Dancing Water Display

An Audiovisual Spectrum Analyzer

Fall 2015 – Spring 2016
Group 37

Katie Corini
Joshua Fabian
Esha Hassan
Timothy Le

Table of Contents

Table of Figures

Table of Tables

1.0 Executive Summary

This senior design documentation will outline the process for Group 37’s project. The paper will lay down the preliminary objective of the project and include a description of the goals and objectives, requirements and specifications, research of existing technologies, design details, prototype and testing. It will also document the selection processes, constraints, and possible errors that the group might run into and how they can be solved.

The project was chosen by a group of four engineering students at the University of Central Florida, three of which are studying electrical engineering and one studying computer engineering. The intention is to have a project that has a mix of both hardware and software experience, as well as one that is enjoyable. Projects involving music or light emitting diodes (LEDs) have been very popular in the past years. Most people, if not all, find pleasure in listening to music, and it would be more entertaining if there could be a visual of it. The group has collectively decided to put together an audiovisual spectrum analyzer in the style of a histogram based on a water pump array. It also includes LEDs to create an appealing show that corresponds with music. Not only does this entail significant hardware design, it is also comprised of a considerable amount of software and coding. Furthermore, since the instrument is intended to be portable, the size and weight needs to be as minimized as much as possible so that it is easy to transport. This requires a great amount of planning and calculations that are within the available resources.

The purpose of the device is for it to be user friendly by taking an audio signal, which it will receive when plugged into a phone, laptop, tablet, mp3 player, etc. and will be outputted by a speaker, as well as have the frequency domain displayed by streams of water which will be pumped by 16 water jets. It is also desired that the users are able to control the lighting and pumps wirelessly so a mobile application will be added on at least one smartphone brand. This will give a custom exhibit that can be flaunted in a party or home setting and is definitely bound to draw attention.

To reach the goals of the project, much research has been done by the group on every necessary piece of hardware, programming languages, structural components, and background studies which will be the basis of the senior design project. Because of the variation in design, the project will be worked on beginning from the inside and working outward. After gathering the fundamental requirements, the process for putting everything together for each individual phase should be facilitated according to the agenda made by the group. Communication and analysis for each phase must be thorough, since failure to do so can affect the other stages negatively and put the group behind schedule.

2.0 Project Description

This project uses a standard 3.5 mm headphone jack to input an audio signal into the custom spectrum analyzer. The analog signal is converted to a digital signal, which is filtered before a Fast Fourier Transform is used to find the frequency magnitudes. The signal will be converted to analog again and amplified to power the motors. The magnitudes are then used to determine how the water will react to the inputted signal. Each water jet is assigned to a specific frequency band, and there will be around 16 water jets. The strength of the water jets will vary based on the strength of their corresponding frequency band. The result will be a physical representation of a spectrum analyzer. In addition, there is a strip of LEDs for each water jet to add color to the exhibit which will also pulse to the music. Additionally, the power supply for the system will be designed by the team.

2.1 Motivation

The motivation of this project is to gain an understanding of digital signal processing and power supply design, along with mechanical and aesthetic design. The display includes an assortment of design characteristics that appeal to each group member, including development of an Android application, embedded system development, electrical and power supply design, as well as a mechanical aspect. The team was excited by the idea of combining electronics, water, math, and art to provide a beautiful show.

2.2 Goals and Objectives

The project will contain a number of goals and objectives that the group hopes to meet. The main goal is to produce a functioning spectrum analyzer, being a visible representation of an audio input. The water display will be portable, of medium weight, and have a stable structure, secure sealing, and a sleek, professional look. The system will be user friendly and will feature a standard 3.5 mm audio jack for input. The group originally discussed taking an audio input with a microphone, but ultimately decided that the audio signals would be much clearer with a direct input. This would allow for a more accurate collection of data on which to perform calculations, which would lead to more accurate spectral analysis, which means a higher quality water and lights show. Learning is also a big objective, and with this project, the group believes that they will acquire a vast range of hands-on skills by applying knowledge from the courses taken in school, as well as build upon new areas of focus, which are mentioned in the Skills Acquired section. The group also hopes to complete the project within the given budget and in a timely manner.

2.3 Requirements and Specifications

The design layout is shown below. The power supply unit will be receiving its power from an outlet on the wall which will deliver 115 volts and the main circuit board will be powered by a switching regulator. The audio signal from the music-playing device will go through a low pass filter that will screen out frequencies higher than 4kHz, since most music frequencies do not surpass that value. This signal will pass through the analog-to-digital converter so that it can be properly sampled, and then converted back from digital to analog. Furthermore, once the Fast Fourier Transform is completed, both the LED array and water pumps should be functioning as a visual display of the audio signal. The third output in this device is the speakers from which the music will be heard.

The wireless device that will be used will be capable of controlling the LED and pump settings on a mobile application. The wireless signals will be sent to the microprocessor via Bluetooth, which will be converted to analog signals that will run the array and drive the current to operate the water pumps. Lastly there will be a kill switch connected directly to the microprocessor that will have the ability to shut down the device without killing the power supply. Figure 2.3 shows the block diagram of the project.

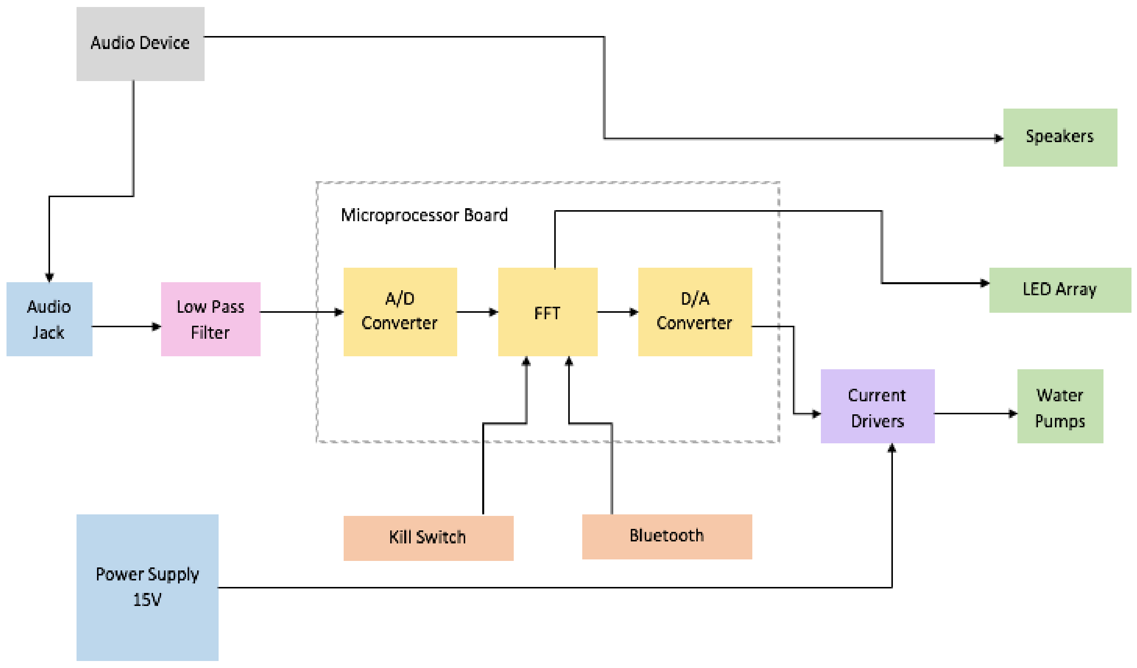


Figure 2.3: Top Level Block Diagram

This project consists of essentially a tank with related circuitry, acrylic casing, and a separate compartment to hold the electrical components. Since Group 37 is being financed by Boeing/Leidos, certain restrictions must be placed so that expenses do not go over the allotted amount of $900. It is important to recognize the physical needs and to define all parameters of the project as a group. The sections below outline the requirements and specifications for the Dancing Water Spectrum Analyzer.

2.3.1 Display

There are certain requirements that the structure itself has to meet in terms of its constructing materials, setting, and physical stipulations. The descriptions for each are mentioned in the following sections.

2.3.1.1 Environment

The display needs to function in the proper room temperature and appropriate environment so that it does not get damaged or risk malfunction. To have the suitable setting, it needs to be in an ambient lab environment where the temperature is between 60 to 90 degrees Fahrenheit. The platform on which the display will be placed also needs to be a flat, stationary surface with zero shock and no vibrations.

2.3.1.2 Portability

One of the main goals is for the device to be portable. It must be lightweight enough to be moved by one person. Most of the weight will be contributed by the wooden reservoir, acrylic casing, and wooden lids. To minimize complications, the group would like to make it as light and as small as possible. To achieve these outcomes, the following requirements must be met:

- Must be approximately 10 pounds
- Easy to fill and remove water rather than keeping gallons of water in the bin which would make transportation difficult and risk damage to the display
- Entire display must be within dimensions of 38” x 10.25” x 20.5”
- Must include a power supply box to retain all the electrical components

2.3.1.3 Water Pumps

Whether one big water pump is used or multiple, smaller, individual ones, they need to represent a frequency spectrum one way or another. Since this the central focus of the display, there are stricter specifications:

- 16 units
- Variable heights for water streams
- Specific frequency range for each division
- Low power requirement of up to 5 watts
- Consistent flow rate of approximately 53GPH (or 240 l/h)

- Weight should not exceed 70 grams
- Positioned so that water shoots vertically

2.3.1.4 Wooden Pieces

A wooden piece will be used as a mounting surface for the water pumps. Additionally, particle board will be used to construct the water reservoir, reservoir lid, and a box to house the LEDs. Doing this entails:

- 2x4 wood with dimensions 26.5” x 1.5” for water pump beam
- Water reservoir with dimensions of 37” x 9.25” x 6”
- Reservoir lid with dimensions of 38” x 10.25” x 1”
- LED box dimensions of 34” x 4.5” x 3”

2.3.1.5 Acrylic Tank

The plexiglass will be used to house the fountain. There will be an opening at the top of the lid of the plastic bin about as long and narrow as the wooden mount so that the tank can be placed on top. To ensure optimal protection, the following is required:

- Dimensions of 12” x 33” x 3.5”
- Plexiglass is clear enough for the water jets to be visible
- Plexiglass is clear enough for LEDs to shine through
- Secure sealing so that water does not escape

2.3.1.6 LEDs

The LEDs will be incorporated into the display to provide an appealing show for viewers. They come in single colors or multicolored. There will be six LED for each water pump and it is intended that they produce an array of colors. This necessitates:

- 96 illumination LEDs (32 each of red, blue, and green)
- 5mm size
- Rounded
- Luminous intensity of 8000 mcd
- Ability to pulse accordingly with audio input

2.3.1.7 Speakers

Ordinary speakers that will be either bought or borrowed will be used for the audio output. They should have a peak power between 6 to 10 watts so that they are loud enough to fill a room with sound, and be capable of having the volume be adjustable by the user.

