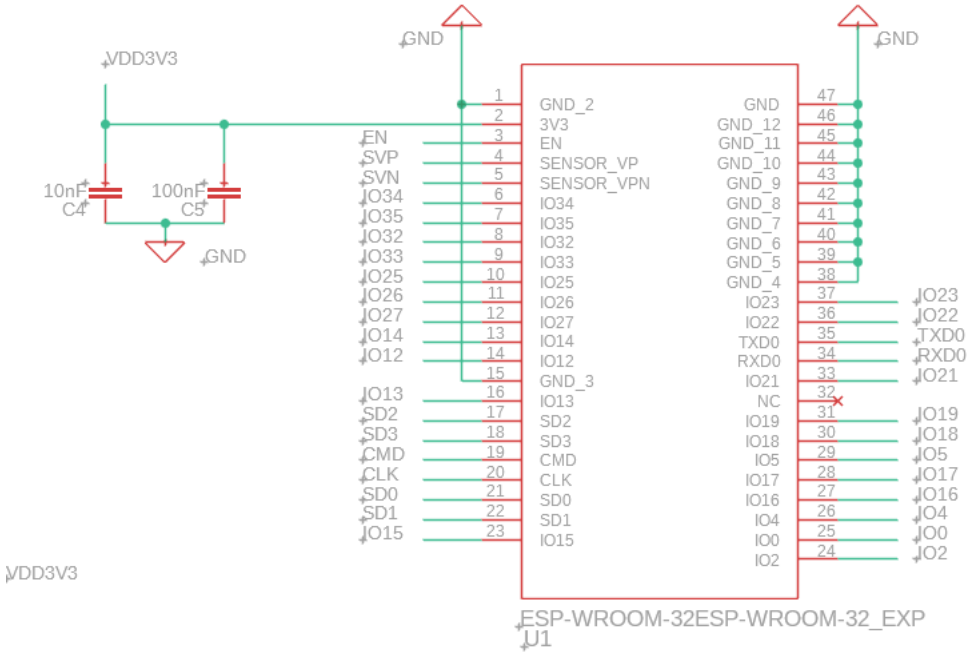

Figure 6.4.1: *ESP-32 module schematic/ layout*

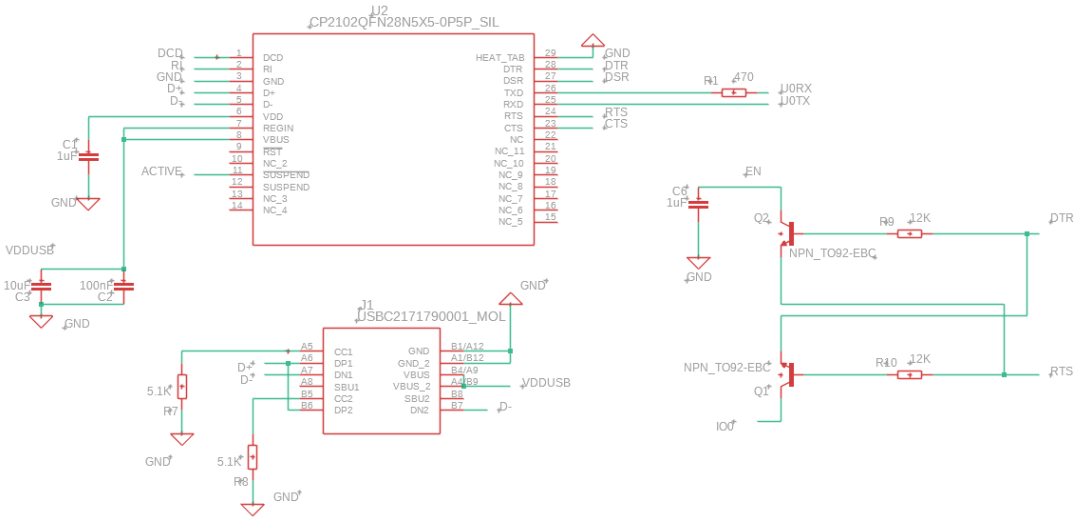

Figure 6.4.2: *USBtoTTL schematic*

Power is supplied to the ESP32 via a regulated 3.3V rail produced by the LM2576-3.3 regulator board. This rail powers the ESP32 and supporting peripherals, including sensors and other logic level devices. The board includes capacitors for local power stabilization and routed traces to minimize voltage ripple and EMI. The 3.3V line is isolated from the higher 5V voltage and the 12V lines used elsewhere in the system to ensure the ESP32 is operating within its safe voltage limits. The location of the selected ESP32 module on the board allows maximum access to I/O pin for servicing the rest of the system.

The ESP32 communicates to several subsystems of the main board layout including the laser and photodiode section, the amplifier stage, and external inputs from sensors. General-purpose I/O pins are used according to functionality and placement, using PWM capable pins as a laser control output and ADC pins for analog sensor data input. Every connection also has resistor protection in series with the connection, and in some cases, logic-level translation protection, to ensure electrical fidelity. The modular design of the board has allowed the key ESP32 elements, Wi-Fi, UART, I2C, GPIO, and ADC, to all be co-located with the surrounding components in a neat and reliable fashion.

6.5 Amplifier and Speaker Design

The amplifier module used in this circuit utilizes the PAM8403, a compact size efficient Class-D audio amplifier for driving stereo output speaker. Included in the schematic are all parts one needs to build, ranging from the decoupling capacitors, input coupling capacitors, to the input audio signals. The PAM8403 runs off the regulated 5V rail and delivers outputs through the left and right channel speaker terminal block. This amplifier is suitable for low-power audio playback and is marked by low distortion and high efficiency and is thus ideal for this type of embedded project where low power consumption and space are of the essence.

The audio signals are fed from the primary microcontroller board, where they begin as pre-processed signals or digital-to-analog outputs and come into the amplifier through specific pins. There are bypass capacitors and gain resistors, which can be added optionally, to stabilize and calibrate amplification. The design keeps signal paths short and well-decoupled, reducing noise and providing clean audio playback. The interfaces to the speaker are straightforward and offer stereo output to small speakers, which can be externally mounted depending on the final enclosure.

To further enhance the reliability and maintainability, the amplifier module is constructed on an independent board which is plugged straight into the main PCB. The modular design facilitates easy replacement of the PAM8403 chip should it be physically damaged due to overheating or electrical malfunction without the need for reworking the whole main board. The interconnect header brings the power, ground, and left/right audio signals, making the connection simpler and keeping the main board uncluttered. It is this philosophy of design that makes the overall system stronger, easier to repair, and more adaptable to changes or upgrades in amplifier type or capability later.

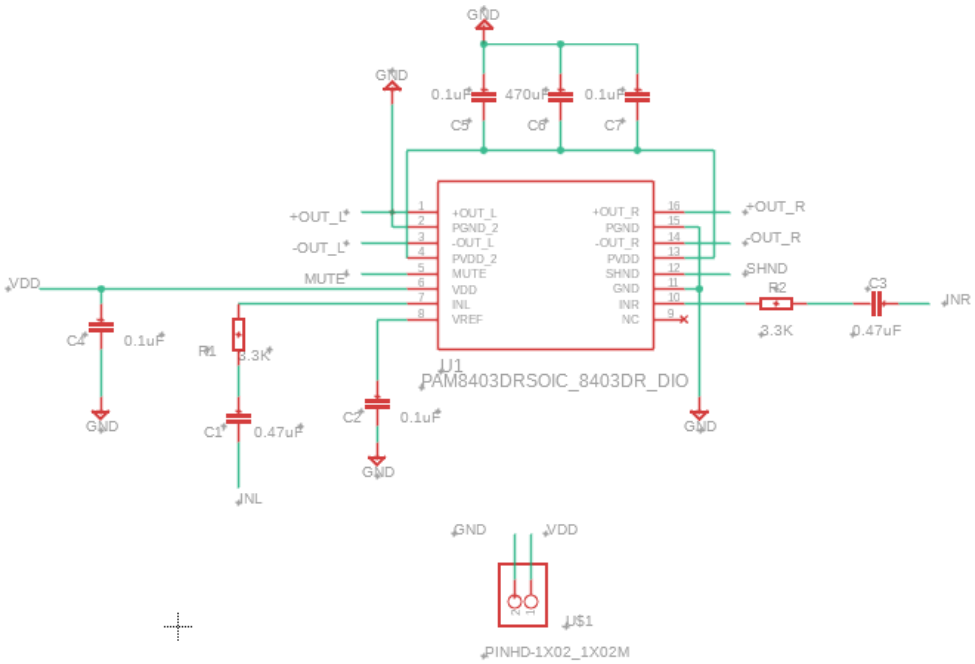

Figure 6.5.1: *The schematic of the amplifier module*

To assess the proper value of the input capacitor in the amplifier module I referred to the formula for the high-pass filter provided by the input capacitor (C_i) and the input resistance (R_i) of the PAM8403 amplifier:

$$f_c = \frac{1}{2\pi R_i C_i}$$

In the case of portable speakers, there is usually no need to reproduce frequencies below 100 Hz for many of the small speakers may not adequately reproduce the low low-end frequencies. Therefore, I settled on a frequency of roughly 100 Hz to fulfill the balance with audio performance and component size. The PAM8403 schematic uses a fixed capacitor value of 0.47 µF. By knowing this fixed value, it allowed me to re-arrange the formula to solve for R_i, and I determined close to 3.3k ohms for this cutoff frequency.

Using 0.47 µF caps and 3.3k ohms resistors both aid in reducing undesired turn-on "pop" sound. Note the datasheet states that larger caps are slower to reach their quiescent DC voltage and hence are more likely to generate clicks or pops when powered up, due to the internal feedback path charging the cap. However, keeping the cap to a reasonable size (but not too large) allows the system to reach equilibrium much quicker, while still maintaining good low-frequency audio performance. Thus, 0.47 µF is a good trade off on size and audio performance (the module operates in a 3.3V to 5V system, so the tradeoff is easier).

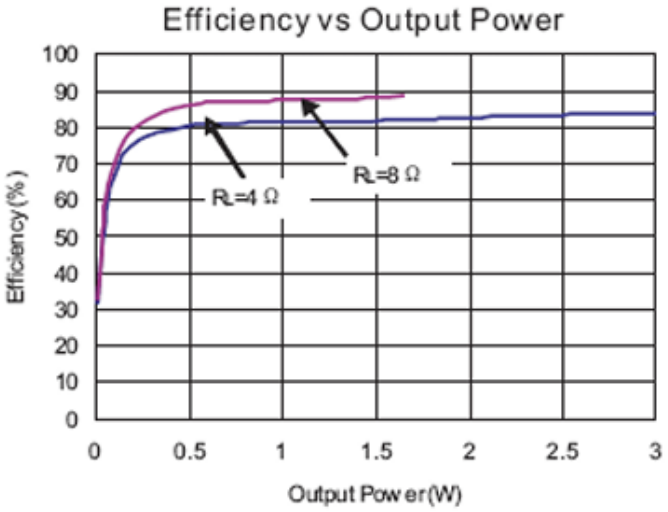

Figure 6.5.2: *Efficiency v. Output Power graph of PAM 8403 (Courtesy of Diodes Incorporated)*

The graph shows the PAM8403 Class-D audio amplifier's efficiency as a function of output power with 4Ω and 8Ω speaker loads. As can be seen, the amplifier is very efficient at all output power levels, with maximum efficiency in the mid-output power range. At the same output power levels, the 8Ω curve shows generally better efficiency than the 4Ω curve. This is expected because lower impedance loads such as 4Ω draw more current, therefore dissipating more power in the amplifier which leads to slightly less efficiency.

The PAM8403 and Class-D amplifiers like it are particularly advantageous in battery-powered or portable systems where ensuring the least amount of heat and the least wastage of power is important for extended battery life and maintaining cooler operation- especially in smaller designs. The PAM8403 exhibits over 80% efficiency even when fully loaded which helps to guarantee the system has the longest life during use. In your amplifier module, using this chip ensures that power is sent to an active speaker effectively with little wastage whether the speaker load is 4Ω or 8Ω.

6.6 MicroSD Card Module

The microSD card reader circuit is utilized to provide external storage and retrieval of data for the project, providing a reliable interface for logging data, audio file storage, or memory expansion for firmware upgrades. The component selected is a surface-mount microSD card socket, model 5031821852 by Molex, with a small size and the required pinout for SPI communication. These include CS (Chip Select), CLK (Clock), MOSI (DI), and MISO (DO), all of which are standard in SPI-based SD card interfacing. The CMD of the microSD socket is utilized for SPI's MOSI, and DAT0 is employed as MISO. These are routed through a voltage level translator to accommodate compatibility with the 3.3V logic levels found on the card.

Due to the fact that our main microcontroller operates at 3.3V but uses 5V-tolerant signals (and likely interfaces with other 5V peripherals), a SN74LVC125APW quad buffer is used as a level shifter. This chip safely shifts the levels from the microcontroller to the voltage the microSD card wants. The OE (output enable) pins of the level shifter are pulled low to enable the outputs all the time, and each data line is buffered to maintain proper signal quality over the traces that run to the microSD socket. This design reduces the chance of damage from a voltage mismatch and also ensures stable clean communication over SPI.

The microSD socket connects to the 3.3V rail (VDD33) for power, with proper ground referencing to avoid any signal issues. The decoupling capacitors could be placed nearby the VDD pin of the socket to maintain a stable voltage when accessing the microSD; accessing SD cards can be bursty and sensitive to power fluctuations. Signal lines (CS, SCK, DI and DO) are clearly broken out and labeled with _LLS for level shifting. The other signals are all covered as well; if the signal lines are labeled properly, it will make time spent tracking logic domains easier. Following the signal layout will save debugging time later.

The SD card interface can be utilized by the ESP32 module for a range of uses based on the application, from logs, configuration files, or to playback audio data. The SD card uses the SPI interface, so it does not require all eight data lines of full SDIO mode, making wiring easier and freeing MCU pins. Overall, this design uses the ethos of compact, safe, and performant external modularized storage.