2.3.2 Hardware

The hardware is the main control source that is going to drive the entire exhibit. All the circuit components of the design entail the DIP package. There are very precise requirements necessary for all the hardware parts, described below, to ensure that everything works exactly as envisioned.

2.3.2.1 Power Supply

The source of the power supply will be a wall outlet, which will have an AC voltage of 115V at 60Hz. An LM317 will be used for voltage regulation. In order to provide enough power to run the main circuit board containing the PIC32 microcontroller, D/A converters, and LEDs, it needs:

- A maximum output voltage of 15 volts
- A maximum output current of 2 amps

LM350 linear regulators will be used to supply power the pumps. They will need:

- A maximum output voltage of 15 volts
- A maximum output current of 3 amps

2.3.2.2 Audio Jack

The audio jack to be used will be a standard 3.5 mm one. This will be able to take input from most, if not all, music-playing devices such as smartphones, iPods, and laptop computers.

2.3.2.3 LED Drivers

The requirements for the LED drivers include:

- Provide constant current
- Receive PWM signal from microprocessor
- Maximum supply voltage of 3.5V

2.3.2.4 Microprocessor

The microcontroller is going to be the powerhouse of the entire spectrum analyzer since it is where all the software codes will be stored and will control the water pumps and LEDs by regulating the power that is being supplied to each component. The microcontroller that is going to be used is PIC32. The hardware part of the microcontroller requires:

- Ability to interact with computer
- Analog to Digital conversion capability
- 16 analog inputs

- 128KB of SRAM memory
- 512KB of Flash memory

2.3.2.5 Analog-to-Digital Converter

An analog-to-digital converter will be used to convert the audio signal from the input into digital format to be sampled. The processor in the PIC32 already has an ADC built into it with:

- 10-bit resolution
- 16 channel input

2.3.2.6 Digital-to-Analog Converter

DACs must be used to send signals to the water pumps after the audio signal has been processed digitally. To do this, 16 converters are needed, one for each pump, meaning that two DACs with 8 outputs each are necessary. The best converter to go with is the LTC1665 Micro-power Octal 8-bit DAC. Other requirements include:

- Supply voltage between 2.7 and 5.5 volts
- 8 output pins
- Supply current of 56 microamps
- Junction temperature of 125 degrees

2.3.2.7 Current Drivers

Current drivers are needed to amplify the voltage to power the water pumps. The current driver device that the team decided to use is the OPA548. The characteristics of this device are:

- Supply voltage between 8 and 60 volts
- Maximum continuous output current of 5 amps
- Input voltage between -0.5 to 5 volts
- Pin count of 7

2.3.3 Software

The software coding is going to be the building blocks of the project. This is going to be implemented by the microprocessor that controls the display.

2.3.3.1 Microcontroller

The software requirements of the microcontroller include:

- Ability to respond to buttons
- Ability to respond to wireless signals

- Programmable in high level languages: C and Java
- Processing aptitude to execute a Fast Fourier Transform on analog audio signal input

2.3.3.2 Application

The group wants to develop a mobile application so that the spectrum analyzer can be controlled by a smartphone. To do this, it needs the following features:

- Wireless communication: Bluetooth
- Android availability
- User-friendly
- Concise code
- Ability to turn device on/off, control LED and pump settings

3.0 Research related to Project Definition

It is important to research relevant products and past assignments in order to discuss research for the project and to stimulate the design procedure. These range from LED music boxes to customized spectrum analyzers and by exploring existing projects, both commercial and hobbyist, it will aid us in constructing a reference in which the design can be roughly based on.

3.1 Existing Similar Consumer Products

There are many products on the market that relate to the project. These vary from basic lab equipment to hobby merchandises whose individual concepts the group is incorporating into their assignment.

3.1.1 Spectrum Analyzer

A spectrum analyzer is an instrument which measures the magnitude of an input signal versus frequency. In other words, it analyzes oscillations into its separate components, especially sound. In 1967, FFT spectrum analyzers were invented following the discovery of Fast Fourier Transforms in 1965. The project is an elementary functional spectrum analyzer, and will use similar algorithms to produce a frequency domain display, with 16 divisions.

3.1.2 Water Speakers

The water speakers consist of a stereo speaker system, a number of streams of water, an equal number of LED lights, and a 3.5 mm stereo audio input. They are usually USB-powered for convenience and portability and are very lightweight. The function of this product is to have water and lights “dance” to music. The project is intended to be a larger scale of this product, and along with the concept of the spectrum analyzer, it will have an audiovisual display of water dancing to a frequency response.

3.2 Existing Student Projects

There are many existing student projects involving LEDs, water pumps, or music. These have ranged from LED game cubes to fountains to equalizers. The projects that were looked at are both spectrum analyzers, one using an LCD screen and the other using water jets.

3.2.1 Project 1

One project that was looked into was done by a group of students at Cornell University in 2012 for their final project. The team put together an audio spectrum analyzer which also displays a histogram-style image of audio signal frequencies up to 4 kilohertz on any ordinary television screen that supports a set resolution of a minimum of 160x200. In order to do this, they used two microcontrollers, one for audio data acquisition and the other for visualization processing and video data transfer. The signal sent through the audio jack is intended to be amplified and filtered, followed by the execution of a Fast Fourier Transform. This device is also interactive by allowing users to connect any sound-producing device via a male-to-male audio cable and letting them select assorted display options.

3.2.2 Project 2

The second project that was looked at is the closest to what the group hopes to accomplish. This project was achieved by a group of electrical and computer engineering students at the Georgia Institute of Technology in 2004. It is operated by 16 water pumps to produce a spectrum analyzer that is interfaced with concurrent audio. The input comes from MP3, a wave input, and a microphone, whereas the output is the array of water jets that display the spectral frequencies. This group used MATLAB GUI to capture the audio from a device to calculate a Fast Fourier Transform and send further data to an FPGA board, which is acknowledged by the Altera board. The Darlington resistor array then controls the water pumps according to the FFT amplitudes. This display also included a chopper circuit which was used to turn the pumps on and off for variable current amplification, among additional characteristics which vary the height of the water.

3.3 Embedded Software

The microprocessor is the brain of the spectrum analyzer. It has many responsibilities, including:

- Performing a Fast Fourier Transform (FFT) on an inputted analog audio signal
- Analyzing the magnitudes of the different frequencies in the input
- Outputting control signals to the water pumps to regulate the power being supplied to each
- Outputting control signals to the various LEDs in the display
- Responding to button presses received from the physical buttons
- Responding to wireless signals
- Receiving a digital audio input signal wirelessly

After carefully considering these requirements it was decided that a PIC32 microcontroller would have enough processing power to serve these needs while being affordable and small enough to fit in a compact circuit board attached to the display. Analog input signals will be fed into the microcontroller via a standard 3.5 mm audio jack. The digital audio signals will be received wirelessly via Bluetooth from a nearby Bluetooth enabled device running the accompanying application.

3.3.1 Languages

Choosing a programming language to use with the microcontroller was a major factor in determining how delightful it would be to program the device as well as how efficiently it would perform once programmed. Due to resource constraints, the developers decided early on that it would be too costly to run an operating system or a Java Virtual Machine on the controller. On one side, a high level language such as C could be used to make the programming process smoother but at the risk of less than optimal performance of the finished product. Coding in assembly, however, could result in improved performance, but it would make the programming process considerably more painful.

Another consideration to make when comparing languages is what kind of programming paradigm would best fit the project. C is set in a procedural, imperative paradigm that is intuitive to write and more familiar to most programmers. Verilog, on the other hand, is a declarative dataflow language that, through judicious use of relevant libraries, could reduce the time required to program the controller. A brief overview of each option is outlined below.

3.3.1.1 C

What is true for many programmers worldwide is also true for the development group: upon hearing the term “programming language”, the first thing they think of is “C”. C is a very flexible and extremely popular language that is used on everything from supercomputers to tiny embedded systems. It is also the language with which the group has had the most prior experience, so it seemed like choosing C would lead to the smoothest programming experience.

C's popularity is one of its biggest advantages. Free tutorials for all kinds of applications exist all over the internet for C and many supporting libraries have been written, making learning and using the language relatively easy and efficient. There are also several IDEs available for use, most of which include convenience features such as code completion and refactoring.

C was designed to use operations that translate efficiently to assembly instructions, so while it would save a lot of time in programming the controller, it would probably not lose too much performance compared to manually programming in assembly. Also, if during testing the group decides that more efficient code is required, the Gnu C Compiler (GCC) will allow the insertion of snippets of assembly code into the C file that will not be altered during compilation. So the group has the option to manually optimize some functions in assembly, if it is deemed necessary, while keeping most of the convenience of programming with C.

3.3.1.2 Verilog

Verilog is a hardware description language (HDL) that is a popular choice for embedded systems programmers. Verilog can run quickly and extremely efficiently on systems that don't have many resources to spare, so it is a natural candidate for the embedded system like the one that will be operating in the spectrum analyzer.

One of Verilog's most attractive features is its use of non-blocking assignments, which can be used to define a number of actions that should not be taken until the next clock cycle. This is especially useful when describing sequential circuits, because multiple calculations might need to be made before assigning new values to a set of variables. The non-blocking assignment will wait for all relevant evaluations to be completed before assigning the new values, while the blocking assignments that are typical of a procedural language will assign new values as soon as possible. Using exclusively blocking assignments might lead to a slight performance increase, but the non-blocking style can speed up the programming process. Because the assignments wait until the next cycle, instructions can be coded in any order within a block and they will all be executed simultaneously, which can reduce potential data dependency issues. This feature also appealed to the developers because it would enable all of the water pumps in the display array to be updated at the same time, which could make the water show look smoother.

Another useful feature of Verilog is its use of "always clauses". Setting up an always clause allows a programmer to define code that is to be executed whenever a certain variable changes value, which allows a program to be more event-driven. This would be useful for the analyzer as a block of code could be defined to update each water pump that would only need to run if that pump needed updating, which would save a bit of processing power.

3.3.1.3 Assembly

Assembly language programming was another option worth considering. The PIC32 family of microprocessors is based on the MIPS processor with a RISC architecture. Programming in assembly is an interesting option as it would give us more control over CPU cycles and allow for manual optimization. Most of the instructions in the PIC32 instruction set are single-cycle, so there is a lot of potential for writing very elegant and efficient code.

There are also many disadvantages to choosing assembly. For starters, assembly has a relatively steep learning curve, and it can take more time to learn how to perform certain tasks than it would with higher level languages. In addition, the programmers would have to break the program down into tiny pieces before it could be written in assembly. This would essentially mean having almost all of the details of the code worked out before even starting the actual programming. Also, the symbolic nature of assembly code severely reduces readability. This would negatively impact the efficiency with which multiple programmers could contribute to the same code, as well as making debugging and maintenance more difficult even for a single programmer, although heavily commenting the code could help mitigate this effect.

3.3.1.4 Comparison and Decision

Due to time constraints, the first option that was ruled out was programming the microcontroller with assembly code. Although it could potentially lead to a smaller code file that ran more efficiently, the dramatic increase in time and effort that would have to be put into the programming process would not be a prudent use of resources.

The choice between C and Verilog was more difficult to make. Verilog has many features that would be useful for the analyzer, such as non-blocking statement blocks and the “always clauses”. The non-blocking statements would be helpful in synchronizing updates to the water pumps, while the always clauses would help modularize the code. Also, the dataflow nature of Verilog makes sense for the application as all of the processing is done in reaction to input signals, such as the audio input and button presses. Verilog also has some useful IDE options available, some with several built-in code assistance features.

In the end, however, the group decided to use C to program the microcontroller. C’s popularity is just too great of an asset. There is a plethora of resources online to assist with C programming, including tutorial videos, websites that explain the inner workings of the language, and even databases of previously written functions for all kinds of obscure applications. Forums are filled with people requesting and offering help with C, and thousands of questions have been asked and answered.

C is also very compatible with the PIC32 processor that the team planned to use. Microchip, the manufacturer of the PIC32, released an IDE called MPLAB to assist with embedded system development. MPLAB comes bundled with a C compiler but unfortunately does not support Verilog programming. Since the team was eager to use MPLAB to ensure maximum access to all of the tools available for the PIC32, C became the clear choice.

3.4 Application Software

Once the microprocessor is operational, it will need to be controllable. Perhaps the easiest option would be to just include physical buttons on the outside of the display, but the group wanted to go a step further than that. They wanted the users to be able to control the display from a distance. A few options were considered for this, including a remote control that resembled a TV remote, a motion sensor to detect hand movements for commands, and even a module for voice control support.

The group decided the development of a mobile application would be ideal, and that the application should have the following features:

- Availability on at least one major brand of smartphone (Android, Apple, and/or Windows phones)
- An easy-to-use user interface with intuitive controls
- Ability to communicate wirelessly with the spectrum analyzer and send control signals, such as turning the master power on and off, changing the color scheme of the LEDs, etc.
- Lightweight and responsive performance
- Small file size

The application will be transferred directly from the developer's computer to one of the group members' smartphones for the demonstration. By not releasing it on a public application repository, it is unnecessary to introduce security measures as the group will have the only application capable of communicating with the spectrum analyzer.

3.4.1 Platform

Once the group decided on the overall functionality they would like to accomplish with the application, they needed to pick a platform to host it. There were several options available, including a web-based application hosted on a third party server, a web-based application that would be connected to through a LAN, and a mobile application that could communicate with the spectrum analyzer directly. These communication schemes are depicted in Figure 3.4.1.

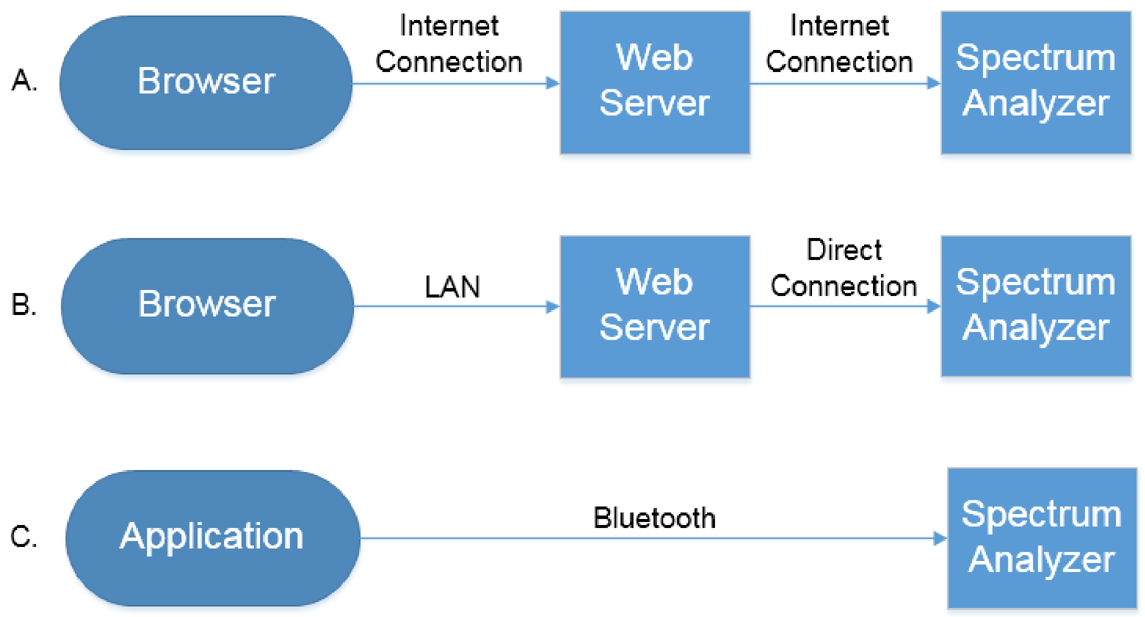


Figure 3.4.1: Comparison of Communication Schemes

A. *Depicts communication through a web server hosted on the internet.*
B. *Depicts communication through a web server located on the spectrum analyzer.*
C. *Depicts communication directly from the application to the spectrum analyzer via Bluetooth.*

From Figure 3.4.1 clearly depicts the differences between three possible methods of communication. Figure 3.4.1 A has the advantage of being accessible anywhere, but requires two internet connections to be set up. Figure 3.4.1 B has the advantage of allowing a direct link between the web server and the spectrum analyzer, but would require hosting the web server locally and setting up a Wi-Fi access point. Figure 3.4.1 C has the advantage of direct communication, eliminating the need for hosting a web server, but has shorter range and requires a Bluetooth-enabled device.

3.4.1.1 Web-Based

A web-based application seemed like a straightforward solution and there would be many advantages to using one. A web-based application can be accessed through a browser, so it does not to be installed. Running through a browser also makes it platform-independent, as it would only need to be written once to be accessible from any computer or phone with a browser and an internet connection. It would also be more centralized; since everyone accesses the same version on the server, future updates would be immediately available to everyone. This can be both good and bad, however, since bugs in those updates will also immediately affect everyone.

Unfortunately, using a web-based application means that the group would have to use a web server to host it, which could either be set up ourselves or by a third party hosting company. They did not want to involve a third party because not only would that introduce a monthly expense, but it would also make the problem of establishing communication between the web server and the spectrum analyzer much harder to solve. When they considered hosting the web server themselves, it was realized that a Raspberry Pi could be placed on the control board of the spectrum analyzer, which would enable seamless communication between the server and the analyzer. Users would still need to be able to connect to the web server, though, so the analyzer would need its own connection to the internet in order to be accessible. Considering the group wants some degree of portability, they decided that a Wi-Fi connection would be preferable to a wired connection, which meant that some sort of Wi-Fi module would have to be installed; although it would still be inaccessible anywhere there wasn't an available Wi-Fi connection. The group then briefly considered the possibility of placing a wireless router on the analyzer itself, which would create its own Wi-Fi access point to allow people to connect to it. This would allow the owner of the display to control it from anywhere, but it was realized that this was a lot of extra work and expense that would have to be put in for a feature that was not really needed in the first place.

Also, a web-based application would be harder to scale up if the analyzer ever hit mass production. Setting it up for the single prototype display would be fairly straightforward, but if there were many users with displays, each would need to somehow isolate their display from everyone else's. This would probably involve setting up some kind of user database, linking a specific display to a user account, and then requiring the user to log in to access their display. This would increase server hosting costs and complexity, as well as make the users go through extra work to enjoy their display.

3.4.1.2 Android

The most popular choice for mobile application development is Android. Android devices have access to the largest app store in the world, the Google Play store, which contains 1.6 million apps available for download. Android is also the most popular mobile phone operating system in the world, with 52.6% of the mobile/tablet operating system market share. Over 80% of all smartphones sold in 2014 were running some version of Android. The numbers do not lie; if you are looking for maximum availability, Android is where you want to be.

Android's popularity is not its only advantage. Choosing to develop an Android application means you get to use Android Studio, a free IDE developed by

Google that includes many fancy features including real-time app rendering, refactoring, code completion, and a drag-and-drop UI editor with the option to preview the layout on multiple screen sizes.

There are some disadvantages to using Android, however. There are many different versions of the official Android kernel, and Android users are generally slower than other operating system users to download and apply updates. The kernel is also open-source, so many different unofficial versions are also in use. To further increase fragmentation, there are hundreds of different devices that are able to run Android, all with different hardware specifications. So when an application is developed for the Android platform, it is implausible to test it on even a fraction of the environments in which it might be used.

3.4.1.3 iOS

iOS was another possible platform for the application. It is currently the most widely used mobile operating system in many countries, including the United States, Canada, Japan, and the United Kingdom, with over 1 billion devices running it worldwide. It was attractive to us because it is known for its smooth and responsive operation, and it is popular enough that it is very likely that, in any given room at UCF, there will be someone with a device running iOS. Even one of the group members owns one, so it would be easy to perform real-world testing of the application.

Apple has released an IDE called Xcode to help with application programming, as well as a software development kit that comes bundled with an “iPhone simulator” that would make the tasks of testing and debugging the application much easier. Xcode is known for its built-in “Interface Builder” application, which provides an easy and intuitive way to create graphical user interfaces.

Disadvantages of iOS include the fact that it is limited to Apple devices and it is closed source, which limits availability, but having fewer devices that can run the app also usually means fewer compatibility issues. Also, if iOS were chosen, it’d almost certainly have to write the app in Objective-C, which is a programming language that none of the group members have any experience with. There is also a steep fee that is required to load an application from the IDE onto an Apple device, as well as a fee to publish the application to the Apple App Store.

3.4.1.4 Windows

Windows is another option to host the app. The group could either develop a simple program for a Windows desktop computer or a mobile app for the Windows Phone App Store. Writing a program for a Windows desktop computer would probably be the quickest and easiest solution. Since a Windows computer will be used to develop the program, creating it for a Windows computer would allow for easy testing and maximum compatibility. It would also ensure wide availability, as Windows accounts for more than 85% of the desktop operating system market share.

A Windows Mobile phone application was also an attractive possibility. It combines all of the familiarity and compatibility of a Windows desktop system with the portability of a smartphone. Microsoft has bundled a Windows Mobile emulator with the newer versions of Visual Studio, so testing the application would be easily available right in the IDE. Unfortunately, none of the group members own a Windows phone, nor were they able to find any close friends with one, so it did not seem like a good idea to require one for the demonstration.