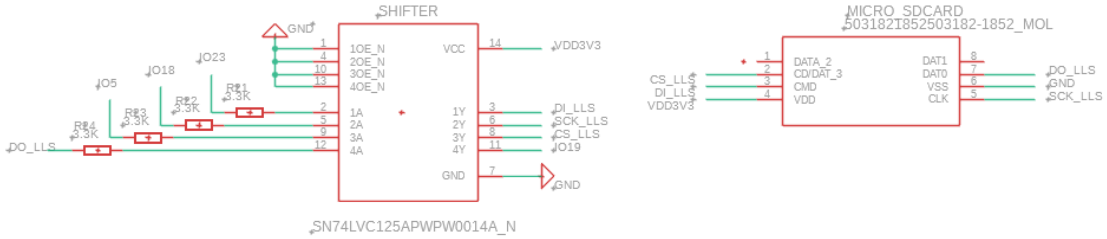

Figure 6.6.1: *The schematic of the microSD card reader module*

Regarding PCB Layout, we focused most closely on the routing of the SPI lines to the microSD card reader to minimize noise and maintain signal integrity. We ensured that the traces for CLK, MOSI, and MISO were short, and tightly coupled to a ground plane to minimize impedance mismatch and avoid potential issues with SPI communication. A series of resistors were also considered to be added to the SPI lines (especially CLK) as a dampening form of impedance to minimize ringing problems when operated at a high speed (clock speeds). Also, we could route the CD pin from the Molex socket to a GPIO input on the MCU. This would give the firmware the capability to more dynamically detect when the card is physically inserted or removed. This would provide the user with a much better interaction framework. Finally, the SD card will be able to be initialized in software using FAT file system libraries available in the ESP32, built-in SD_MMC or SDFS library depending on how we implemented the application. And all with robust read/write features, while using a microSD provides a great deal of flexibility in storage options for Aqua Strings.

Chapter 7: Software Design

The Aqua Strings system's software design is organized to provide uninterrupted operation, user engagement, and ease of managing system states. This chapter describes the software architecture through important diagrams that focus on system functionality and user engagement. The main program design describes the initialization and running of components such as the ESP32 microcontroller, water pump, laser sensors, and audio system with different modes to control the functions. The use case diagram illustrates user engagement with the system and how individuals will interact with the system to generate musical notes and visuals. Finally, the system state diagram describes the operational states and transitions in the system. This ensures that the responses of the system are clear, faults are managed, and the system can adapt in real time. All of these come together to develop the Aqua Strings software to provide smooth performance and good user experience.

7.1 Main Program Design:

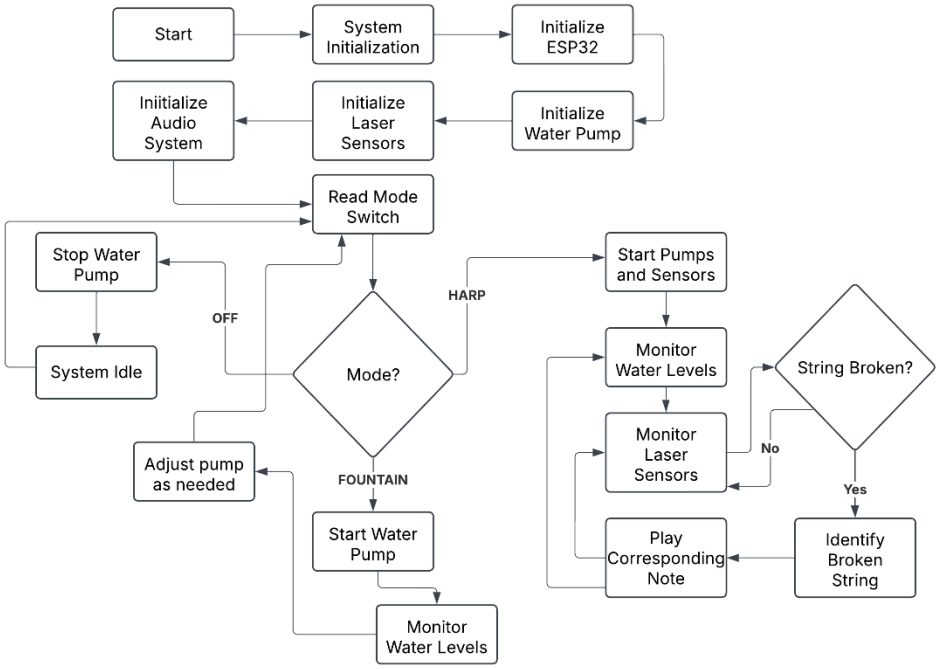

Figure 7.1.1: *Main Program Block Diagram*

The Main Program Flowchart illustrates the working flow of the Aqua Strings system from initial complete initialization of all the primary components like the ESP32 microcontroller, water pump, laser sensors, and audio system. Once initialized, the system proceeds into its main working loop, continuously checking the mode switch position to determine the appropriate operating state. OFF setting makes the system shut down the water pump and idle down; FOUNTAIN setting makes the system switch on the water pump for effect and switch off sound interaction; HARP setting makes the system switch on both water pump and laser

detection system for full interactive capacity. The system checks the levels of water in both reservoirs during operation continuously to provide optimum flow conditions and adjusts the pump accordingly. In HARP mode, it scans all the laser sensors at once for breaks, recording what water string has been broken when it senses one and producing the resultant musical note via the audio system. This systematic sequence offers responsive input without sacrificing water system integrity.

7.2 Use Case Diagram

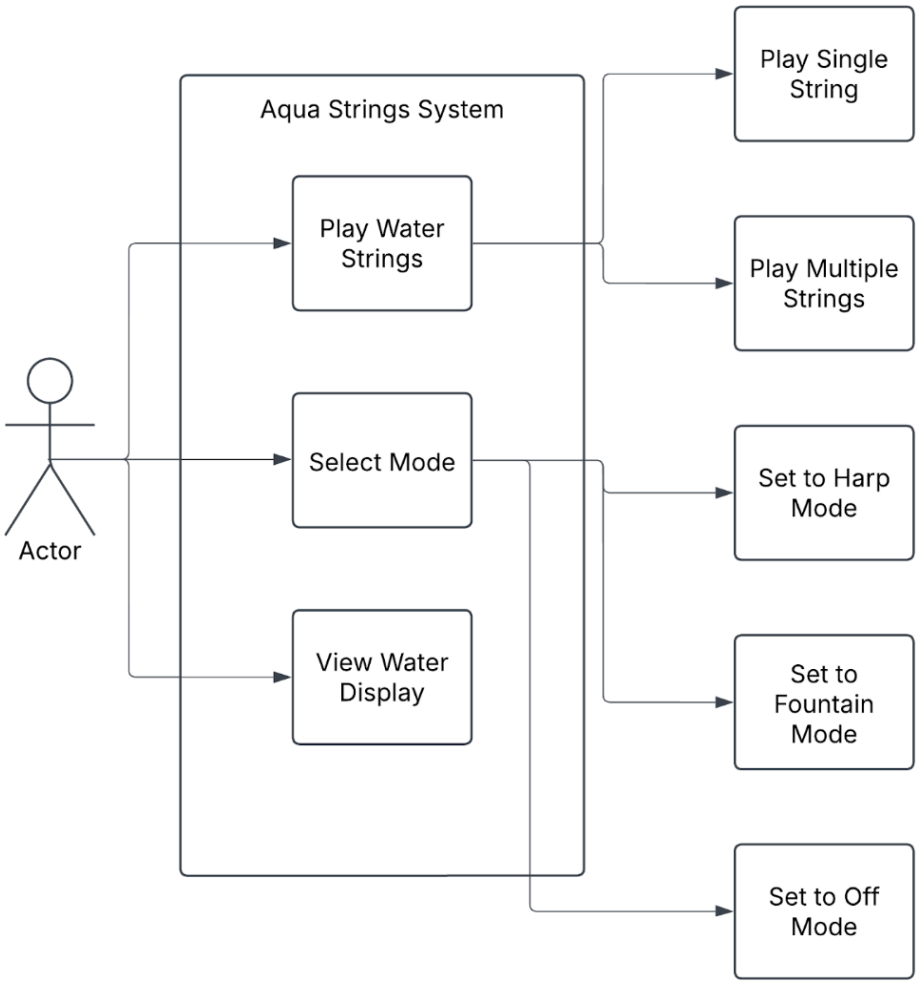

Figure 7.2.1: *Use Case Diagram*

The Use Case Diagram shows the various user interactions with the Aqua Strings system, which is the principal way an individual can interact with the interactive waterfall screen. The essence of the user experience is contained in five principal interactions: turning the system on and off, selecting between three various operating modes (OFF, FOUNTAIN, and HARP), playing the water strings by interrupting water streams, and watching the visual water display. The diagram also breaks up these interactions into more specific actions—users interrupt individual strings for solo notes or break a group of strings at once for chordal

effects for their optimal visual and interactive experience. This systems-level visualization of user interactions highlights the functional aspect from an outside-in perspective, with a focus on the ease of use and intuitiveness of the system over its internal technical processes. By mapping these interactions, the diagram is then used as a blueprint for user instruction and system creation for the Aqua Strings experience to meet expectations of interactive participation.

7.3 System State Diagram

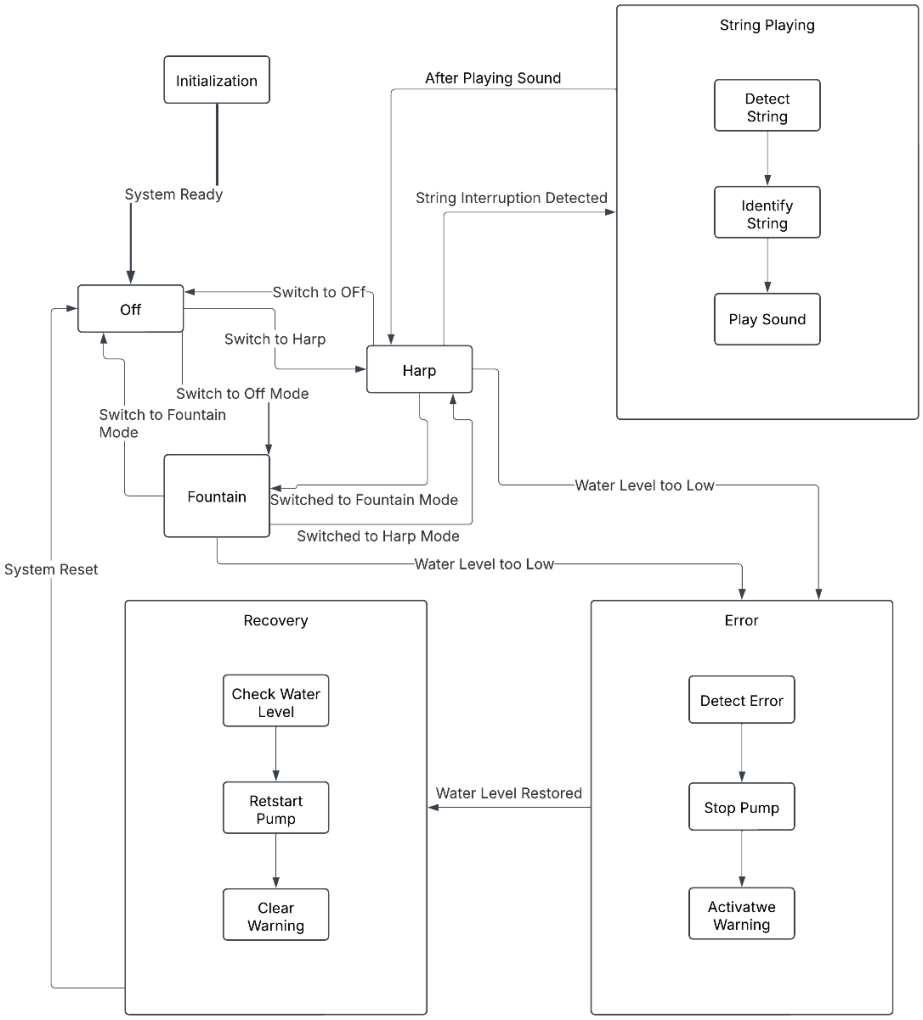

Figure 7.3.1: *System State Diagram*

The System State Diagram displays the states of operation and transitions among them of the Aqua Strings system, presenting an overall picture of the operation of the system in various states. From an Initialization state where all the systems are initialized, the diagram displays transitions to three principal operational states: Off (powered but not running), Fountain (flow of water without interaction with sound), and Harp (complete interactive mode with production of sound). The figure

illustrates how users can switch between these states at will by the mode selector, and how the system responds to some events like string interruptions and error conditions. In Harp mode and the presence of string interruption, the system enters the transient String Playing state where it figures out the specific string, plays the respective notes, and resumes the Harp state afterward. The diagram also defines error management in its Error state (triggered by low levels of water) and Recovery state (when water levels are refilled). Each high-level state contains nested substates representing the sequential processes involved in that mode of operation—for example, the String Playing state represents the sequence of detecting an interruption to sounding, and the Error state represents the sequence of detecting a fault, shutting down the pump, and warning. This hierarchical abstraction provides both a summary overview of system behavior and detailed information on individual operating processes.

7.4 Planned Software Features and Stretch Goals

As part of our ongoing development, several key features are planned on the software side to enhance interactivity, responsiveness, and user customization of the Aqua Strings system. These features are aimed at increasing both the musical expressiveness and the adaptability of the instrument through creative software implementations on the ESP32 microcontroller.

One of the primary enhancements is the ability to play multiple notes simultaneously, allowing users to break more than one water string at a time and create chords or layered tones. This requires implementing audio mixing functionality in software, where multiple preloaded audio samples are blended and output through the ESP32's DAC or I2S interface. By mixing overlapping `.wav` files stored on the microSD card, the system can emulate real-time polyphonic playback and maintain clarity even during simultaneous interactions.