3.4.1.5 Comparison and Decision

There are pros and cons to each platform option. The group was first drawn to the web-based application because it would be easy to access, maintain, and update for a multitude of users. Unfortunately, most of the pros of this option do not really apply to this project because it is not desirable to have many people accessing the analyzer. For now, there is only one display and it would not make sense to grant multiple people access to it simultaneously. It would make more sense to use a program that is installed on a single device so that it can ensured only one person is able to control the analyzer at any time.

The next option that was ruled out was writing a program for a desktop computer. Although Windows computers are everywhere, the group did not want the user to have to sit down at a laptop or desktop to interact with the spectrum analyzer as they felt it would draw their attention away from the colorful light show. So it was decided that for the best user experience, the application should be available on a mobile device.

Once it was decided on developing a mobile application, the decision of which platform to host it was narrowed down to Android, iOS, and Windows Mobile. Windows Mobile seemed like the easiest choice since the group has more experience developing Windows desktop programs than anything else. Since the group would be using a Windows computer to develop the application, they thought a Windows Mobile simulator would be more realistic and offer more compatibility than an Android or iOS simulator. Also, they would be able to use the languages and IDEs with which they are already familiar, further speeding up the development process. However, Windows Phones are not nearly as common as Android or iOS devices, comprising less than 3% of the market share for mobile and tablet operating systems. The group wants the application to be widely available, so they decided to opt for one of the more popular platforms, and the options were narrowed down to just Android and iOS.

Upon researching these two options, the group came across dozens of articles supporting each, and both seem to offer many advantages. As far as availability is concerned, Android is currently the most popular operating system in the smartphone market, while iOS is dominating the tablet market. Android is open source and can be run on hundreds of different devices, which is good for availability, while less than 30 different devices have access to the closed-source iOS, which is a plus for compatibility. iOS applications have fees associated with development, but also tend to make more money on the app store, and are harder to pirate. The group, however, is not trying to make money from the app, and they are not using a bunch of fancy artwork that will need to be redrawn for each different screen configuration that will try to use it. They just want to develop a simple and clean UI that could be easily distributed to the group members. So for those reasons, they decided to develop the application for Android.

3.4.2 Languages

Once the group decided to develop the application for Android devices, they then needed to decide which language to use. To speed up the application development process, they wanted to pick a popular language so they could easily find tutorials and supporting material on the internet. They also wanted a language that would offer some portability.

3.4.2.1 Java

Java seemed like the obvious choice. The TIOBE index, which ranks the popularity of programming languages, lists Java as the most popular language worldwide, and attributes some of that popularity to its use on Android. Java is the standard language for Android application development, and is the one recommended by the official Android website. Even the Android Studio IDE and the Android SDK were written in Java. There are several advantages to using Java.

Being the standard language for Android ensures maximum compatibility with whichever device is going to run the application. Not that compatibility is a concern with Java anyway; Java apps create a JVM (Java Virtual Machine) which is an abstract environment in which to run. This increases portability since a Java application can be written once to run on any machine that can support the JVM. And with over 3 billion devices running Java worldwide, you are rarely more than a few feet away from a compatible device.

One disadvantage of Java is that it has a relatively steep learning curve, although this is somewhat mitigated by the fact that the group has some prior experience with Java programming. That, combined with the large and helpful Java community, left the team feeling confident that they could easily learn whatever they needed to develop the app.

3.4.2.2 C++

C++ was another option for the application. The Android operating system itself was written primarily in C and C++, which is why the team was originally drawn to the idea. The Android NDK (Native Development Kit) provides some tools for C++ app development, however the official Android documentation discourages it, saying most applications will not benefit from writing in the native language. The NDK itself is based on command-line tools and lacks a GUI for building and debugging applications, although third-party tools can be used to integrate the NDK into an IDE such as Eclipse or Visual Studio.

3.4.2.3 Python

Python is another high-level programming language that has been gaining popularity lately. Python is known for being easy to learn and is popular enough that plenty of tutorials and learning materials exist online. It has a simpler syntax than Java or C++ and is very readable. White space matters in Python, which sounds like an inconvenience, but it actually serves to create a standardized style. It also allows for dynamic typing, which is a convenient little feature that can simplify code.

Python is also known for its flexibility; it can used in several different programming paradigms, including procedural, imperative, and object-oriented. Another thing the group likes about Python is that it is free and open-source. And although programming in Python requires the installation of an interpreter, third-party tools can be used to convert Python programs into independent executable code that can be run without one.

3.4.2.4 Comparison and Decision

After reviewing some of the options, it was clear that Java is the most sensible language to use. The open-source nature of the Android platform has allowed it to spread to hundreds of different devices, and many different hardware architectures. Using a language like C++ would not only restrict access to available APIs, but would also introduce compatibility issues and potentially bloat the executable file to accommodate different architectures. And while some people claim to see small performance gains by programming applications entirely in Python, Java provides more than enough power to run the application. The app does not need breakneck speed or to utilize every ounce of processing power available. The team is more concerned about being able to build a simple and clean UI that works reliably, and for that, Java cannot be beaten. Java even offers a Bluetooth API for use with Android devices that will surely simplify the task of establishing wireless communication between the application and the spectrum analyzer.

On a personal note, using Java would also provide a more rewarding experience. Since Java is so versatile and widely used, employers tend to value a Java background more highly than one with a more specialized language like Objective-C.

3.4.3 App to Microcontroller Communication

Once the microcontroller is functional in the spectrum analyzer and the application is functional on a smartphone, the two need to be able to communicate with each other. There are several options available to facilitate this, such as Bluetooth, Wi-Fi, ZigBee, and NFC.

When choosing a tool to implement wireless communication, there are several important aspects to consider. The first consideration was the maximum range that would need to be covered. If the analyzer were connected to the internet, it could theoretically be controlled from anywhere on Earth. NFC, on the other hand, is a protocol that only operates in the 5 to 15 centimeter range.

Another aspect to keep in mind is the power requirement. The spectrum analyzer will be plugged into a wall, so a steady flow of power can be safely assumed for it, but the application will be communicating from a smartphone, which is powered by a battery. Therefore, the ideal communication protocol would be one that has low power usage, at least on the application side, to prevent unnecessarily draining the user’s phone battery.

Another consideration to keep in mind is cost. The communication module should be as inexpensive as possible while still providing fast and reliable communication. A brief outline of the available options is presented below.

3.4.3.1 Bluetooth

Bluetooth is a short-distance wireless communication protocol originally standardized as IEEE 802.15.1. It operates on frequencies ranging from 2.4 to 2.485 GHz and generally has a range of about 10 meters or less. It uses a master-slave structure with one master being able to serve up to seven slaves at the same time.

Bluetooth devices come in three different classes: Class 1 are the most powerful, using a maximum of 100 mW to achieve a range of about 100 meters; Class 2 modules are middle-of-the-line, using up to 2.5 mW with a range of about 10 meters; and Class 3 are the weakest, consuming only 1 mW or less, but with a range of only about 1 meter. A Class 1 module would be a bit overkill for the spectrum analyzer; the range is impressive but the power consumption is simply too high for a smartphone to sustain for very long. A Class 3 module uses very little power, but the range is very short, and actual operating range is usually quite a bit less than the theoretical maximum range anyway. Class 2 modules are a good compromise between power consumption and range, so that is the type that will be taken into consideration.

One of the cool features of Bluetooth is its use of frequency-hopping spread spectrum. Using between 40 and 79 channels, it “hops” between frequencies at a rate of about 1600 hops per second, in a pseudo-random order that is known to both the sender and the receiver. This makes the signals difficult to jam or intercept. It also allows one master node to communicate with several slave nodes without worrying too much about interference.

Bluetooth modules are also fairly cheap. At the time of writing, a Bluetooth v4.0 BLE (Bluetooth Low Energy) USB dongle is available for less than $10. The v4.0 modules have been updated to reduce latency from about 100 ms to about 6 ms, while reducing average power consumption and increasing security, but at the cost of reduced throughput from about 0.7 – 2.1 Mbit/s to about 0.27 Mbit/s.

3.4.3.2 Wi-Fi

Wi-Fi is a wireless communication protocol based on the IEEE 802.11 standard. Most commonly used for WLANs (Wireless Local Area Networks), Wi-Fi is a very popular protocol for home networks to connect computers to the internet. Most Wi-Fi standards, such as 802.11n and 802.11g, operate at the 2.4 GHz frequency band, while 802.11a uses the 5 GHz band. Wi-Fi’s throughput is fairly impressive; the old and slow 802.11b caps out at 11 Mbit/s, while the newer 802.11ac protocol has been used to in excess of 1 Gbit/s. A typical access point can have an effective range of about 20 meters, although signal boosters and amplifiers can be used to greatly increase this.

Wi-Fi has undergone a considerable evolution since the inception of the 802.11 protocol in 1997, which could only handle 2 Mbit/s data transfer. Just two years later, the 802.11b standard was released and upped the maximum data rate to 11 Mbit/s. The most popular standard today, according to PCMag.com, is 802.11n. Released in 2009, 802.11n increased the maximum transfer rate to 600 Mbit/s. It uses multiple antennas in a MIMO (Multiple Input Multiple Output) scheme to multiplex data streams. This allows multiple signals to be passing into and out of the access point simultaneously without causing interference.

Implementing Wi-Fi on the spectrum analyzer would require an access point and a Wi-Fi adapter. Luckily, smartphones capable of running the Android application used to control the analyzer will almost certainly have a built in Wi-Fi module. The access point could be mounted on the analyzer itself. At the time of writing, a cheap access point capable of 300 Mbit/s data transfer on the 2.4 GHz band costs about $25. Unfortunately, 802.11n requires a lot of power. According to a study that measured the power consumption of 802.11n modules in smartphones, even using a low transfer rate of about 7 Mbit/s consumed over 300 mW of power.

3.4.3.3 ZigBee

Based on the IEEE 802.15.4 standard, ZigBee is a relatively new technology that was first standardized in 2003. It was designed to be a simpler and less expensive competitor to Bluetooth. Its range can theoretically reach up to 75 meters, although in practice this is usually limited to about 10 – 20 meters. A ZigBee module can operate at the 2.4 GHz frequency band, with a maximum transfer rate of 250 kbit/s. ZigBee modules consume very little power; generally 1 – 2 mW while transmitting, and less than 0.2 mW when not transmitting.

The ZigBee standard uses an interesting modulation scheme called DSSS – Direct Sequence Spread Spectrum. DSSS works by continuously phase-shifting a sine wave with a pseudorandom string of pulses, called chips, at a rate much higher than the data rate. The string is known by both the sender and the receiver, so the receiver can negate the chips to retrieve the original signal. While in transmission, the signal appears as noise to receivers that don't know the string, which helps to guard against interference when multiple devices are communicating in close proximity.

There are three different types of ZigBee devices: ZigBee Coordinators, ZigBee Routers, and ZigBee End Devices. One Coordinator is used per network and contains all of the information about the configuration of the network. Routers are used primarily to relay information from other ZigBee devices to increase the range of the network in a mesh topology. End Devices are the cheapest and most basic, and are only used to transmit data to their parent device (either a coordinator or a router). Implementing ZigBee on the spectrum analyzer would require one Coordinator and one End Device. At the time of writing, these would cost about $17 each online.