Another planned feature is the dynamic adjustment of audio output based on interaction strength. As a stretch goal, we intend to measure the speed at which a water string is interrupted to estimate how hard or quickly a user "strums" the string. A fast, sharp interruption could result in louder and longer note playback, simulating velocity-sensitive behavior similar to acoustic instruments. This will be implemented by monitoring the timing and duration of laser beam interruptions, with mapped outputs that modulate the gain and decay of the corresponding sound (TensorFlow, 2025). Additionally, we aim to add a web-based interface as a user-accessible control panel. By configuring the ESP32 as a Wi-Fi access point, users could connect to the device through a local network and access a lightweight browser-based UI. Through this interface, users could switch between different musical scales, adjust the instrument's operational mode (e.g., harp or fountain), or modify sound settings like volume or playback speed. This stretch feature would not only improve user experience but also eliminate the need for physical controls, offering a more flexible and modern interaction method

7.5 SD Card Audio Handling and Real-Time Playback

7.5.1 Audio File Organization

Aqua Strings relies on a structured library of.wav audio files on a microSD card to provide musical feedback when a user interacts with the water streams. Each note is assigned to its own unique audio file, and in some cases, multiple versions of a note are stored to record different interaction dynamics for example C_soft.wav, C_loud.wav, or C_long.wav. These files are organized in directories based on different playback modes like /classical/, /synth/, or /ambient/, so that the ESP32 can quickly switch between whole sets of sounds depending on the selected mode. Not only does this file system organization keep the SD card tidy, but also it lays the foundation for easy future expandability, introduction of new instruments, scales, or tones.

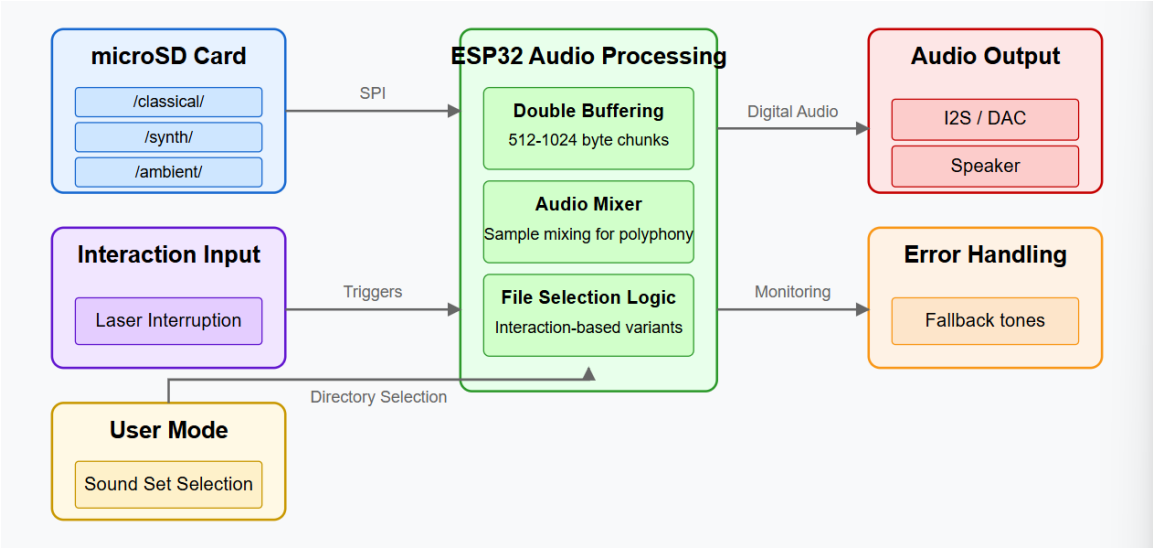

Figure 7.5.1.1: *SD Card Audio System Architecture*

7.5.2 Streaming and Buffering Strategy

To minimize memory usage while keeping playback low latency, the ESP32 plays back audio from the SD card using small, buffered chunks. Rather than reading full audio files into RAM, which would be beyond the ESP32's capabilities, the system reads data blocks, around 512 to 1024 bytes at a time while at the same time sending previously read blocks to the I2S DAC. This can be handled by libraries such as ESP32-audioI2S or ESP8266 Audio that manage this double-buffered playback very well (Schreibfaul1, 2023; Philhower, 2017). By reading the next chunk while playing the current chunk, the ESP32 is able to provide continuous and smooth sound output even for longer audio clips. Playing uncompressed .wav files also reduces processing overhead for quicker response without decode delays. This setup causes the musical response to sound instant when a user breaks up a stream of water.

7.5.3 Responding to Polyphonic Playback

Polyphonic playback, the ability to sound more than one note at a time is essential to a more natural and musically engaging user experience. Aqua Strings achieves this by mixing multiple audio streams together in software when two or more strings are engaged nearly simultaneously. Instead of playing back audio files sequentially, the ESP32 reads multiple.wav files concurrently and mixes their PCM samples together to create a blended output. The mixing occurs in real-time using light-weight methods such as sample-by-sample averaging (sampleA + sampleB) / 2). Libraries like XT_DAC_Audio and AudioTools have built-in mixer modules that make implementing this simple (WeekendWarrior1, 2019). Although performance is limited by memory and processing resources, the ESP32 can typically play two or three voices at a time with no delay or distortion, allowing for rapid strumming or overlapping gestures by the user.

7.5.4 Intelligent File Selection

To make playback more expressive, Aqua Strings includes logic for selecting alternate versions of each note based on user input. For example, a fast pass of the hand over a stream might make a louder or longer version of the note play, while a slow interaction might make a quieter sound play. These variations are accomplished by detecting the length or duration of the sensor break, and then playing the corresponding file based on a naming convention and pre-programmed file paths. In the future, this logic can be expanded to consider vertical hand position or gesture type, so that the same stream can be used to trigger different pitches or effects depending on how it is played (Edge Impulse, 2025). By choosing the right file on the fly, the system adds emotional and musical subtlety to what would otherwise be simple one-to-one triggering.

7.5.5 Error Handling and Fallbacks

To ensure robustness and reliability, file access and playback safeguards are included in the audio system. On startup, the ESP32 performs a basic integrity check to ensure that all folders and audio files that it expects to find are present on the SD card. The system records any file that is absent or cannot be read and can optionally also play a default fallback tone to maintain functionality. In operation, when the SD card becomes unplugged or corrupted, the audio engine pauses gracefully and provides visual notification through LEDs or control interface. Such fail-safes are important to guarantee reliable operation of Aqua Strings in controlled and public installations, where unexpected events like card ejection or power surges may occur.

7.6 Photodiode Calibration and Ambient Light Compensation

7.6.1 Calibration Procedure

To ensure reliable laser interruption detection under varying environmental conditions, the Aqua Strings system includes a software driven calibration routine

that runs at startup. The calibration process begins with all laser emitters turned off. During this phase, each photodiode sensor captures the ambient light level in the room, which is stored as the environmental baseline. Once these readings are collected, the system activates the lasers and immediately takes a second set of readings from each corresponding photodiode. The software then computes the difference between the laser-on and laser-off values to isolate the signal contribution of the laser. Using this difference, the system sets a dynamic threshold that determines whether a beam has been interrupted. This threshold is calculated as a value between the two measurements, often with a safety margin to avoid false positives. This startup calibration process ensures that each photodiode is tuned specifically to its operating conditions and can accurately detect a laser break regardless of the surrounding light.

7.6.2 Software Strategies for Reliable Sensing

From a software perspective, several strategies are implemented to enhance the accuracy and consistency of photodiode readings. The system maintains two key values per sensor: the current ambient baseline and the laser-specific signal. During runtime, the program continuously compares real-time sensor data against the pre-established threshold. To improve stability, the software incorporates basic filtering such as averaging or median smoothing, which helps eliminate rapid fluctuations due to water spray or flickering lights. In addition, the software uses hysteresis control to prevent rapid toggling of states when the signal is hovering around the threshold. For example, the detection threshold to register a “beam broken” event may be slightly higher than the threshold to return to “beam unbroken,” providing enough space to avoid false triggers caused by minor noise.

7.6.3 Adaptive Calibration and Dynamic Adjustment

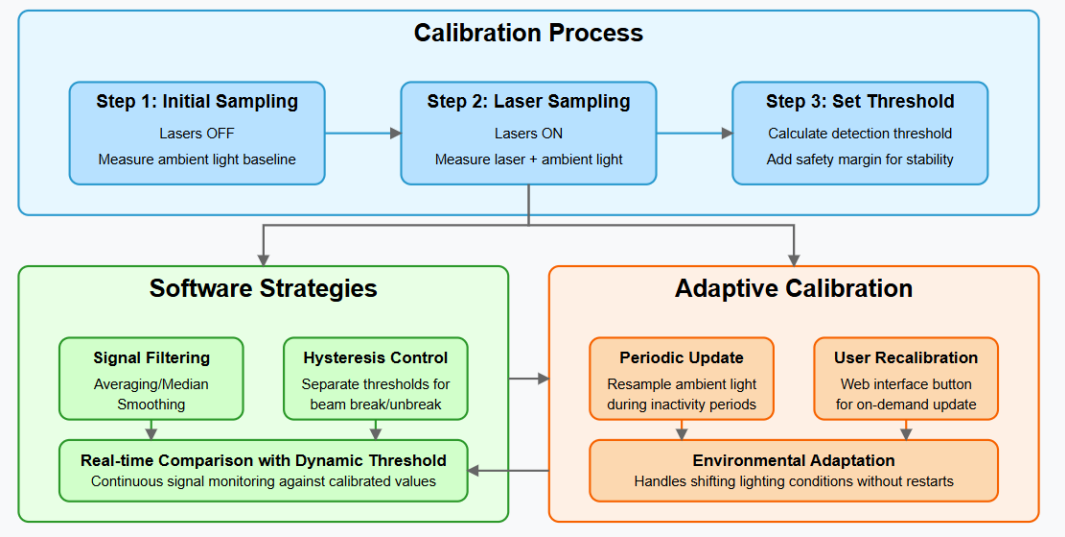

***Figure 7.6.3.1**: Photodiode Calibration and Ambient Light Compensation*

To accommodate long-term drift in lighting conditions such as changes in sunlight, indoor lighting, or reflections from water the software can periodically update the ambient light baseline. One approach is to re-sample ambient levels when the system has detected no user interaction for a prolonged period and the lasers are briefly turned off. This enables recalibration without requiring a full system restart. Another strategy is to include a user-triggered recalibration button within the web interface, allowing on-demand recalibration during setup or live operation (me-no-dev, 2025). These features provide flexibility for installations in changing environments, such as during live performances or public exhibits, where lighting may shift over time. By implementing a blend of startup calibration, real-time filtering, and adaptive thresholding in software, Aqua Strings ensures highly accurate, environment-aware sensor detection without relying on constant hardware adjustments.

7.7 Software Design for Active Water Flow Regulation

The Aqua Strings system uses a closed-loop water recirculation design, managed entirely through software running on the ESP32. Water is pumped from a large lower reservoir to a smaller upper reservoir, which feeds the falling streams used in the instrument's interactive interface. To regulate this flow and prevent overfilling, the system uses a float-based water level sensor mounted at the top of the upper reservoir. This sensor consists of a plastic pole and a magnetic float ring that rises and falls with the water level. When the water reaches the top of the reservoir, the ring aligns with a magnetic switch (e.g., a reed switch or Hall effect sensor), sending a digital signal of 1 to the ESP32. When the water level drops and the float moves away from the sensor, the signal returns to 0.

From a software perspective, this binary input is interpreted as either “full” (1) or “not full” (0). On startup, the system begins in a FILLING state with the pump enabled. While the float sensor remains low, the pump continues to fill the upper reservoir. When the float rises and the ESP32 receives a signal of 1, the software transitions into a WAIT state, disabling the pump. To avoid rapid on-off cycling of the pump due to small fluctuations in water level caused by minor disturbances or partial retraction of the float the software implements a debounce delay or hold-off timer. When the signal drops from 1 back to 0, the system does not immediately reactivate the pump. Instead, it starts a countdown (e.g., 5–10 seconds), during which the signal must remain low consistently. Only after this delay does the pump turn back on, re-entering the FILLING state.

This logic is implemented using non-blocking timing functions, such as millis() in the Arduino framework or the esp_timer API in ESP-IDF, allowing the main program to remain responsive while the timing logic executes in parallel (Arduino, 2025; Espressif Systems, 2025). The state machine controlling this behavior consists of three primary states: FILLING, WAIT, and COOLDOWN. The COOLDOWN state acts as a buffer zone to verify that the float sensor signal has remained low long enough to justify restarting the pump, reducing wear, and preventing instability. The float sensor is handled through a dedicated I/O pin

configured for digital input with internal pull-down or pull-up resistors, depending on the sensor design. The logic is encapsulated in a modular function or class that tracks the current state, last signal change timestamp, and cooldown duration. This modular approach allows the water control logic to be reused, tested, or extended independently of other system components.

Finally, to enhance transparency and remote monitoring capabilities, the current state of the pump and the float sensor reading can be published to the system's web interface. This allows users or developers to verify that the system is behaving correctly, initiate manual pump control if needed, or tune the debounce delay in real time. Overall, by combining a simple hardware sensor with clear software-driven state management, the Aqua Strings system ensures safe and consistent water flow, allowing the artistic interface to function reliably and continuously.

7.8 Software Design for Concurrent Task Execution on the ESP32

Aqua Strings combines several concurrent systems: laser break detection, audio play, SD card access, water flow control, and a web interface that must all work smoothly without interfering with one another. To achieve this, the software design is structured around modular operations, non-blocking logic, and FreeRTOS-based multitasking, all optimized to run on the dual-core ESP32.

7.8.1 Non-Blocking Execution Patterns

To make all the software subsystems responsive in real-time, they are written on non-blocking design principles. Reading photodiode input or water level monitoring with delay() or other blocking functions is prevented. These functions use millis()-based timing or software timers to call logic only when needed and then give control to the main loop at once. For example, photodiode sensors are polled at fast intervals, and audio playback is controlled using streaming libraries with DMA buffering. In this manner, during audio playback, the processor is still free to execute other tasks like reading sensors or updating the web interface. Similarly, the water pump controller has a non-blocking cooldown timer that postpones pump reactivation without halting the system.

7.8.2 State Machines for Subsystem Control

All core subsystems are coded as state machines in order to programmatically define what operating logic it should have. These include AudioManager, SensorManager, and PumpController, which all have internally defined states and transitions based on input or passage of time. For example, the pump controller is in FILLING, WAITING, and COOLDOWN states. The state transitions are based on the float sensor reading and a software timer. This state-based approach makes the code modular and deterministic. State machines all have a single update() method invoked periodically in the main loop, which provides tight integration without blocking.