3.4.3.4 NFC

NFC, short for Near Field Communication, was another option for wireless communication that was briefly investigated. NFC operates at a frequency of 13.56 MHz and can transfer data at speeds ranging from 106 kbit/s to 424 kbit/s. NFC devices can operate in three different modes: Card emulation mode and reader/writer mode allow simple unidirectional exchanges of data, while peer-to-peer mode enables two NFC devices to transmit data with each other. NFC devices aren't terribly expensive; at the time of writing, a 5-pack of NFC tags costs only about $6, while a USB NFC tag reader is available online for about $27. Unfortunately, NFC only operates in very close ranges, typically 10 cm or less.

3.4.3.5 Comparison and Decision

Table 3.4.3.5 summarizes the options investigated for facilitating communication between the mobile application and the microcontroller.

	Bluetooth v4.0	**802.11n Wi-Fi**	**ZigBee**	**NFC**
Operating Frequency	2.4 GHz	2.4 GHz or 5 GHz	2.4 GHz	13.56 MHz
Maximum Throughput	0.27 Mbps	300 Mbps	250 kbps	424 kbps
Power Consumption	2.5 mW	300 mW +	1-2 mW / 0.2 mW*	Minimal
Maximum Range	10 m	20 m	10 – 20 m	10 cm
Approximate Cost**	$10	$25	$34	$27

Table 3.4.3.5: A summary of wireless communication options
** 1–2 mW while transmitting, ~0.2 mW while not transmitting*
*** The costs for Bluetooth and NFC assume use of a smartphone with these capabilities built-in.*

The first option that was ruled out was using NFC. It was originally appealing because of its ad-hoc nature and the extremely low power consumption, but the 10 cm range of NFC is just too short for it to be seriously considered for this application.

The next option that was eliminated was ZigBee. ZigBee looked good on paper because it had low power consumption and decent range. It's also designed to be used in ad-hoc scenarios without little to no existing infrastructure. ZigBee is good for a lot of things, but it might not be the best tool for this project. For one, ZigBee has the lowest data transfer rate of the four options, with a paltry 250 kbps being the maximum over-the-air data rate, while the actual effective transfer rate is even lower. Plus, since the application is being developed for use with an Android smartphone, it can be assume that the phone will have Bluetooth and NFC capabilities built-in. This is not the case with ZigBee, meaning implementing it would require extra hardware compared with those options. For those reasons, ZigBee was discarded from consideration.

With the options narrowed to just Wi-Fi and Bluetooth, it was time to look at more technical specifications. Both options provided plenty of range, but Wi-Fi has much more throughput; 802.11n in particular is capable of transmitting about a thousand times faster than Bluetooth. This was an interest to the group since one of the stretch goals for the project was the ability to use the application to send a digital audio input signal wirelessly to the spectrum analyzer, a feature that might require some extra throughput.

Bluetooth, on the other hand, is cheaper, easier to implement, easier to use, and requires less infrastructure. Plus, upon further consideration, it was decided that even if the audio file to be inputted to the analyzer had to be compressed down to 128 kbps in order to send it over Bluetooth without lag, this would be plenty of information to warrant a decent performance from the analyzer. So for those reasons, Bluetooth was selected.

4.0 Related Standards and Constraints

This section of the report will discuss the related standards and constraints of the Dancing Water Spectrum Analyzer. The project will have to comply with standards provided from the various parts being used. Constraints applied to the project deal with the amount of time available to work on the project and budget.

4.1 Related Standards

One of the goals of the project is to have a user interface that is an Android application and many standards are required for an application. The project will be getting its power from the wall outlets and there are standards on what wall plug to use. This section will discuss in further detail the related standards that might be used in the Dancing Water Spectrum Analyzer.

4.1.1 Bluetooth

Bluetooth is a wireless communication protocol. It was originally standardized by the Institute of Electrical and Electronics Engineers as IEEE 802.15.1, although responsibility of the maintenance of the standard has since shifted to the Bluetooth Special Interest Group. The standard outlines all of the constraints a device must follow to be called a legitimate Bluetooth device. Some of these constraints are outlined below.
Bluetooth operates in the ISM (industrial, scientific, and medical) band of radio frequencies that are internationally unlicensed. Specifically, Bluetooth uses frequencies ranging from 2.4 GHz to 2.4835 GHz. It uses 79 different channels. Each channel is 1 MHz wide and there are 2 MHz guard bands on each end of the range. The power specifications vary depending on what class of radio transmitter is being used. The Bluetooth modules present in cell phones use Class 2 transmitters, which have a maximum output power of 2.5 mW, a minimum output power of 0.25 mW, and a nominal output power of 1 mW.

The modulation scheme used by transmitters is GFSK (Gaussian Frequency Shift Keying) with a bandwidth-bit period product BT = 0.5 and a modulation index between 0.28 and 0.35. Binary ones are represented by positive frequency deviation and zeros are represented by negative frequency deviations. Data transmission has a symbol rate of 1 Ms/s.

Receivers have constraints too. Receivers must have a sensitivity of -70 dBm or less, so that a raw bit error of 0.1% is maintained. The receiver should operate with a maximum usable input level greater than -20 dBm. Interference on adjacent channels is measured with the wanted signal 10dB over the reference sensitivity level, while interference from frequencies outside of the normal Bluetooth operating bands is measured with the wanted signal 3dB over the reference.

4.1.2 Android Application

One option for a user interface for the Dancing Water Spectrum Analyzer is to have an android application. In order to make the application, Android has many standards that are required for each of their programs. Android’s standards range from visual design to functionality and readability.

The Android standards for the user interface require the application to not use a system icon for another purpose. For example the back button cannot be used for any other function then for its original purpose. The application cannot replace the looks of a system icon with something completely different. If the application’s system icon is very similar it can be used but must perform the standard system behavior. Every screen or window of the application must be able to be closed by the back button. The home button must return the user to the home screen at any point in the application. If the application has notifications they must be stackable if there are multiple notifications. The notifications cannot contain advertisements or anything unrelated to the application. Notifications are only used for updating content relating to the user personally.

Functionality standards for Android require only the minimum permissions from the user in order for the application to function. The application cannot request permission to the user’s contacts and other sensitive data. It cannot access functions like the phone or messaging that would cause the user money unless that is what the application was designed to do. It is recommended that Android applications are able to support being installed on the phone’s SD card. Sound from the application should not play if the screen is off unless it is the purpose of the application. The application needs to support both portrait and landscape orientations of the device and provide the same features in both set ups. For both orientations the application must take up the entire screen of the device. When the application is running in the background it should not have any services running while not being used. If the user navigates away from the application, the application does not lose any data. When the application is resumed it returns to its previous window. If the device was put to sleep when it wakes up the applications resumes to its previous state. If the user pressed the back key the application prompts the user if they would like to save their place on the app otherwise the information will be lost.

Stability and performance for Android applications also have standards. The goal is to have that application not crash or function abnormally on any device. The application should load quickly in under two seconds otherwise the application should notify the user that the loading is in progress. Power management should be a feature for the application in the settings. If audio is played from the application it should be smooth with no distortion. All images from the application should be clear and not distorted or blurred. The text must be easily readable and not cutoff or have any improper word wraps.

Having the application on Google Play also requires it to have specific standards and policies. The application should not have any inappropriate content and adhere to all the guidelines in the Google Play Developer Content Policy. The maturity level of the application needs to be selected based on the Content Rating Guidelines. If the application requires to request access to the device's location it cannot be rated "everyone". The graphic for the applications needs to be a high-quality picture and should not contain any small or illegible words. The graphic should not be a screenshot of the application or resemble an advertisement. Any pictures or videos in the description of the application on Google Play cannot contain a non-Android device. The pictures or videos cannot mislead customers on what the application does in anyway. If there are common bugs that are reported by the user the application development group should respond and attempt to address the bug reports.

4.1.3 Power

In the Unites States of America the standard for transporting electricity is a voltage of 115V and a frequency of 60Hz. Power is sent by using an alternating current source from a power distribution center to homes and businesses. For a customer to use the provided electricity in the United States a wall outlet is the connection. The NEMA 5 is the standard wall outlet in the USA for homes and businesses. These have a have three wires with a ground in the middle rated for a maximum voltage of 125V. The wider T shaped slot is the neutral connection. The Dancing Water Spectrum Analyzer will use a NEMA-5 connecter cable to receive its power from the wall outlet.

4.1.4 LEDs

The Dancing Water Spectrum Analyzer will be using LED's as the primary source of lighting for the water display. LED's are characterized by their luminaire efficacy. The standard LM-79 is used in calculating this statistic by using the net-light output and dividing it by the input power in terms of lm/W. The standard LM-80 is designed to provide information on how long an LED will provide the correct lumen before it declines below 70% of its initial lumens. Energy Star is a standard that requires LED's to have a life time of 25,000 hours under normal usage and 35,000 hours in industrial usage.

4.2 Realistic Design Constraints

In order to design and construct the Dancing Water Spectrum Analyzer within the two semesters of senior design 1 and 2, certain limitations had to be applied to the project. The project also had a budget provided by a sponsorship from Boeing and Leidos. The following sections will discuss the limitations of the project due to budget and time constraints.

4.2.1 Portability

The goal of this project is to create a water display that is portable. For the display to be portable it must be small enough to be carried by one person. If one person is carrying the Dancing Water Spectrum Analyzer it needs to remain as lightweight as possible.

Small fountain water pumps will be used in the water display. The fountain pumps selected for the display are lightweight and weigh about 50grams. Sixteen of them are going to be used in the fountain and their combined weight is around 1.8lbs. Most of the weight from the project will be due to the display structure around the water pumps. The display structure will be made of wood, plywood, and acrylic. The measured weight of the display structure is not calculated yet because the display is still subject to change. The group would like to keep the weight of the project under ten pounds.

One option to keep the water display lightweight is to allow the water reservoir to be removable. This will allow for easy filling and emptying of the water. It is currently planned to have the water pumps be attached to the inside of the water reservoir but not have the reservoir attached to anything itself. The reservoir for the project will be a plastic bin able to hold about 13 gallons of water. Carrying 13 gallons of water is not recommended and would make the Dancing Water Spectrum not portable if the plastic bin was not removable from the display structure. It would be ideal to leave the plastic bin inside the display structure while adding the water to not have to move the display structure after the 13 gallons of water has been added. In order to remove the water a hose or small bucket can be used to remove the majority of the water. Once most of the water has been removed from the plastic bin, the plastic bin itself can be lifted out of the display and completely emptied. The other components like the power supply box and the controlling box will be small enough to be able to fit inside the water reservoir once emptied of water for easy transportation.

The clear acrylic display will be attached to the lid of the water reservoir. The lid of the plastic bin is able to lock on to the bin itself allowing the acrylic display to be secured in place. The total size of the display frame is slightly over three feet in length and one and one half feet in width and half a foot high. The acrylic display will be about three feet in length and one and one half feet in width and one foot high. Having these dimensions for the Dancing Water Spectrum Analyzer will allow the project to be carried one piece at a time by a single person.

4.2.2 Time

The time allowed for this project is two college semesters. The first semester is used for researching the topic and the seconded semester will be making any necessary changes and the projects construction. In order for the Dancing Water Spectrum Analyzer to be completed on time, the group decided to keep the project small in scope. The initial project was to have a large fountain display that used thirty-two water pumps. This idea was altered into only using sixteen water pumps.

4.2.3 Cost and Budget

The Dancing Water Spectrum Analyzer received funding from Boeing and Leidos of $899.46. The sponsorship will be able to cover the expenditures of the project. The majority of the cost of the project will be from the display structure and the water pumps. The fountain display was reduced from its initial size due to budget restrictions. The next largest purchase will be the power supply and the developmental board for the PIC32 microcontroller.

5.0 Project Hardware Design Details

The following sections describe the hardware of the project in a comprehensive level. All of the internal circuit components, such as drivers and microcontrollers are included in this section.

5.1 Analog Input

In order to for the Dancing Water Spectrum Analyzer to analyze music, the device must have an ability to take in a music sample at its input in order to visually display. Audio signals can be represented in digital or analog format, and they can be processed in both ways. For the project, the input signal will be an analog signal in the time domain. Thus, the signal will be represented as a set of continuous values, rather than discrete. This analog signal coming from an external source could either come from a microphone or a phone jack. For the project, the design team decided to use an audio jack. The design team chose this because getting an audio from a microphone may bring in other external audio sources other than the music itself, and in order to prevent this, the design team determined that using a direct audio jack would be the best option for us. In addition, in order to play the music from a speaker and analyze the audio signal simultaneously, the design team needed to have a way to split the audio signal two ways. The other option was to play the same audio from two different devices at the same time, with one device connected to speakers and the other to the analyzer. Due to potential synchronous error, the design team decided to use only one input source. In order to do this, the design team decided to use a split headphone jack. This jack will divide the audio signal, so that there will be two identical outputs but with different amplitudes. One output will directly go to the input of a speaker, which will play the music for all to hear. The other output will directly go to the input of the spectrum analyzer processor, which will then be analyzed by the processor and consequently displayed by the water pumps. A block diagram of the input of the analyzer is displayed below on Figure 5.1.

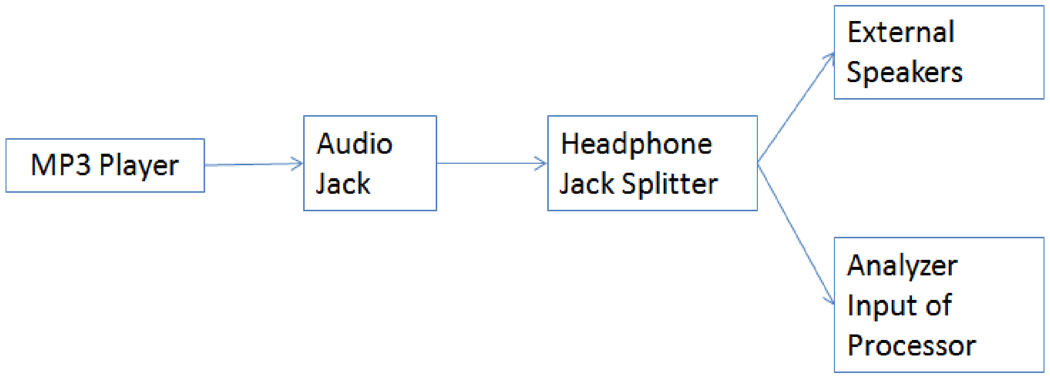


Figure 5.1: Block Diagram of Split Music Audio Input Signal

5.1.1 Audio Signal Characteristics

Audio signals are characterized by several qualities such as voltage, bandwidth, and power. If the upper requirements for frequency and voltage are too low, the signal will be lost and the spectrum analyzer will not be accurate. If the design team sets the limits too high, they will maintain the signal, but it will be inefficient. Since the input signal is in the human audio spectrum, the design team will limit the voltages and frequencies of the input signal in order to fit these requirements.

5.1.2 Input Voltage

For the input to the spectrum analyzer, the design team will be using a line input electrical signal, rather than a microphone input. Since the audio is coming in through a jack, the design team had to use line levels, which is the strength of an audio signal used to transmit audio signals through different components. For electrical input signals, voltages have values ranging from +2 to -2 volts, or 2 Volts Peak to Peak. Since the voltages may be negative, the design team will have to level shift the signal voltage so that they are all positive values for the analyzer.

5.1.3 Input Frequencies

The frequencies that the design team will utilize and display on the spectrum will be limited to the frequencies that are audible to the average human. Any higher frequencies will not be audible and are pointless due to the fact that the spectrum analyzer only responds to music audio, and thus they will be filtered out. The standard range of frequencies for humans is 20 to 20,000 Hertz. 16-32 Hertz are the human threshold of hearing, synonymous to the lowest notes of a pipe organ. At around 16,000 Hertz, it sounds like sibilance in speech. At around 4,000 Hertz, one can hear the highest note on a standard piano. Since most music will not go above this frequency level, the design team decided to limit the highest frequency to this, as the design team felt it was unnecessary to include frequencies that were not common in music. Thus, the design team decided to only analyze the frequencies from 0-4,000 Hertz. The rest of the frequencies will be filtered out the by utilizing a filter.

5.1.4 Level Shift and amplification

Since the audio input signal must be converted from an analog signal to a digital sampled signal within the processer using an Analog to Digital Converter, the design team must shift the audio voltage so that they are all positive values. In addition, A/D converters only operate at certain voltages, and the design team must take care to fit the signal within this given voltage. The A/D converter that the design team will be utilizing for the project operates at 0-3.3V as given by its datasheet. Since the audio signal the design team will use will swing from -2 to +2 Volts, the design team must shift the voltage level by 2V to ensure that there are no negative voltages and consequently the design team must attenuate the voltage of the input to the A/D converter so that the maximum amplitude will be limited to 3.3 Volts. Thus, the design team will utilize a specific configuration of two Operational Amplifiers to do the level shifting and gain.

5.1.4.1 Amplifier Notes

In order to perform a level shift, the design team had to properly utilize and configure an operational amplifier. For the purposes, the design team decided to use the Texas Instruments TLV2772CP model Operational Amplifier. This device boasts a high slew rate of 10.5 V/µs, a high bandwidth of 5.1 Mhz, rail to rail output swing, and a 1mA supply current per channel. In addition, it features a 360 µV Input offset voltage, and a low distortion of 600 Ω. More importantly, this op amp operates at 2.5-5.5V which fits the 3.3V criteria. This operational amplifier comes in a DIP package, which is required, as the design team will be soldering the own Integrated Circuit. In the calculations for analysis, team used ideal op amp rules, since can ignore the error due to the values design team are operating with. Thus, the design team was able to use the equation E.5.1.4.1 which is the equation for the gain of a non-inverting op amp.

$$Gain = 1 + \left(\frac{R_2}{R_1}\right) \qquad (E.5.1.4.1)$$

Where R_2 is the feedback resistor and R_1 goes from the inverting terminal to ground. For the audio signal, the design team will use both a non-inverting op amp with a <1 gain, along with a buffer op amp for the reference. Thus, for the input, the design team will use 2 TLV2772CP devices.

5.1.4.2 Level Shift Schematic

In order to minimize the size of the level shifter so that it may reduce space on the circuit board, along with reducing cost, design team desired to implement single supply circuitry so that design team only needed one power source for the op amp. Audio signals must be shifted because they are usually referenced to ground, while single supply circuits need to be modified for signals above ground. Thus, design team needed to shift the audio signal above ground. Traditional operational amplifier level shifters require several op amps, and more importantly, a negative rail. Thus design team had to implement an alternative configuration to maintain single supply. The following schematic on figure 5.1.4.2-1 level shifts a ground referenced signal with only one op amp, while only using a single supply voltage. There is also unity gain op amp to provide a buffer to the level shifting op amp. This buffer op amp provides the reference voltage. The non-inverting summing op amp will sum the input signal with a reference voltage. Consequently, this configuration will convert a negative to positive signal into only a positive signal. In addition, the resistors on the op amp will ensure a gain that will attenuate the new 0-4V signal to a 0-3.3V signal.

Our single supply voltage is set to 3.3V for the signal voltage. It is 3.3V because that is the operating voltage specification of the microprocessor and design team found it efficient to only use one power supply for all of the signal components. Then, the voltage divider equation (E5.1.4.2-1) is used to set the input voltage of the reference (buffer op amp) where:

- V_{cc} = 3.3V
- V_{ref}= reference voltage
- $R_{55} = R_{54}$ = 10kΩ
- Voltage divider output = Voltage input of level shift op amp

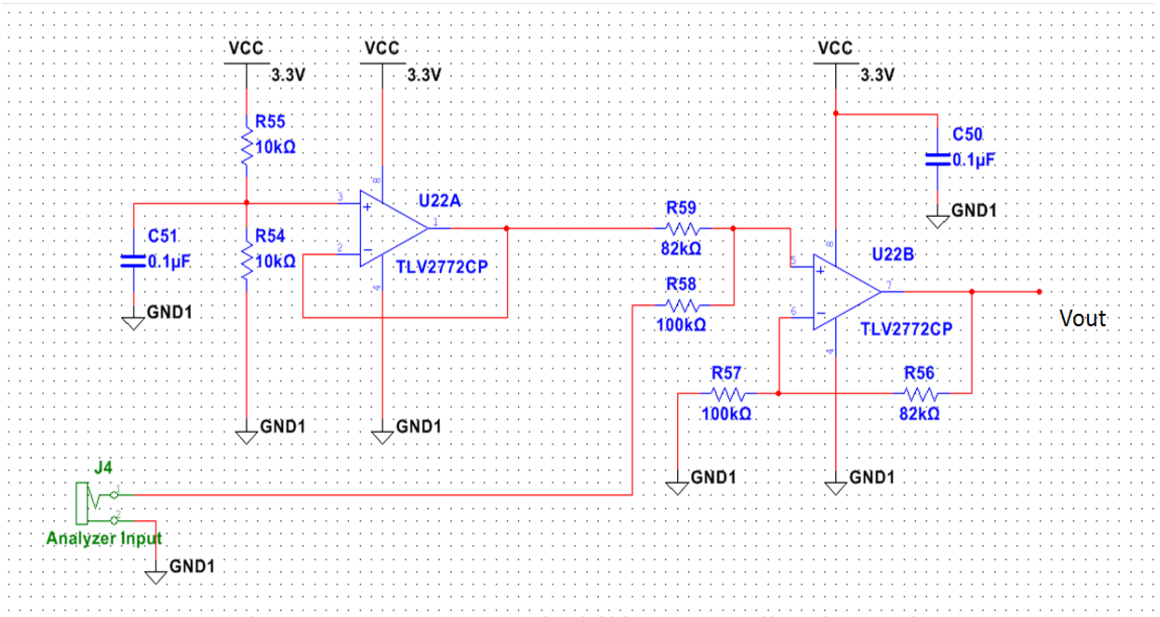


Figure 5.1.4.2-1: Level Shifting Overall Schematic

$$V_{ref} = V_{CC} * \frac{\text{R54}}{\text{R54+R55}} = 3.3 * \frac{10k}{10k+10k} = 1.65V \qquad \text{(E5.1.4.2-1)}$$