7.8.3 Task Segregation with FreeRTOS and Dual-Core Utilization

To make the most of the ESP32's dual-core processor, Aqua Strings uses FreeRTOS to create dedicated background tasks. These tasks are pinned to individual cores using xTaskCreatePinnedToCore(), isolating time-critical functions from lower-priority functions (Espressif Systems, 2025). For instance, real-time audio playback and sensor polling tasks can be scheduled on core 0, and the asynchronous web interface and pump control loop can be run on core 1. This partitioning ensures that user interaction and visual feedback are never delayed by network traffic or file system usage. Each FreeRTOS task has its own loop with a vTaskDelay() to relinquish execution back to the scheduler.

7.8.4 Synchronization of Shared Resources

Certain system resources such as the SD card and serial output are used by multiple components. To prevent conflicts, access to shared resources is synchronized through flags or mutexes (Espressif Systems, 2025). For instance, before commencing audio playback, a flag indicating SD card use is set up by the audio subsystem. When the web server also attempts to utilize the SD card, it looks at the flag and retries once the resource is free. This ensures data consistency and avoids unspecified behavior. Also, facilities like the web interface are designed to schedule tasks (presenting accessible sound files) rather than attempt their immediate execution, which facilitates the safe coordination of accesses.

7.8.5 Software Timers and Event Scheduling

Rather than processing each action in continuous run mode, the system uses lightweight program timers to regulate how often separate actions occur. These are carried out by comparisons of millis() against timestamps once stored, so the main loop can watch out for time events without suspending execution.

Applied to:

- Management of cool-down periods between pump activations
- Sensor state transition debouncing
- Refreshing web dashboard outputs
- Running background calibration routines

By coding events instead of iterating them, the ESP32 avoids idle cycles and is available to process real-time responses to sensor inputs or audio events. combining non-blocking functions, state machines, FreeRTOS task scheduling, and good resource synchronization, Aqua Strings provides smooth, concurrent operation of all its major systems. This software architecture ensures responsive interaction, efficient use of hardware, and flexibility for future extension or feature addition.

7.9 Software Stress Testing and Reliability Evaluation

To ensure Aqua Strings performs predictably under real conditions and heavy usage, rigorous stress testing is done during software verification. These stress tests will determine how the system reacts to prolonged use, rapid interaction, and simultaneous subsystem usage. The goal is to uncover edge cases, resource limits, and potential failure modes before shipping.

One of the key focus areas is concurrent subsystem operation, where two or more user inputs (laser beam interrupts) are being triggered one after another or simultaneously while audio is in active play and pump controller is in an ON state. The program is exercised within its ability to manage this type of load without loss of triggers, warping sound, or system crash. Automated test routines cause constant string triggering over extended time periods, enabling event-driven, non-blocking design to withstand the test of time.

Another stress situation is the playback of audio from the SD card continuously while background activities like polling of water level, debouncing of sensors, and servicing of web interfaces are ongoing. These tests confirm the efficiency of DMA buffering, ensure that there are no memory leaks, and confirm that SD access conflicts are properly avoided through task synchronization and mutex management.

The water control logic is also stress-tested by actively controlling the float sensor signal to simulate unstable water level oscillations. Debounce logic, cooldown timers, and pump state transitions are observed so that the system will not enter rapid or unstable cycling. Logging is enabled in these tests to observe state transitions and verify that every event is processed only once despite noisy input.

Stress testing is extended to the web interface, in which several clients can connect simultaneously to the ESP32's web dashboard or access point. Load testing includes frequent parameter updates, file browsing requests, and real-time sensor value streaming to ensure that the asynchronous web server continues to respond without interfering with core audio or sensor operations. System stability and memory consumption are monitored in all stress tests. ESP.getFreeHeap() and uxTaskGetStackHighWaterMark() are utilities used to monitor dynamic memory consumption and task stack usage to prevent any component from exceeding supplied resources (Espressif Systems, 2025). In case of persistent faults, the system logs error events and may incorporate a self-recovery mechanism like soft reset, watchdog trigger, or safe-mode fallback to maintain uptime. Through formalized stress testing along all critical avenues sensor input, audio streaming, water management, and networking the Aqua Strings software is hardened against unexpected states and trusted for constant, failure-free operation both during normal operation.

7.10 Version Management and Team Collaboration with GitHub

GitHub is used in the Aqua Strings project as the center for version control, source code management, and team documentation. Git provides an auditable means of tracking modifications through time, and GitHub allows everyone involved an open area in which to see, modify, and manage the project's firmware, configuration files, and documentation. The codebase is organized by means of an ordered branching policy. Active development occurs on feature-specific branches, sensor-calibration, audio-system, or web-interface so that team members can work on new functionality without compromising the stability of the base code. Changes are merged into the main branch once they have been tested and reviewed, which represents the most recent stable version of the firmware.

Members should use the collaborative tools GitHub offers natively, like pull requests, issue tracking, and code review, to collaborate. This guarantees any contribution whether its touching critical logic, refining interfaces, or adjusting configuration is merged safely and transparently (GitHub Docs, 2025; Kinsta, 2023). Issue tracking makes it possible to flag bugs, suggest improvements, and maintain open communication about work being done during development.

In addition to code, GitHub is also used to version and hold project documentation, such as wiring diagrams, system architecture notes, and calibration procedures. This keeps all technical references up to date and accessible to everyone on the team. As development progresses, tagged release versions are pushed onto GitHub, providing a tidy history of change and quick rollbacks in case this becomes an option. This process facilitates efficient collaboration, promotes accountability, and leaves Aqua Strings with a tidy, documented, and stable pipeline for development for its whole lifetime.

Chapter 8: System Fabrication and Prototype Construction

One of the main requirements of this project is to have a substantial and implementable PCB design. This design must include at least one MCU, a power supply unit, and several peripherals that are integrated with the MCU. Additionally, no breadboard or solderless connections are allowed. This portion of the project is significant in many ways. For one, our project will involve a large number of peripherals. If we are able to get the harp to work at full scale, it will require eight individual laser diodes and eight corresponding photo diodes. To simplify these connections, our PCB will be designed to power all eight laser beams from the same power connection. Then, each photo diode will receive its own connection directly to the MCU. Another reason why the PCB is so vital is that the audio output will require several development boards. To upload and read audio files (.WAV or .mp3), a microSD module is required. And to power and scale the voltage for proper speaker volume, an amplifier module is required. According to stipulation three, a final product with multiple development boards integrated is not sufficient. We are required to replace some, if not all, of these development boards by following their schematic diagrams. Since both of these units are external, pre-built boards, we will redesign and incorporate them into our custom PCB. Alongside satisfying the previously mentioned stipulation, this will allow us to work on our audio system from one custom board. Additionally, our project will handle many different peripherals at different voltage levels. The ESP32 will receive an input voltage of 3.3V (as compared to its potential 5V input that gets regulated internally anyways). The laser diodes will need a 5V input to power their beam. The amplifier that will power the speaker also requires a 5V input. Also, the power from the pump will need to be moderated by the PCB. Considering all of these power factors and how the PCB can solve them, it’s readily apparent how vital this portion of the project is.

8.1 PCB Design

We began with the main custom PCB. This board will house the ESP32S and be designed to house all of the connections with the peripherals. Specifically, there is a line of pin connections along the top of the board that provides 5V in parallel to eight access points. Additionally, nearby there are eight pinpoints for the sensors as well, driven by the 3.3V. Both of these features are located near the top middle of the board and are adjacent to the rest of the GPIO pins. These are routed to empty pin spaces in the case that we need to use them. At the bottom of the board lies the microSD card reader, which is just below the USB_C port. Both of these ports will be used to flash and update the programming on the ESP32. On the right side of the board lies the 3.3V and 5V outputs. To power these two separate voltages, the main board will receive a regulated 12V from one of two regulator boards, which features two variants of the recommended buck regulator, LM2576.

8.1.1 Regulator PCB

To properly manage the different power needs across the system (i.e. the 3.3V and 5V requirements for some components), a dedicated regulator board was designed as a separate PCB layout. This board handles voltage conversion and distribution by stepping down the 12V input from the wall adapter into two regulated voltage rails: 5V and 3.3V. To achieve this, we used the LM2576 series of buck regulators. These are switching regulators, which makes them significantly more efficient than linear counterparts, especially for a project with power-sensitive zones like ours. Additionally, they were explicitly recommended by the professors overseeing the project, which made them an easy pick as the main component for this board.

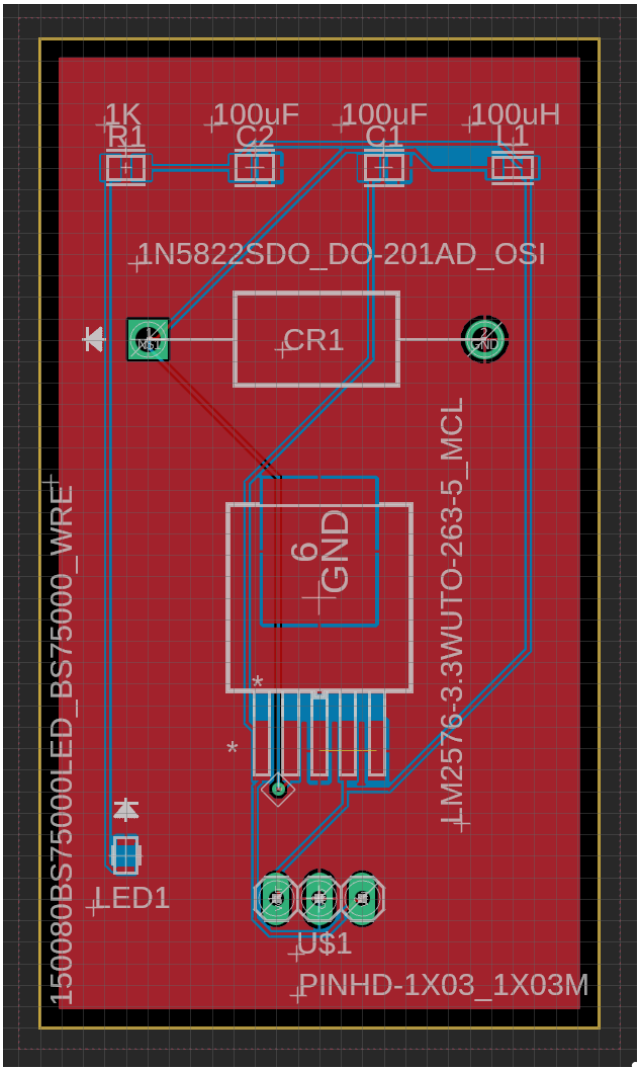

Figure 8.1.1.1: PCB Design of Voltage Regulator (identical for both 3.3V and 5V

Both regulator boards were designed around two separate LM2576 modules—one for each voltage level. The LM2576-5.0 supplies the 5V rail, powering peripherals

like the laser modules and amplifier, while the LM2576-3.3 supports the ESP32, microSD reader, and photodiode sensors. Each regulator follows the typical application schematic from the datasheet, including a fast-recovery diode (such as the 1N5822), input and output capacitors, a standard 100µH inductor, and an LED to indicate power status. These components help suppress voltage ripples and stabilize the output under varying load conditions.

Both voltage rails are returned to the main board via 3-pin headers on the regulator, consistent of the input VDD12V, GND, and the respective output voltage (either VDD5V or VDD3V3). These headers simplify testing and replacement and keep high-power and low-voltage paths organized. They also coincide with the design ideology of these regulators. These boards were intentionally kept separate from the main board in the event that they burn out or break. Having the critical components, like the microcontroller, isolated on the main board away from the regulators, safeguards them from damage.

8.1.2 Main Microcontroller PCB

The brain of the project lies in the main PCB design, which houses the ESP32 microcontroller alongside all the core peripherals: the laser diodes, photodiodes, GPIO pin headers, audio amplifier, and microSD module. This PCB was designed to house the entirety of all of the electronic connections for the project, including room for expandability with the additional pin headers. All power, logic, and control are routed through either direct component connections on the board itself of labeled pin headers that correspond to off-board modules or components.

At the heart of this board is the ESP32 (although located in the upper left of the design), which communicates with the computer via USB through the CP2102 USB-to-UART bridge. We opted for this setup to allow firmware upload and serial debugging directly over USB via the USB-C port located on the left border of the design. Both of these components were placed physically close to each on the PCB layout for clarity. The CP2102 bridge is located adjacent to the USB-C port on its right, while the port itself borders the left edge for greatest ease of access. Basic flash and reset circuitry were included to support manual uploading, although automated upload was not implemented due to time constraints.

To control the laser diodes, the ESP32 drives the gates of three N-channel IRLZ44N MOSFETs, which are located in the center of the board. Each MOSFET switches the ground line of a laser diode, and the MOSFETs are placed on the board with pull-down resistors to ensure the lasers remain off by default at boot. For detection, each of the three photodiodes is reverse-biased and connected to a dedicated LM324 op-amp configured as a transimpedance amplifier. This configuration converts the photocurrent to a voltage signal, which is then routed to the ESP32’s analog-to-digital inputs. The op-amp stages are critical to amplify the low-level signals from the photodiodes to a usable voltage range.

All laser and photodiode headers are labeled and broken out along the perimeter of the board for ease of access and wiring. Power is supplied to the board via pin headers along whichever voltage rail is needed (3.3V or 5V). Both of these rails and their initial pin headers are located on the right end of the board. The design of the laser's pin headers is of a single, continuous row while each photodiode features its own pair of pins in two separate columns.

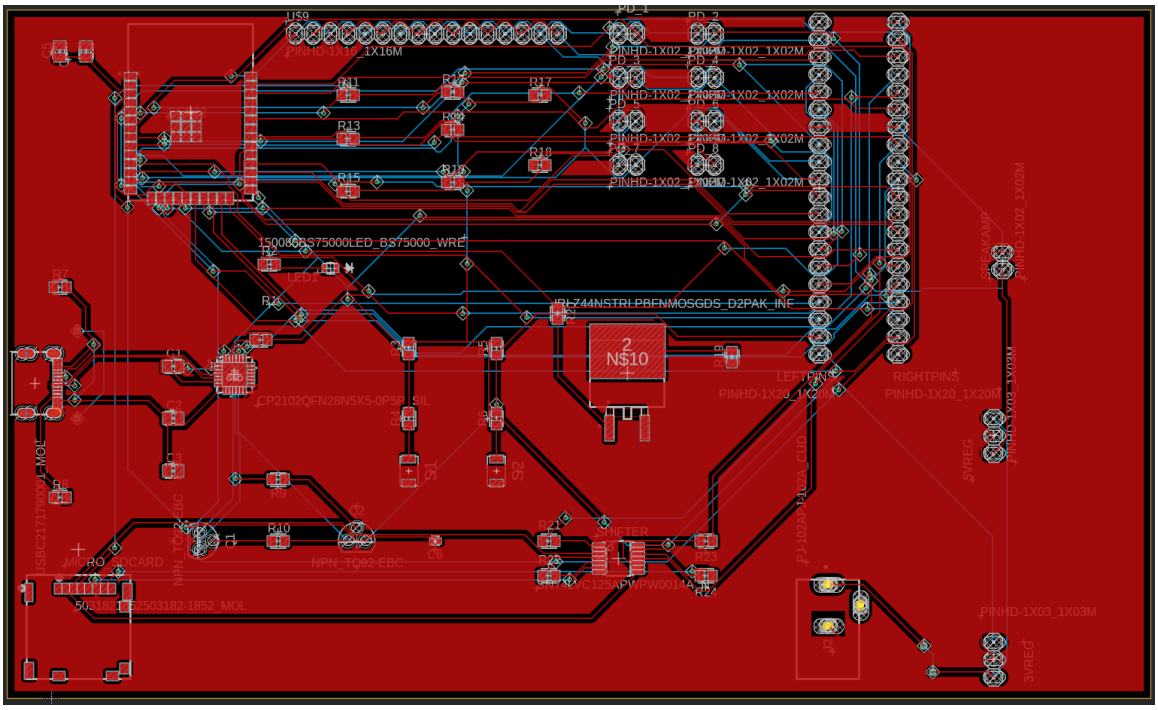

Figure 8.1.2.1: *PCB Design of Microcontroller Board*

8.1.3 Audio Amplifier PCB

To drive a speaker directly from our ESP32-based system, we designed a standalone audio amplifier board using the PAM8403, a low-noise, class-D amplifier IC capable of delivering 3W per channel into a 4Ω load. Through testing, we've determined that this amplifier provides enough power to drive an 8Ω speaker. That being said, after further research has been completed with this amplifier, its usage may be better synchronized with a less powerful passive speaker. Regardless, the design of its PCB remains unchanged despite which peripheral it is paired with. While an off-the-shelf amplifier module is commonly available and was used in our prototype, we chose to recreate our own custom PCB per the requirements of the project. Based on the reference design in the PAM8403 datasheet, we customized this PCB to better integrate with our system and tailor the layout for our enclosure on the back of the harp's frame.

The amplifier circuit consists of the PAM8403 chip configured in stereo mode. Supporting components include two feedback resistors and seven ceramic capacitors, primarily used for power supply decoupling, gain setting, and output filtering. All capacitor values were chosen based on recommendations in the

reference design to maintain audio fidelity and reduce noise and ripple in the output signal. These supporting components were placed around the chip housed in the center. They were spaced for ease of soldering and balance in the design.

The board breaks out three 2-pin headers: one for power input (VDD and GND), and two for the left and right audio output channels. For clarity the left and right channel's headers were placed on their respective sides of the board. Both pins were oriented vertically so that the plus and minus pins were in the same order (top to bottom respectively). The power input pin was placed in the center in a different orientation to differentiate itself from the other two. This separation of PCB modules allows for more flexibility in system layout while reducing reliance on pre-built boards.

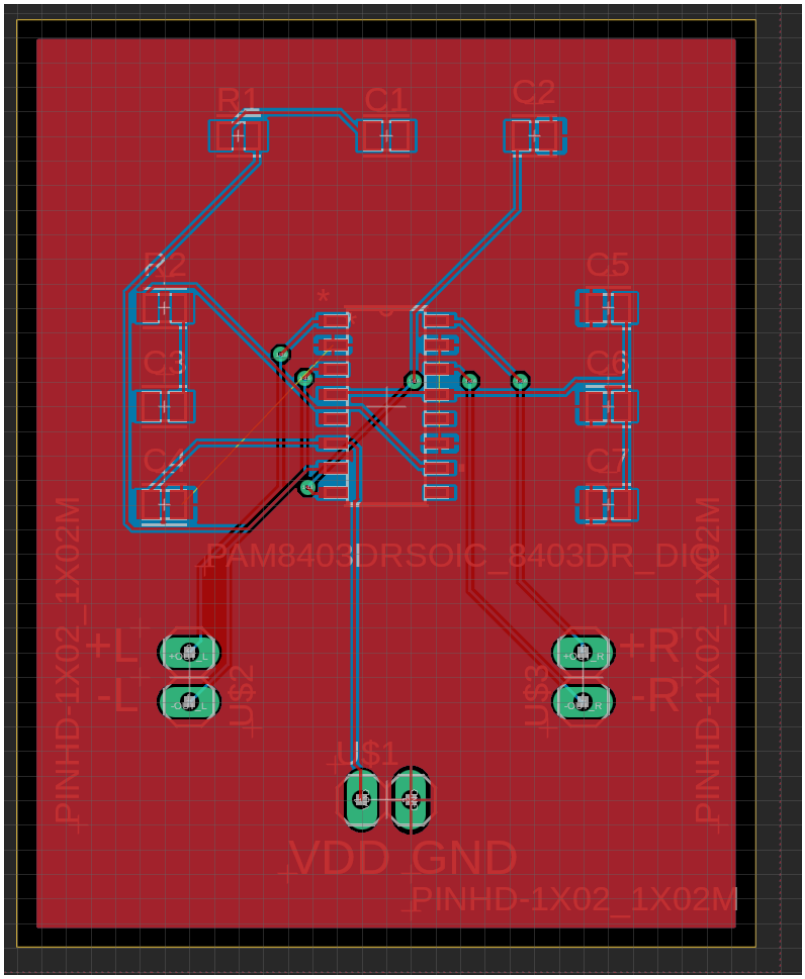

Figure 8.1.3.1: *PCB Design of Audio Amplifier Board*

Chapter 9: System Testing and Evaluation

This chapter will cover the hardware and software testing, performance evaluation, overall integration, and plans for Senior Design 2. This chapter is vital as there is a significant level of uncertainty regarding Aqua Strings design. Much of this uncertainty comes from the implementation of flowing water. Factors like how the water flows, how it interacts with the electronic components, and how it can jeopardize the other systems are serious considerations during our design. The best way we can learn about these concepts is through testing these systems in tandem. Even disregarding the water element, the project relies on so many different elements that need to work accordingly. Our best means of ensuring a reliable and consistent project is through continuous prototyping, experimentation, and testing. This framework will hopefully carry the design through to Senior Design 2 where we will ultimately build more intricate iterations. If all goes well, after considerate system testing and evaluation, we will develop our project towards its final iteration.

9.1 Testing and Overall Integration

9.1.1 ESP32

Testing begins with the most important component of the project: the ESP32. The MCU is involved with all of the systems in this project, so ensuring it works accordingly is of great priority. For prototyping, we tested the ESP32S chip with the ESP32 WROOM Devkit. For information regarding this MCU, consult chapter 3 and Appendix C. The most important features we wanted to test were the ESP32's built-in DAC and its ability to output either 3.3 volts or 5 volts output. There are twelve analog input pins on this board, but only two analog output pins. This is acceptable as our project shouldn't need to leverage any more than two speakers. These two pins will act as the output signals that will drive the different speaker tones according to what strings are played. The other pins can be used for interfacing with the photodiodes. Using USB-C, we interfaced with the board and flashed it using Arduino IDE. Simple checks showed that the both the analog outputs and voltage outputs were working. However, we discovered that the board can only use its 5V input as an output when it's being powered through its USB-C port. This worked for the prototype, as the board needed to be plugged in to update and tweak its programming. However, for the final design, the board will not be using this method to receive power. Instead, our custom PCB will include a voltage regulator that will output the necessary 3.3V to power the board. Thus, this method of 5V output cannot be utilized to drive the speaker amplifier. This is an important discovery as it changes the design of our voltage regulator. Its output of 3.3V will go towards the ESP32S while the 5V output will go towards the speaker amplifier and laser diodes. This testing of the ESP32 Devkit allowed us some insight into how the specific chip operates.

9.1.2 Laser and Photodiode Sensor

The laser detection system is the first critical system that we decided to test. The whole project leverages the photodiodes' ability to detect when there is a break in the beam. Thus, we set up a simple frame to house the laser beam and homed in to where the photodiode would be placed. We designed the prototype to reflect the approximate distance between both electronics that would exist in the final product. After some slight modifications, the laser beam's red dot rested steadily on the photodiode. We then constructed a simple circuit that allowed us to test the ability of the photodiode. Utilizing the USB-C port to power and interface the MCU, we leveraged the 5V pin as an output to provide power to the laser diode. The photodiode was connected to one of the board's analog input pins. Additionally, it was also a driving an LED on a small circuit board. The intent of the prototype was that when the laser was interrupted, or "strung," the LED would light up, indicating the sensor is working appropriately. However, before we could begin testing, an important piece of setup is required with the photodiode. For every time we tested the sensor, we had to ensure that the same relative amount of ambient light was hitting it. This meant leaving a small cover over the end of the frame where the photodiode was housed. If we didn't, the environment's overhead lighting would saturate the sensor's range of detection and make it inconsistent. This shouldn't be a problem in later iterations of the project's design, as the photodiodes will rest in the lower reservoir and be shielded by both the upper half of the frame and the water. For the sake of this prototype, however, we shaded the sensor to ensure good results.

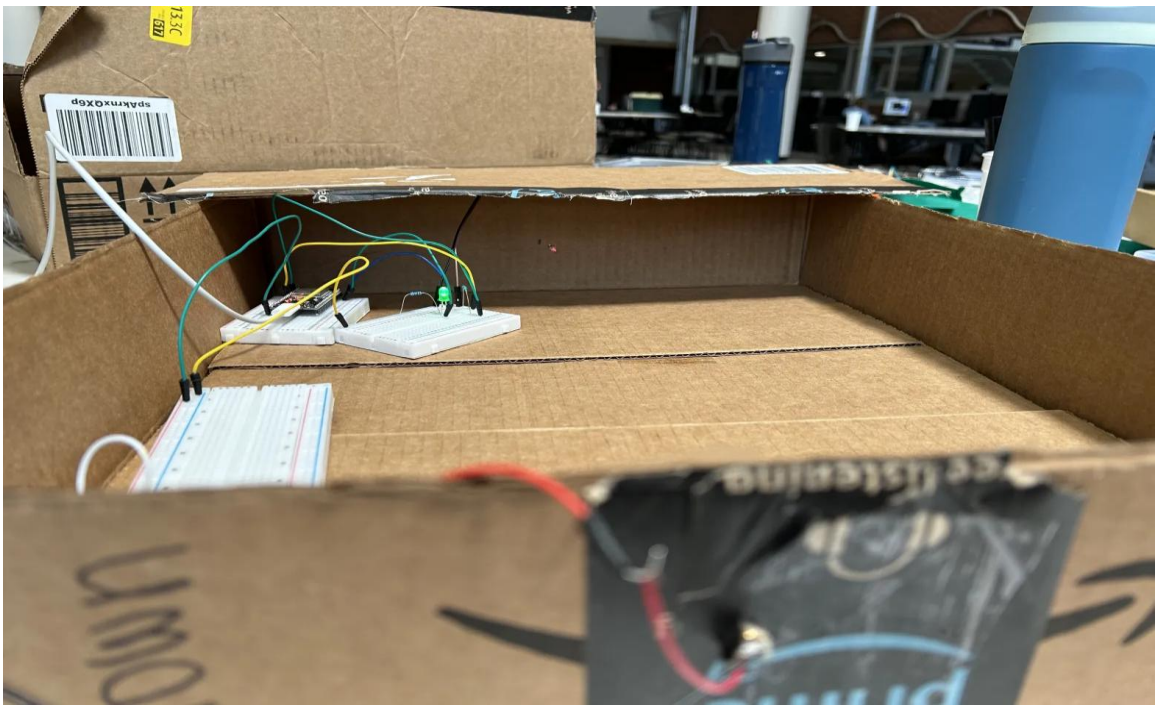

Figure 9.1.1: *Prototype of Laser-Photodiode System*

When testing, the LED remained offline even while the laser beam shined directly at it. This showed how the sensor was calibrated correctly and wouldn't incorrectly signal a false positive. When a hand passed through the beam's direction, the photocell detected a drop in responsivity and signaled back to the MCU. This was shown in the LED lighting up during these incidents. While very primitive, this prototype shows that a core feature of our project is working, and its related components can communicate well. Additionally, while testing, we believed that if we could calibrate the sensor upon operation, we wouldn't have to worry about its range of detection being off during operation. While intended to be more of a stationary instrument, Aqua Strings' size and weight allow the user to move the device if they desire. This means that the sensor within the device might be exposed to an environment with different light levels. For instance, it could be calibrated to a lab room's dim lighting conditions and then be moved to an open conference room with bright lights. To ensure the sensors could work in any environment, we started experimenting with a programming algorithm that could adapt to different light conditions. First, it would take in the ambient light of the room before the laser diode activates and record that light value. Then, the laser diode would activate, and the sensor would receive the ambient light in addition to laser diode's beam and record that second value. The program could then average these two light values together and produce a threshold for detection. If the detected light remains larger (brighter) than this average, that means the laser beam is still hitting the photodiode. If the detected light drops (dims) below the average, then that means the laser has been interrupted and the sensor is only detecting light ambiently. This algorithm would allow for the harp to adjust to new environments instead of being statically coded to one lighting condition. As we progress further through testing and assembly, this algorithm can be tweaked to ensure its working most optimally.

Additionally, we wanted to test how the audio system would trigger upon the laser being tripped. Since we were running both the prototype for the laser system and the prototype for the audio system at the same time, we decided to substitute the amplifier and speakers with a simple passive buzzer. This allowed us to keep the prototype for the audio system operating while only modifying this system slightly. The logic of the passive buzzer is parallel with its substitute (passive speakers). Similar to LED, the photodiode was able to send a signal back to the MCU, which sent a signal to the passive buzzer's corresponding output pin. The result was a high-pitched tone emanating from the buzzer as the laser was tripped. The amplifier and speakers will achieve a similar result, albeit through different pins and utilizing the ESP32's built-in DAC. One effect we noticed while testing the audio system alongside the laser detection was that the tone would end too abruptly. If the user wished to string the same laser multiple times in quick succession, the sound was somewhat choppy. This wasn't a result of latency or reaction time. The system was working exactly as designed: buzzer beeps when light level drops and is silent when the sensor's reading normally. The LED backs up this observation, as it displayed the same results in parallel. The issue was that the system was working too well. A standard harp has some resonance to its notes. The tones it produces stays in the air for a little as the harp's string oscillates back

and forth. We decided that this system was too effective in its response time, and added in a small delay so the buzzer held its note for about a half second after the interruption ended. This greatly improved the effect of strumming the harp, as now the note held for a small amount after playing.