Consequently, the following equations (E5.1.4.2-2), (E5.1.4.2-3), and (E5.1.4.2-4) were used to get the final output of the level shifter:

$$Gain = \left(\frac{R_{59}}{R_{57}}\right) * \left(\frac{R_{57} + R_{56}}{R_{58} + R_{59}}\right) \qquad \text{(E5.1.4.2-2)}$$

If $R_{57} = R_{58}$ and $R_{59} = R_{56}$,

$$Gain = \frac{R_{59}}{R_{57}} \qquad \text{(E5.1.4.2-3)}$$

If $R_{59} = 82.5\text{k}\Omega$ and $R_{57} = 100\text{k}\Omega$,

$$Gain = \frac{82.5k\Omega}{100k\Omega} = 0.825$$

$$V_{out,max} = V_{in,max} * Gain = 4V * 0.825 = 3.3V \qquad \text{(E5.1.4.2-4)}$$

This is the desired scaled down maximum voltage, and thus this is a valid signal to send in to the antialiasing filter. Thus, design team used the resistor values of 82kΩ and 100kΩ for the board. The design team picked an 82kΩ resistor because there are no manufacturers that produce 82.5kΩ resistors and the error is negligible. The capacitors design team used were decoupling capacitors to stabilize and reduce noise. The design team picked a 0.1µF capacitor since that is the standard value for decoupling capacitors given in the datasheet. The following Figures 5.1.4.2-2 and 5.1.4.2-3 show a simulation of the input signal. The design team used this to verify the calculations.

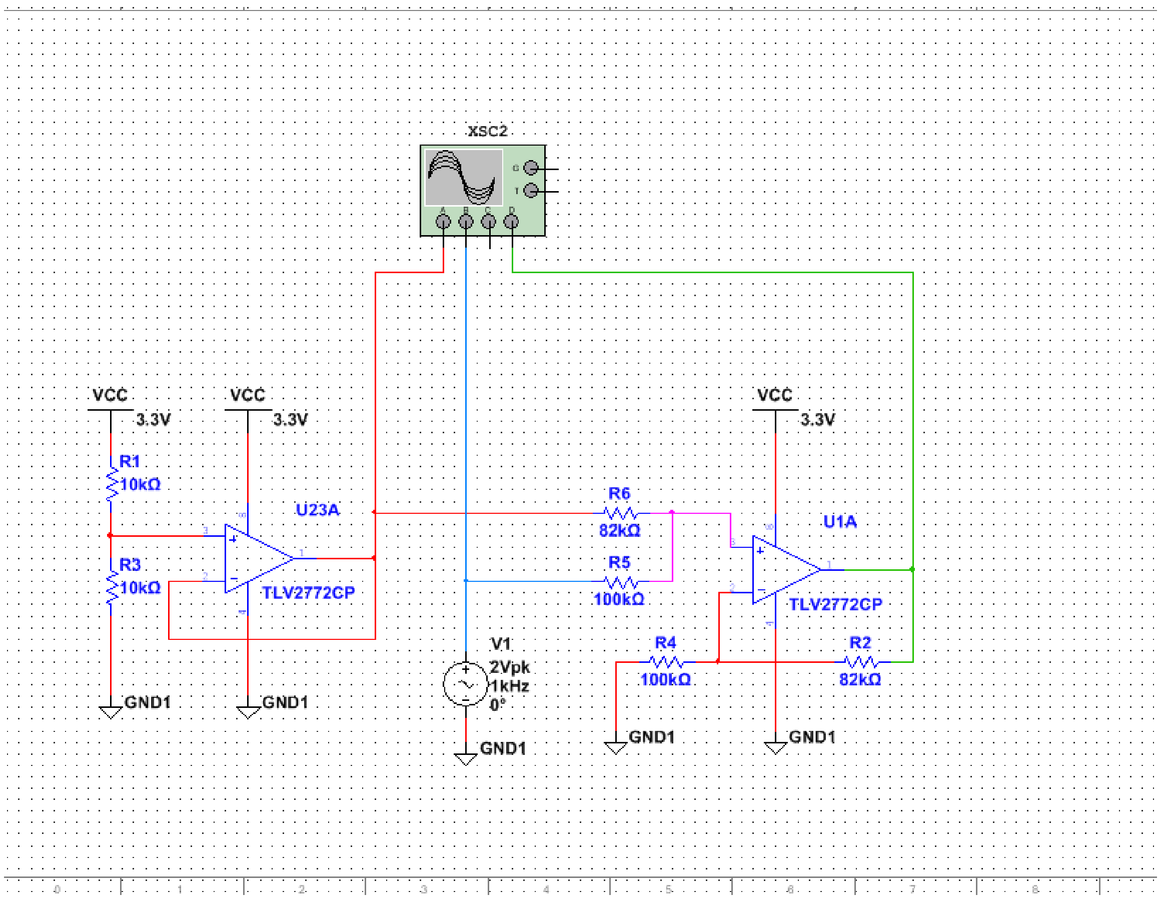
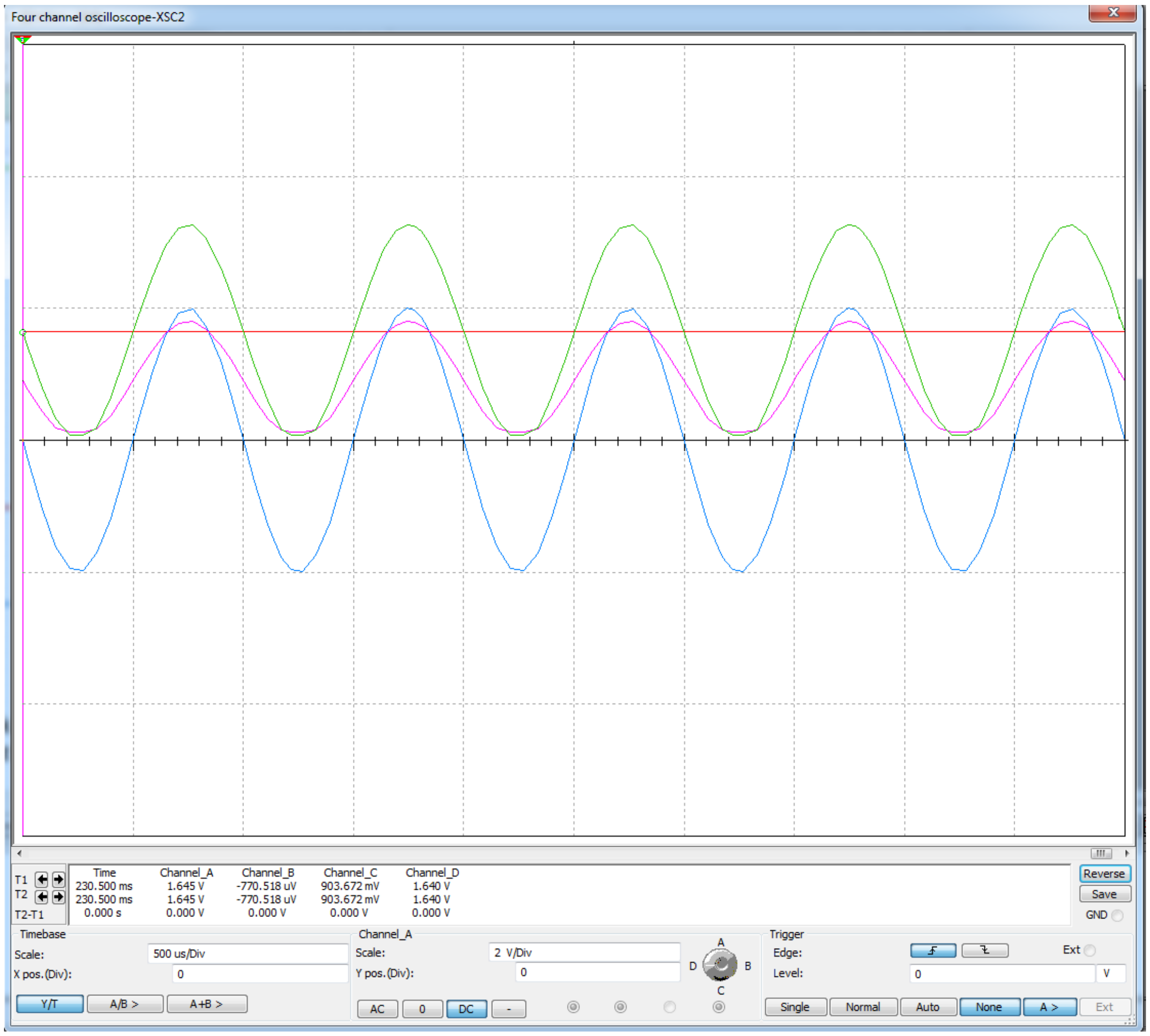


Figure 5.1.4.2-2: Level Shift Configuration with an Oscilloscope for Checking

In figure 5.1.4.2-3, the signal is being properly attenuated and shifted. The red signal represents the 1.65V reference voltage. The blue signal represents the audio signal, which has a +2 to -2 voltage swing. The green signal is the final 0-4V output of the level shift. Thus, the configuration does its intended purpose.

5.1.5 Buffer Op amp/Reference Voltage

The Follower, Unity Gain, or Buffer Op amp is a configuration for an ideal op amp that has a gain of 1. Thus, the output voltage is the same as the input voltage. While it seems useless, it is a practical circuit in that it offers almost infinite input impedance, and very low output impedance. This is useful if one wants to chain together several circuits without any issues with impedance. Since the reference voltage is created by utilizing a voltage divider from the power supply, the design team needs to prevent loading from the divider to the non-inverting level shifting amplifier, so that the calculations can be correct. This is implemented using the buffer op amp just described. With this configuration, the output reference



	Time	Channel_A	Channel_B	Channel_C	Channel_D
T1	230.500 ms	1.645 V	-770.518 uV	903.672 mV	1.640 V
T2	230.500 ms	1.645 V	-770.518 uV	903.672 mV	1.640 V
T2-T1	0.000 s	0.000 V	0.000 V	0.000 V	0.000 V

Timebase
Scale: 500 us/Div
X pos.(Div): 0

Channel_A
Scale: 2 V/Div
Y pos.(Div): 0

Trigger
Edge:
Level: 0 V

Figure 5.1.5.2-1: Oscilloscope in Time Domain

voltage will be preserved with a gain of 1, loading will be prevented and the low output impedance of a buffer will drive the reference voltage to the next circuit with minimal loss. The decoupling capacitor stabilizes the circuit and the standard value of 0.1 microfarads was used. The schematic design on figure 5.1.5 shows how the design team implemented this functionality.