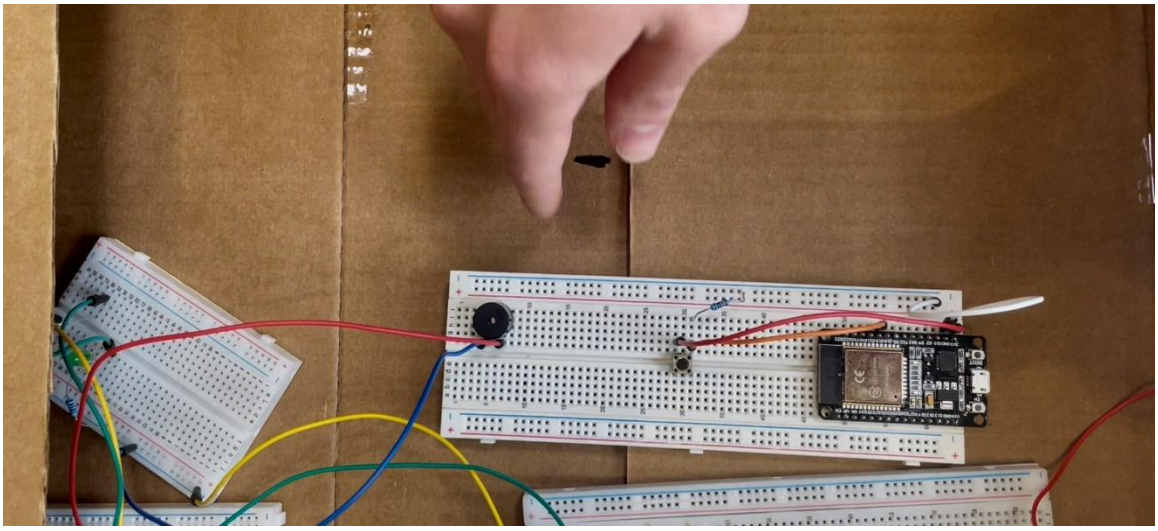

Figure 9.1.2: *Prototype of Audio Triggering with Laser-Photodiode System*

9.1.3 Speaker and Amplifier

For the project's music element to function like a proper instrument, the audio system needs to be both clear and loud. To achieve this effect, our prototype is running two passive speakers driven by an audio amplifier. This amplifier, a PAM8403, needs to take in at least 5V to operate, which is provided from the ESP32. For the iterations beyond the prototype, this voltage will come from our voltage regulator's 5V output. The amplifier takes an analog signal from the ESP32 and passes it along to the speakers. It features a dual channel design; thus, it can operate two speakers at once along the left and right channels. These channels are plugged into the two GPIO pins that can leverage the ESP32's internal DAC, being pins 25 and 26. We ran a simple program that played two different frequencies along the two channels. Both speakers received this input through the amplifier and were able to play these tones simultaneously. This amplifier also featured a potentiometer that acted as a volume knob. This feature would be a nice addition to our custom PCB design, which intends to replace this board in later iterations. The prototype testing involved powering the amplifier board with its 5V input from the ESP32, outputting this power to the speakers, and connecting the internal DAC from the Devkit to the speakers.

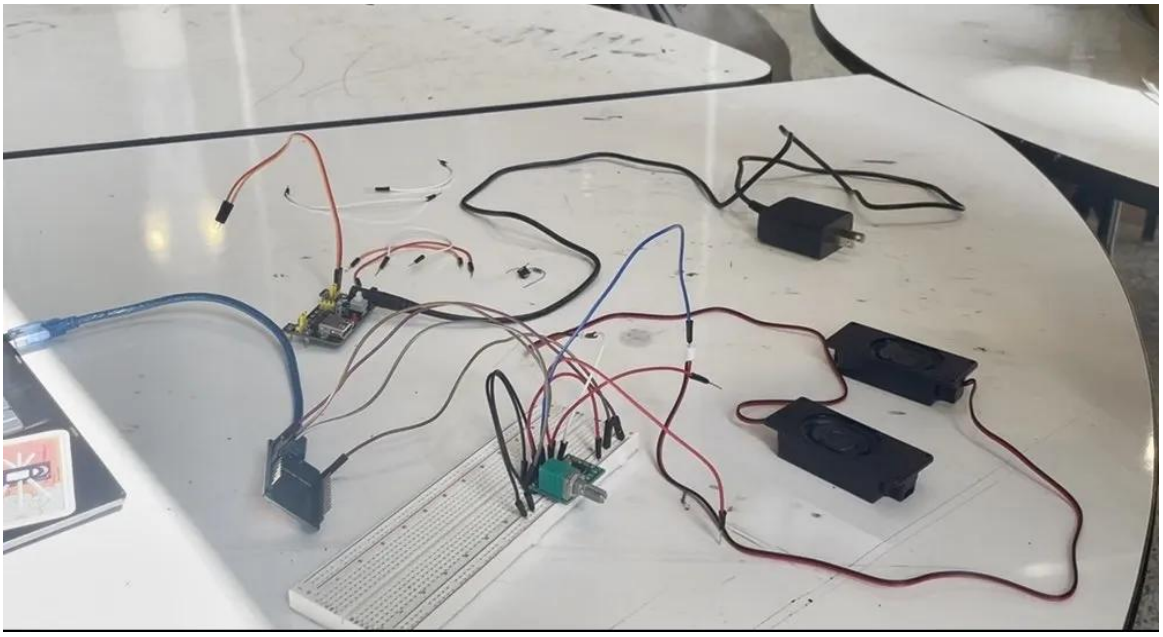

Figure 9.1.3: *Prototype of Speaker System with Amplifier*

While testing, we noticed that the speaker's output was quieter than we anticipated. While acceptable, the loudest output from these speakers might be hard to hear in conjunction with the noise from the water pump and water columns. We had used passive speakers rated at three watts and eight ohms for this prototype. Upon further research, we realized that this was pushing the PAM8403 to its limits. While still functional, the speakers running in dual mode were only receiving about half of its rated wattage. Further iterations might utilize the same style of speakers at a rating of half the resistance (three watts, four ohms). This is a very easy adjustment, as these speakers can be disconnected and reconnected from the design at will. This change would allow a louder sound at a small compromise of audio quality. As we proceed with further testing, this change will be considered if volume continues to be an issue.

9.1.4 Water Pump

The water pump was the most straight-forward of these systems to test. The pump we are using features a variable speed dial that allows us to modify the rate at which water is pumped. It also features a built-in safety mechanism that shuts the pump off in the event that the lower reservoir is pumped empty. The purpose of our prototype system was to test the water pump's maximum pump rate in gallons per minute. We timed the pump's ability to fill an empty container up to a one-gallon mark, which it had reached at about 25 seconds. This meant that the pump could output more than two gallons per minute, which more than satisfied our requirement for this component. The addition of the variable speed dial will allow us to modify this flow rate as needed. In further iterations of the project, if we determine that the pump is overturned and pumping water faster than it can descend, we have the option to compensate and lower this speed. The prototype was a simple test of the component and its capabilities.

Figure 9.1.4: *Prototype of Water Pump System*

9.2 Plan for Senior Design 2

Our time period for Senior Design 2 will be shorter than most students who take this course. Compared to the Fall and Spring semester, which provides students about four months of time from the first week to demonstration day, the Summer Semester only gives our group about two and a half months. This loss of approximately four weeks is a major detriment to the project. To compensate, our group will need to proceed with a careful mindset and continuous determination.

To start, we need to prioritize any portion of the project that has a considerable lead time. The PCB is the most obvious offender. Even the quickest PCB retailer has a turnaround time of around one to two weeks from the moment of ordering to delivery. This means that designing, finalizing, approving, and ordering the PCB must happen as early as possible. We cannot afford to have such a critical component potentially be delayed a few weeks before the demonstration day. We have already designed two custom circuit boards: one for the voltage regulator and the second for the MCU, peripherals, and amplifier. These are in their first iteration and need approval before ordering. They will likely need to go through a second round of improvements after they are checked. We anticipate this process will take one to two weeks and leave us with our finalized PCB around mid-May. From there, we immediately order the two boards and have them within our possession before June. Ideally, we also work towards soldering and constructing the boards. Additionally, we also need to have the first prototype of the harp's frame assembled by then. This will allow us to test the lasers and sensors in an assembly before we add water. Building a prototype frame early on will also give us more time to improve the frame as we progress further through the project.

For June, we need to prioritize on getting the dry prototype working. This entails having all of the components that don't deal with water operational, including the power supply PCB, the microcontroller PCB, the lasers, the photodiodes, and the

speakers. This system will operate similarly to a regular laser harp. There will be no working water system at this moment. Excluding the water will allow us to operate on all of the electronics safely and without concern of short-circuiting. Additionally, it will allow us to work in the Senior Design Lab and other laboratories without concern about ruining other projects or the equipment. Once we start testing with water, the likelihood of us being able to use the spaces diminishes due to concerns of water spillage. This will make the wet prototypes more complicated than they already are, which is exactly why we need to ensure that a dry prototype works first. There are many elements that need to be managed with the laser harp before adding water. For instance, the lasers and photodiodes need to be secured and enclosed properly in the frame. The lasers need to be able to reach the sensors through their enclosure in the lower reservoir. The electronics, such as both PCBs, need to be housed in the backside of the frame while still being able to connect to components both above and below it. The frame itself needs to be sturdy enough to support the gallon or two of water that will be weighing in the upper reservoir. At the same time, it needs to be open enough so that the user can comfortably reach the strings and strum the lasers. And when they do, the speakers need to be loud and clear enough so that the user can hear the instrument play. All of these factors, and possibly many more, will need to be ironed out before we can add water. Thus, June will be the month we develop the dry prototype before anything else.

For July, provided that the dry prototype is operational, we begin adding water. We can start with incorporating the pump into the lower reservoir and adding both water level sensors. These components need to interface with the microcontroller, so connections to the PCB can happen at this moment. Before we introduce the element that could destroy all of the electronics, we double-check the areas of the frame to ensure the electronics are sealed where needed. We can start by testing the laser-sensor relationship when water is passing through the lasers. Realistically, this will somewhat have an impact on the laser's ability to reach the sensor (i.e., through refraction or some absorption). However, this can be adjusted within the parameters of the sensor and should not prose too big of an issue. The lasers we are implementing have the capability of shining on surfaces over thirty feet away. It should make a contact with a sensor less than feet away even when passing through a small stream of water. If this is a larger issue that we anticipate, we will adjust the method and rate at which water flows from the upper to lower reservoir, such as shrinking the water column or making it so less water pools below the laser. We can adjust these things because we know that all of these components work when dry. That is why it is so important to test it before adding water. Once the sensors work in the wet prototype, we can test running the pump constantly and ensure water flow remains consistent. If all of these factors are accounted for, Aqua Strings should be ready for demonstration day.

Chapter 10: Administrative Content

This chapter discusses the administrative considerations for Aqua Strings which involve determining a budget for the project and the schedule for the project to be developed in an organized and timely manner. The costs associated with distinct aspects of the project, e.g., water systems, detecting sensors, electronics, and structural materials, are estimated with alternatives to lower costs taken into consideration. In addition, the project timeline of events is established throughout the two semesters of Senior Design 1 (SD1) and Senior Design 2 (SD2) to include the research phase, purchasing components, prototyping, system integration, and final testing. Overall, by developing a budget and timeline, the Aqua String team has established financial and scheduling considerations that take the project from an initial concept to completion and through final demonstration.

10.1 Budget and Financing

Table 10.1.1: *Budget and Financing*

Component	Subsystem	Quantity	Unit Estimated Cost	Total Estimated Cost
Water Pumps	Water System	2	$11.00	$22.00
Water Reservoir (Top and Bottom)	Water System	2	$4.00	$8.00
Flow Control Valves	Water System	8	$5.00	$40.00
Tubing (PVC Pipes)	Water System	10 ft	$3.50 for 10 ft	$3.50
Laser Modules	Detection	8	$8.00 for 10	$8.00
Photodiodes	Detection	8	$6.00 for 10	$6.00
Custom PCB	Electronics	1	$50.00	$50.00
Speaker System	Audio	1	$4.00	$4.00
Solenoid Valves	Water system	8	$13.00	$104.00
Power Supply	Electronics	1	$30.00	$30.00
Water Flow Sensors	Water System	2	$9.00	$18.00
Wooden Frame	Structure	1	$60.00	$60.00
Control Panel	User Control	1	$10.00	$10.00
Miscellaneous Cost	All	1	$50.00	$50.00
Total				$413

Our budget for Aqua Strings will be completely self-funded by the group. The estimated cost is subject to change and will be further adjusted later throughout the actual ordering process due to available suppliers. The total estimated cost for this project is $413, however some components may be able to be sourced from the senior design lab or from cheaper options, reducing our overall expenses.

A key requirement of our project is the Custom PCB, which will integrate the microcontroller, laser sensor system, power distribution, audio output and water flow control logic. This PCB will handle the signal processing for detecting when the water strings are played, playing the correct note audio output. The estimated cost of designing and the fabrication of the PCB is $50, but this could change based on design, complexity, manufacturer, and component selection.

Our project will require a 12V/5V power supply, capable of powering the electronic sensors, our custom PCB, and the water pumps. To retain the portability of the harp we will be using a LiPo battery system, this will allow for our harp to run on battery power or be directly plugged into the wall. The estimated cost for the power supply and battery system is $30.

The frame of our harp will be constructed out of wood, we chose this material for its affordability, ease of modifying and to fit the theme as wood is what many traditional harps are made of. The wooden frame for the harp along with any fasteners or adhesive to keep the frame together is estimated to be $60. Additional miscellaneous costs such as any wiring, connectors or tools are estimated at $50.

Many of the components were estimated based on current market research, the final prices may vary when we go to purchase our components. Certain components may also be found from alternative sources such as 3D printing or preowned materials, also possibly lowering our costs.

10.2 Project Milestones for SD1 and SD2

Our Aqua Strings project is structured over two semesters Senior design 1 in the Spring and Senior Design 2 in the Summer. Senior Design 1 focuses on research, documentation, and planning, while Senior Design 2 spends its semester prototyping, testing, finalizing the design and assembly. Some of the key milestones of Senior Design 1 include the completion of major documentation, initial component selection and dry prototyping to ensure the systems' functionality. Senior Design 2 is dedicated to system integration including reservoir construction, PCB fabrication, and wet testing the laser dedication system. The final stage will include system optimization, repeated testing, and a final live demonstration.

Table 10.2.1: *Senior Design 1 Milestones*

Milestone	Week #	Due Date	Description
Complete Divide and Conquer Document	Week 4	January 24, 2024	First 10 pages of our Divide and Conquer Report
Order Electronics	Week 6	N/A	Ordering our essential electronic components for testing compatibility and initial prototyping
Complete 60-Page Mid-Report	Week 10	March 24, 2024	First half of our final 120-page report
Build Single Laser, Dry Prototype	Week 11	N/A	Constructing and testing a single laser-based detection system without introducing water
Determine Components for Custom PCB	Week 12	N/A	Finalize needed components and designing the layout of our custom PCB
Complete 120-Page Final Report	Week 16	April 22, 2024	final documentation detailing our research, system design and initial testing

Table 10.2.2: *Senior Design 2 Milestones*

Milestone	Week #	Due Date	Description
Build Test Reservoir Enclosure	Week 1	TBD	Construct the enclosure for the water reservoir and pump system
Complete PCB Design for Print	Week 3	TBD	Finalize and prepare the custom PCB

			layout for manufacturing
Test Single Laser, Wet Prototype	Week 4	TBD	Evaluate our laser-based detection system with water in the system
Test Multiple Laser, Wet Prototype	Week 6	TBD	Expand our testing to multiple laser water strings simultaneously
Finalize Reservoir and Pump System	Week 8	TBD	Ensure optimal water flow for consistent operation
Perform Final System Tests	Week 10	TBD	Conduct full-scale testing and debugging of the whole water harp system
Final Live Demo Presentation	Week 12	TBD	Showcase our completed project to faculty for final review and grade

10.3 Work Distributions

Throughout the semester, our senior design group has managed to have collaboration and an overall strong team dynamic through our project report writing and mini demo. The tables below highlight the major contributions of each team member. Our work distribution has been balanced, and many of the tasks were able to be completed with the assistance of more than one person. Throughout our semester in Senior Design 1, a large amount of time has been dedicated to research for component selection to have the mini demo video ready and working effectively. Throughout this time, each team member has been diligent in overseeing different parts of project research and thus focusing on a certain area of the report writing. With this being said, maintaining clear communication and seeking reinforcement have been constant themes as well. Each team member has their own set of strengths and experience, which has helped create a smooth process during our prototyping and overall dynamic. Overall, our team will continue to rely on our balanced workflow and work distributions to carry on for our project testing **in Senior Design 2** next semester.

Table 10.3.1: *Work Distribution Layout*

Electrical Engineering	Project Responsibilities
Christopher Brown	Chapter 1: Executive Summary Chapter 3: Sensor Selections / Laminar water Flow Chapter 6: Hardware Design Chapter 8: PCB Design Laser Detection Mini Demo Video

Computer Engineering	Project Responsibilities
Ryan Bori	Chapter 5: Comparison to AI Chapter 7: Software Design Chapter 10: Administrative Content Report Citations Speaker Output Mini Demo Video

Electrical Engineering	Project Responsibilities
Trenton Morris	Chapter 3: MCU/Speaker Selection Chapter 3: Audio Output Chapter 8: PCB Chapter 9: System Testing Power Supply Documentation Component Overview Mini Demo Video Speaker Output Mini Demo Video

Electrical Engineering	Project Responsibilities
Mikaella Penagos	Chapter 2: Prototype Illustrations Chapter 3: Water Pump Selection Chapter 4: Project Standards and Constraints Chapter 10: Administrative Content Chapter 11: Conclusion Water Pump Mini Demo Video

Chapter 11: Conclusion

Our Aqua Strings senior design project is an interactive display that blends music, electricity, and water into one single idea. Our project integrates fluid dynamics, laser sensing, and real-time audio processing, all in an effort in creating and functioning like a musical instrument. With our original inspiration from a traditional harp, our project uses vertically falling streams of water instead of physical strings. The core concept is based on the interruption of laser beams traveling through these water streams; when a user's hand disrupts a stream, a corresponding photodiode sensor detects the change and triggers a sound output. This enables users to interact with the system and produce music.

The objective of the project was to develop a fully functional and responsive prototype featuring eight continuous laminar water streams, each acting as an independent musical "string." The system was designed to ensure minimal delay between user interaction and sound response, with the goal of closely mimicking the responsiveness of a traditional string instrument. In addition, a significant motivation behind Aqua Strings was to provide an inclusive platform for music, particularly for individuals with limited mobility or those who lack experience with other conventional instruments.

From a design standpoint, the Aqua Strings system includes multiple subsystems. The mechanical structure has a lower water reservoir and a 155 GPH submersible pump that supplies water to a PVC pipe. This pipe is drilled to produce eight vertical water streams, each flowing in a funnel that maintains laminar flow and reduces splash. The optical sensing system employs red LED laser modules mounted above each stream, with corresponding photodiodes positioned at the base. The choice of red lasers was based on their improved visibility and performance through water. Upon interruption of the water, the system detects a voltage drop across the photodiode and maps the interaction to an audio output.

An ESP32 microcontroller forms the core of the control system, handling sensor input processing and coordinating real-time sound playback using a DAC and microSD card module. Audio is played through two passive speakers mounted to the frame. The system is powered through an external power supply, rather than a batter. Our frame is also designed with clear separation between wet and dry zones to ensure user safety and component protection. Waterproofing sealants and insulated zones are put in place to mitigate the risk of water damage and maintain compliance with basic IEEE safety and manufacturability standards. Additionally, the inclusion of a manual "mode" switch allows the user to alternate between different audio modes or presets, enhancing the system's versatility.

Throughout the prototyping process, we are addressing a few engineering challenges. One of the most critical was environmental light interference, which affected the accuracy of the photodiodes. To compensate for this, the idea is to calibrate the photodiode to assess the ambient light levels and adjust to improve the performance of movement detection. Another significant challenge involved

power distribution ensuring that the pump, sensors, microcontroller, and speakers could all operate simultaneously without voltage drops or impact in performance. Being careful about component selection and specifications has helped maintain system stability under load.

Extensive prototyping and testing validated the core functionality of each subsystem. The pump met flow rate requirements for consistent laminar streams, while the laser-photodiode pairs demonstrated reliable gesture detection in controlled conditions. Speaker output was successfully mapped to individual stream interactions using the ESP32 platform and Arduino C++ framework. These results affirm that the Aqua Strings system can function effectively as an interactive musical interface. The final hardware implementation reflects careful consideration of manufacturability. All major components that are being utilized, such as pumps, sensors, power supplies, and electronics, are accessible and replaceable. The system is enclosed and designed with safety as a primary concern. From an engineering perspective, Aqua Strings represents a successful integration of mechanical, electrical, and software subsystems into a unified product.

Moving forward, several opportunities exist for refinement and expansion. Future versions of Aqua Strings may benefit from the addition of RGB LED lighting synchronized to user interaction, wireless audio output via Bluetooth, or integration with MIDI controllers for digital music composition. Further improvements to the photodiode housing and laser alignment mechanisms could enhance sensing reliability in varying environmental conditions. Additionally, conducting more comprehensive user testing, particularly with individuals with disabilities, and this will provide valuable feedback for improving accessibility and user experience. Overall, Aqua Strings demonstrates a successful application of electrical and computer engineering principles to create a musical interface. Our senior design experience has practiced and valued the use of research, prototyping, system integration, and overall design. With further development, Aqua Strings has a high potential to serve in the market and musical applications.

Appendix A: References

[1] Teledyne ISCO. "LaserFlow Non-contact Velocity Sensor." Teledyne ISCO. https://www.teledyneisco.com/water-and-wastewater/laserflow

[2] Dancer Design. "Vesiharp Musical Fountain." DancerDesign.co.uk. https://www.dancerdesign.co.uk/vesiharp-musical-fountain.html

[3] The Kid Should See This. "The Waterphone, an inharmonic, otherworldly instrument." TheKidShouldSeeThis.com. https://thekidshouldseethis.com/post/the-waterphone

[4] Laser Spectacles. "The Laser Harp." LaserSpectacles.com. https://www.laserspectacles.com/resources/the-laser-harp/

[5] Cascade. "Thread Fountain." Cascade.it. https://www.cascade.it/en/prodotto/thread-fountain/

[6] Instructables. "Arduino Laser Harp." Instructables.com. https://www.instructables.com/Arduino-Laser-Harp-1/

[7] Instructables. "Quick Arduino MIDI Laser Harp." Instructables.com. https://www.instructables.com/Quick-Arduino-MIDI-Laser-Harp/

[8] Kingston Technology. "Mineral Oil Cooling in PC Builds." Kingston Technology Blog. https://www.kingston.com/en/blog/gaming/mineral-oil-cooling-pc-builds

[9] RealPars. "Capacitive Sensor Explained | Different Types and Applications." RealPars.com. https://www.realpars.com/capacitive-sensor/

[10] MaxBotix. "What Is an Ultrasonic Sensor?" MaxBotix.com. https://www.maxbotix.com/articles/how-ultrasonic-sensors-work.htm

[11] OpenStax. "14.1 Fluids, Density, and Pressure." University Physics Volume 1. OpenStax.

[12] Fresh Water Systems. "What Is a Water Pump and How Is It Used?" Fresh Water Systems Blog. https://www.freshwatersystems.com/blogs/blog/what-is-a-water-pump-and-how-is-it-used

[13] Production Technology. "Updated List API ISO Standards ESP." Production-Technology.org. https://production-technology.org/updated-list-api-iso-standards-esp/

[14] IPC. "IPC Standards for PCBs." MCLPCB Blog. https://www.mclpcb.com/blog/ipc-standards-for-pcbs/

[15] IEEE. "IEEE Standards." IEEE Future Networks. https://futurenetworks.ieee.org/standards

[16] MPE Electronics. "What Are the Environmental Impacts When Manufacturing Electronic Products?" MPE Electronics Blog, 2024. https://www.mpe-

electronics.co.uk/2024/07/09/what-are-the-environmental-impacts-when-manufacturing-electronic-products

[17] SketchUp. "Sketchup 3D Modeling Software." SketchUp.com. https://www.sketchup.com/en

[18] OpenAI. “Introducing GPT-4 and GPT-o Series.” OpenAI, 2024. https://openai.com/gpt-4

[19] Wired. “OpenAI’s GPT-4.1 Is a Breakthrough for Developers and Coding Assistants.” Wired, 2024. https://www.wired.com/story/openai-announces-4-1-ai-model-coding

[20] Khattar, D., Sachdeva, A., Kaur, A., and Sharma, N. “Leveraging Large Language Models for Embedded Software Development.” Journal of Embedded Systems and Applications, vol. 18, no. 2, 2023, pp. 101–118.

[21] NASA. “Engineering Applications of AI: SciTech 2024 White Paper.” NASA Technical Reports Server, 2023. https://ntrs.nasa.gov/api/citations/20230018609/downloads/SciTech2024.pdf

[22] Google DeepMind. “Gemini 2.5 Pro Technical Overview.” Google DeepMind, 2025. https://deepmind.google/technologies/gemini

[23] Blog.Google. “Gemini Model Family Updates: March 2025.” Google AI Blog, 2025. https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/

[24] AI.Google.Dev. “Gemini API: Long Context Documentation.” AI.Google.Dev, 2025. https://ai.google.dev/gemini-api/docs/long-context

[25] Anthropic. “Claude 3 Model Family.” Anthropic, 2025. https://www.anthropic.com/index/claude-3

[26] CoinReporter. “Claude AI – The Ethical AI Powerhouse of 2025.” CoinReporter.io, 2025. https://www.coinreporter.io/2025/03/claude-anthropic-the-ethical-ai-powerhouse-of-2025

[27] Anthropic. “Tracing Thoughts in a Language Model: Interpretability Research.” Anthropic, 2025. https://www.anthropic.com/research/tracing-thoughts-language-model

[28] DeepSeek. “DeepSeek R1 Open-Source AI Model.” DeepSeek, 2025. https://www.deepseek.com

[29] Omdia. “DeepSeek’s Implications for Edge AIoT.” Omdia Tech, 2025. https://omdia.tech.informa.com/om128496/deepseeks-implications-for-edge-aiot

[30] The Verge. “Claude 3.7 Sonnet: Anthropic's Hybrid Reasoning AI.” The Verge, 2025. https://www.theverge.com/2025/03/anthropic-claude-3-7-sonnet-hybrid-model

[31] Diagrams.net. "Block Diagram Software." app.diagrams.net. https://app.diagrams.net/

[32] ZDNet. “How to Use ChatGPT to Write Code—And Debug What It Generates.” ZDNet, 2023. https://www.zdnet.com/article/how-to-use-chatgpt-to-write-code-and-my-favorite-trick-to-debug-what-it-generates

[33] Cheng, Y., Zhang, Q., Sun, L., and Li, H. “AI-Augmented Engineering: A Framework for Integrating Large Language Models in System Design.” IEEE AICAS Conference, 2023.

[34] Vinuesa, R., et al. “The Role of Artificial Intelligence in Achieving the Sustainable Development Goals.” Nature Communications, vol. 11, no. 1, 2020, pp. 233. https://www.nature.com/articles/s41467-019-14108-y

[35] Bubeck, S., et al. “Sparks of Artificial General Intelligence: Early Experiments with GPT-4.” arXiv, 2023. https://arxiv.org/abs/2303.12712

[36] BytePlus. “DeepSeek Offline AI Deployment and Privacy Control.” BytePlus, 2025. https://www.byteplus.com/en/topic/384004

[37] Built In. “DeepSeek R1: China’s Open-Source GPT-4 Competitor, Explained.” Built In, 2025. https://builtin.com/artificial-intelligence/deepseek-ai-model

[38] arXiv. “Prompt Engineering Guide for Developers.” arXiv.org, 2023. https://arxiv.org/abs/2312.13115

[39] Theori. “DeepSeek Security, Privacy, and Governance: Hidden Risks in Open-Source AI.” Theori Blog, 2025. https://blog.theori.io/deepseek-security-privacy-and-governance

[40] Microchip Technology Inc. "LM2576/LM2576HV 52 kHz Simple 3A Buck Regulator." Datasheet, DS20006238A. https://ww1.microchip.com/downloads/en/DeviceDoc/LM2576-52kHz-Simple-3A-Buck-Regulator-DS20006238A.pdf

[41] Diodes Incorporated. "PAM8403 3W Class-D Audio Amplifier." Datasheet. https://www.mouser.com/datasheet/2/115/PAM8403-247318.pdf

[42] Simple Electronics. "Simple PAM8403 Audio Amplifier Circuit." Simple-Circuit.com. https://www.simple-circuit.com/pam8403-stereo-audio-amplifier/

[43] Texas Instruments. "SN74LVC125A Quadruple Bus Buffer Gate With 3-State Outputs." Datasheet, Rev. T.

[44] HowToMechatronics. "Arduino Micro SD Card Module Tutorial – How SD Card Module Works and How to Use It with Arduino." HowToMechatronics.com. https://howtomechatronics.com/tutorials/arduino/arduino-micro-sd-card-module-tutorial/

[45] TensorFlow. “TensorFlow Lite for Microcontrollers.” TensorFlow, 2025. https://www.tensorflow.org/lite/microcontrollers

[46] me-no-dev. “ESPAsyncWebServer.” GitHub, 2025. https://github.com/me-no-dev/ESPAsyncWebServer

[47] Schreibfaul1. “ESP32-audioI2S Library.” GitHub, 2023. https://github.com/schreibfaul1/ESP32-audioI2S

[48] Philhower, Earle. “ESP8266Audio Library.” GitHub, 2017. https://github.com/earlephilhower/ESP8266Audio

[49] WeekendWarrior1. “XT_DAC_Audio: Real-Time Audio Playback on ESP32.” GitHub Mirror, 2019. https://github.com/WeekendWarrior1/XTronical_XT_DAC_Audio_Mirror

[50] Edge Impulse. “Machine Learning on Microcontrollers.” EdgeImpulse.com, 2025. https://www.edgeimpulse.com/

[51] Arduino. “millis() – Arduino Reference.” Arduino.cc, 2025. https://docs.arduino.cc/language-reference/en/functions/time/millis/

[52] Espressif Systems. “FreeRTOS Task Management on ESP32.” Espressif Docs, 2025. https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/system/freertos.html

[53] Espressif Systems. “ESP-IDF I2S API Reference.” Espressif Docs, 2025. https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/peripherals/i2s.html

[54] GitHub Docs. “Understanding the GitHub Flow.” GitHub, 2025. https://docs.github.com/en/get-started/using-github/github-flow

[55] Kinsta. “What Is GitHub? A Beginner’s Introduction to GitHub.” Kinsta, 2023. https://kinsta.com/knowledgebase/what-is-github/

Appendix B: Copyright Requests

PAM 8403: Figure 6.5.2

CB

Christopher Brown

To: legal@diodes.com

Thu 4/17/2025 9:25 PM

Hi Diodes Incorporated,

My name is Christopher Brown, and I am a student at the University of Central Florida studying electrical engineering. I am currently working on my senior design project and we require an amplifier for the set of speakers we are using. We plan to use the PAM 8403, and I came across a graph in the respective datasheet that I would like to include in our report. I saw online it should be okay to include for educational reasons, and it includes "Courtesy of Diodes Incorporated." I will attach a screenshot of the image with the explanation I plan to include. If you could let me know if this is approved, I would appreciate it.

Thank you,
Christopher Brown

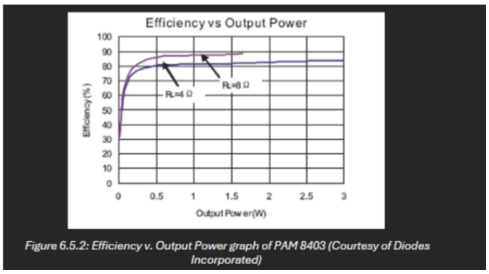

Figure 6.5.2: Efficiency v. Output Power graph of PAM 8403 (Courtesy of Diodes Incorporated)

Appendix C: Copyright Approvals

LM2576: Figure 6.2.1

Use of Copyrighted Materials for Non-Commercial Purposes

Written permission is not required for these two uses of copyrighted material:

- **Personal Use:** If you use Microchip copyrighted material solely for your personal use, you do not need our written permission to use such material. However, distribution or reproduction of such materials and images to others (including posting on a website) does require our written permission.
- **Educational and Non-Profit Use:** If you use Microchip copyrighted material solely for educational (non-profit) purposes falling under the "fair use" exception of the U.S. Copyright Act of 1976, then you do not need Microchip's written permission. For example, our permission is not required when using copyrighted material in:
  - An academic report, thesis or dissertation
  - Classroom handouts
  - A presentation or article that is solely educational in nature (e.g., technical article published in a magazine)

PAM 8403: Figure 6.5.2

Permissions and Conditions

Diodes grants permission to download, print copies, store downloaded files on a computer, and reference this Information in your documents and use the Information internally for the purpose of evaluating Diodes' products, or designing, developing, and manufacturing products that incorporate Diodes' products. You may not modify the Information or further display, reproduce, or distribute the Information outside of your organization without prior written permission from Diodes. You may not, without our permission, "mirror" this Information on your own server, or modify or re-use this information on another system.

Diodes further grants permission to non-profit, educational institutions (e.g., K-12, universities, and community colleges) to download, reproduce, display, and distribute the Information solely for use in the physical or virtual classroom. This permission is conditioned on not modifying the Information, retaining all copyright notices, and including on all reproduced Information the following credit line: "Courtesy of Diodes Incorporated." Please send us a note describing your use of the Information under the permission granted in this paragraph according to the request for permission explained below.

Any permissions, including those granted herein, immediately terminate if the use of this Information under such permissions adversely affects the reputation or business interests of Diodes, as determined solely by Diodes. These permissions also immediately terminate if you breach any of the conditions stated herein. Upon termination, you agree to delete and/or destroy all copies (including partial copies) of any Information.

Appendix D: Data Sheet

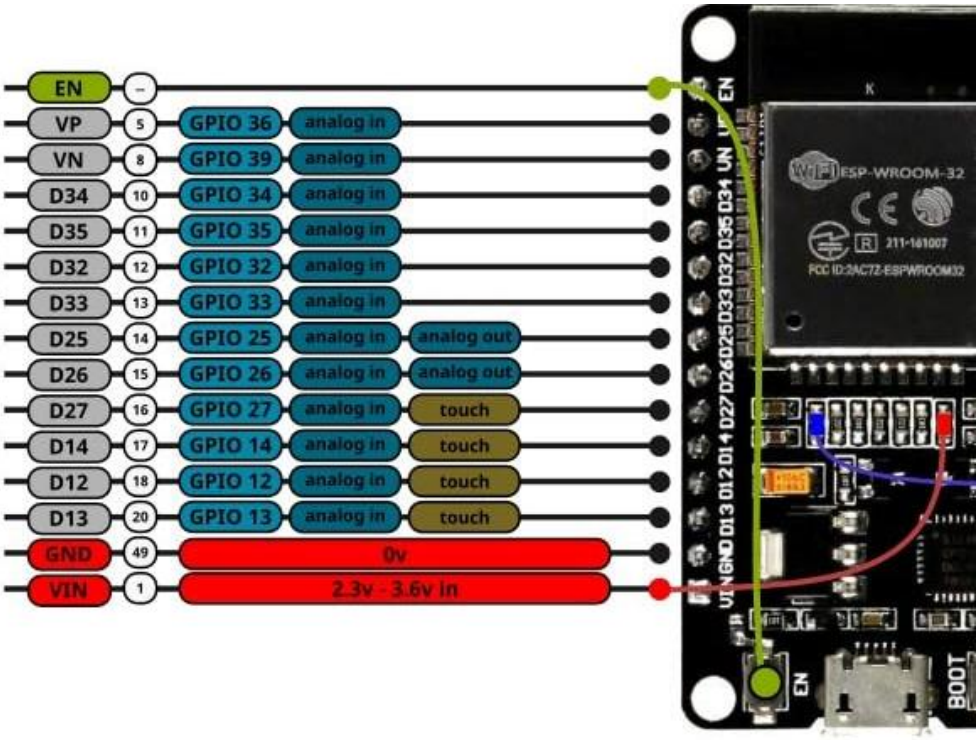

Figure D1: ESP32 WROOM Left Side Pinouts

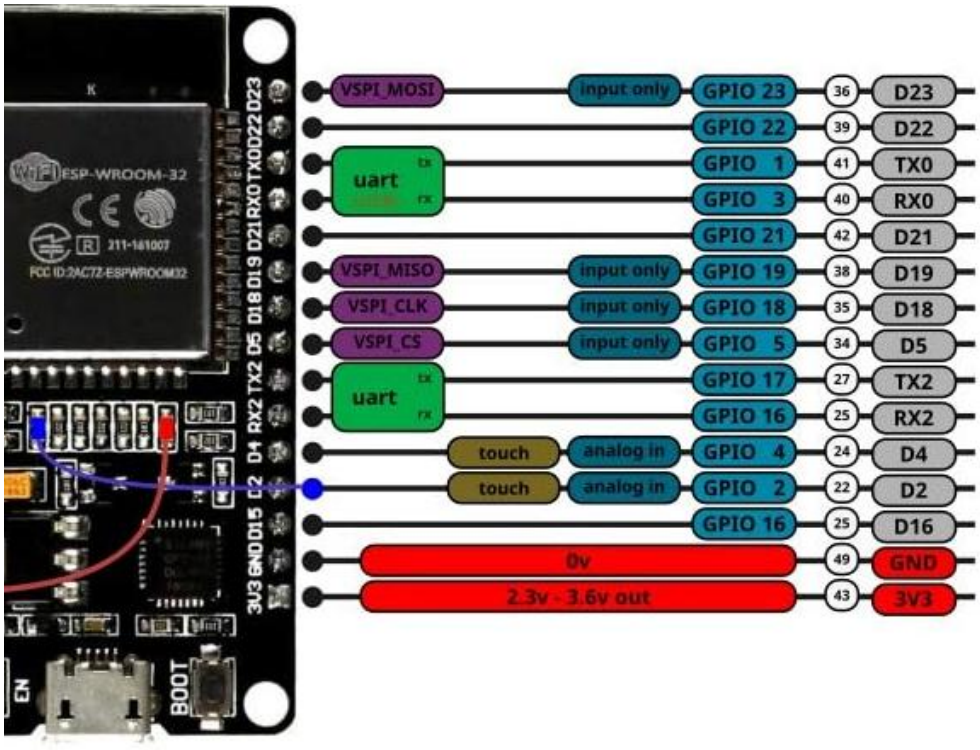

Figure D2: ESP32 WROOM Right Side Pinouts

Appendix E: Software Code

```
const int buzzerPin = 18;        // Pin connected to the buzzer
const int photodiodePin = 34;    // Pin connected to the photodiode (use any available ADC pin)
const int ledPin = 23;           // Pin connected to the LED
int photodiodeValue = 0;         // Variable to store the photodiode reading
int threshold = 2000;            // Threshold to detect beam break (calibrate based on your setup)

void setup() {
  // Initialize the LED pin as an output
  pinMode(ledPin, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);     // Set the buzzer pin as output
  // Initialize serial communication for debugging
  Serial.begin(115200);
}

void loop() {
  // Read the value from the photodiode (analog input)
  photodiodeValue = analogRead(photodiodePin);

  // Print the value for debugging
  Serial.println(photodiodeValue);

  // Check if the photodiode value is below the threshold (beam is broken)
  if (photodiodeValue < threshold) {
    // Turn on the LED if the beam is broken
    digitalWrite(ledPin, HIGH);
    tone(BUZZER_PIN, 1000);  // Play 1kHz tone when button is pressed
  } else {
    // Turn off the LED if the beam is not broken
    digitalWrite(ledPin, LOW);
    noTone(BUZZER_PIN);      // Stop tone when button is not pressed
  }

  // Small delay for stability
  delay(50);
}
```

```
// Configuration

 #include <esp32-hal-ledc.h>
const int pwmChannel = 0;      // LEDC channel 0
const int pwmFreq = 1000;      // Frequency in Hz (e.g., 1000 Hz = 1kHz tone)
const int pwmResolution = 8;   // PWM resolution (8 bits = 0-255 duty cycle)
const int audioPin = 26;       // GPIO 26 on ESP32

void setup() {
  // Set up PWM on the specified pin
  /*ledcAttach(audioPin, pwmFreq, pwmResolution);

  // Start playing a tone by setting duty cycle to 50% (square wave)
  ledcWriteTone(pwmChannel, 128); // 128 is 50% of 8-bit (0-255)

  delay(10000);

  ledcWriteTone(pwmChannel, 0);
  ledcWriteTone(audioPin, 128);*/
  //tone(26, 1000);
}

void loop() {
  // The tone plays continuously
```

```
  tone(26, 1000, 1000);
  delay(10000);
  // Add code here to change the tone or stop it if needed
}

```

```
const int photodiodePin = 34;   // Pin connected to the photodiode (use any available ADC pin)
const int ledPin = 23;          // Pin connected to the LED
int photodiodeValue = 0;        // Variable to store the photodiode reading
int threshold = 2000;           // Threshold to detect beam break (calibrate based on your setup)

void setup() {
  // Initialize the LED pin as an output
  pinMode(ledPin, OUTPUT);

  // Initialize serial communication for debugging
  Serial.begin(115200);
}

void loop() {
  // Read the value from the photodiode (analog input)
  photodiodeValue = analogRead(photodiodePin);

  // Print the value for debugging
  Serial.println(photodiodeValue);

  // Check if the photodiode value is below the threshold (beam is broken)
  if (photodiodeValue < threshold) {
    // Turn on the LED if the beam is broken
    digitalWrite(ledPin, HIGH);
  } else {
    // Turn off the LED if the beam is not broken
    digitalWrite(ledPin, LOW);
  }

  // Small delay for stability
  delay(50);
}

```