5.1.6 Low Pass Filter/Antialiasing filter

A low pass filter is a filter that passes signals with a frequency lower than the cutoff frequency and will attenuate frequencies higher than the cutoff. Additionally, the attenuation will depend on the configuration of the low pass filter used. An anti-aliasing filter is a high order low pass filter that is used before sampling a signal, usually for Analog to Digital conversion. They need to be high order, because the filter must satisfy the Nyquist sampling theorem. Thus, it must be as close to an ideal brick wall filter as possible.

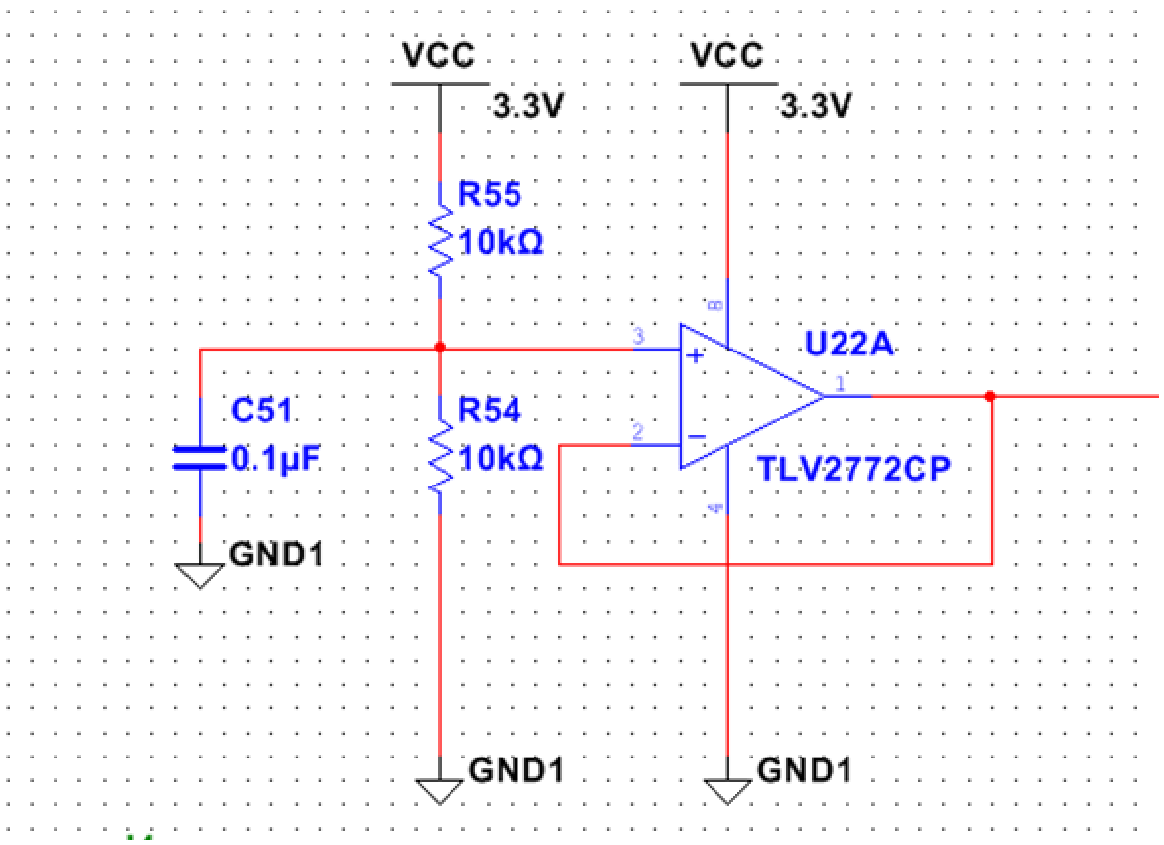


Figure 5.1.6: Buffer Op Amp for Reference

As stated earlier, humans can only hear frequencies from 0-20,000 kHz, but most audio energy in music do not exceed 4 kHz. Thus, for the signal, the design team will only need to analyze frequencies from 0 to 4,000 Hz, and the design team will filter out the rest to keep the design efficient. This will allow a lower sampling rate for the processor, along with less data to process. To do this, the design team will use a low pass filter with a designed cutoff frequency of 4 kHz. Furthermore, since the design team is going to eventually sample the signal, the design team will use an anti-aliasing filter.

5.1.6.1 Transition Band/Roll off

In real life applications, there is no such thing as a perfect filter. For every filter, there is a set of frequencies outside the pass band that is supposed to be attenuated, but is not. This is called the transition band, and it is the desire to minimize this frequency band. The roll off rate is the rate of change of power (in dB) at 10 (decade) or 2(octave) times. The smaller the transition band, the higher the roll off rate, and this is what the design team strives for. This can be achieved by increasing the order of a filter. A first order filter attenuates a signal by 20dB/decade, and a second order filter attenuates a signal of double the dB level, or 40dB/decade. Thus, the only way the design team can minimize the transition band is to increase the order of the filter as much as possible, but with the limitation of minimizing space on the integrated circuit.

5.1.6.2 Comparison and Decision of Filter

In all, there are several types of configurations of operational amplifiers to create filters. Theses configurations are famous for their ability to approximate ideal filters, and are designed to have the highest roll off rate and shortest transition band possible. The three filters that the design team compared together were the Butterworth, the Chebyshev, and the Elliptic filter. the design team analyzed the characteristics of all three of these filters, and determined the filter the design team would use based on the criteria of the shortest transition band, since that is all that the design team desired in the project, and other unique characteristics between the types of filters were not as high on the priority.

First of all, the design team considered the prospects of a high order Butterworth filter. A Butterworth filter is a common filter taught in analog circuit classes, and thus is the type of filter that the design team the design team were most familiar with. It’s simple design and calculations made this a filter that was easy to analyze and implement. Due to its unique design, a Butterworth filter is considered a maximally flat filter. This means that the filter is designed to have the most flat frequency response as possible in the pass band, and rolls off to zero in the stop band. This is a useful feature in general, but for the project, the design team did not need to minimize ripple. In addition, a Butterworth filter has one of the lowest roll off rates of all filters and this was detrimental to the design, since the design team desired a high roll off rate. Due to this constraint, the design team decided to not use a Butterworth filter in the design.

Second, the design team decided to look at a Chebyshev filter. This was another filter the team familiar with, and decided to look into this type of filter due to the combined familiarity with it. A Chebyshev filter has a steeper roll off rate than a Butterworth filter, but came with the characteristic with having a ripple effect in the pass and stopband. This steeper roll off was desirable, so the design team kept this filter in consideration. But in the search for the design with the steepest roll off, the design team decided to look into another desirable filter.

Finally, upon researching more deeply into filter design, the design team came across the Elliptic filter. This filter, also called a Cauer filter or Zolotarev filter, boasted an even steeper roll off than the Chebyshev filter, at the cost of an equalized ripple in the pass and stop band. In fact, this filter had the steepest roll off rate of any filter design the design team found. Since the design team prioritized high roll off rate over ripple effects, the design team decided that the Elliptic design was perfect for the requirements, and the design team decided to implement this type of configuration for the antialiasing filter.

5.1.6.3 Filter Implementation

Upon deciding to use an elliptic filter of a high order, the design team decided to use a premade chip due to the complicated configuration of high order filters. the design team decided upon the Maxim MAX7404 filter chip. The 8th order, low pass, elliptic, switched capacitor filter came in a DIP package, meaning the design team could solder it into the board, and since this is what the design team desired, the design team picked this model for the project. The MAX7404 additionally operates at +3 volts, which is approximately the same as the operating voltage of the microchip at 3.3V, so the design team could use an identical power supply. Since the design team wanted to minimize space on the IC, the design team wanted to minimize the amount of power supplies so this choice was optimal for the design. This filter also allows corner frequencies from 10 to 10,000 Hertz, making this an ideal choice due to the desired cutoff frequency of 4 kHz. The following figure 5.1.6.3-1 shows the Operating Circuit of the MAX7404 chip.

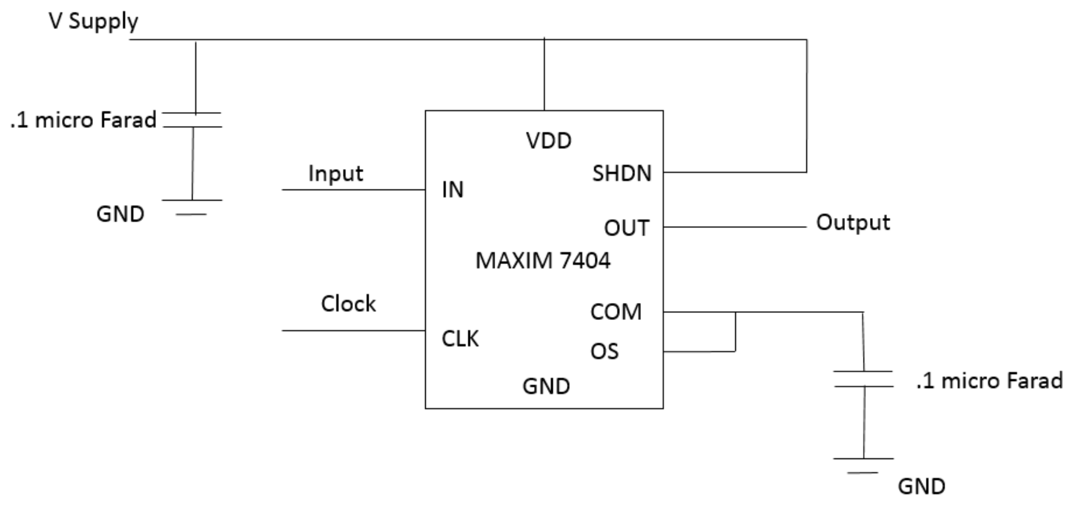


Figure 5.1.6.3-1: Circuit Model

Pin 1 is COM, which is the common input. Since it is bypassed, the design team will put the pin to ground with a 0.1μF capacitor. The design team added the capacitor because the datasheet for the MAX7404 directed us to add a 0.1μF capacitor in series with the ground. Pin 2 is IN, which stands for the pin for the input signal to be filtered. The design team attached the audio signal to be analyzed to this pin. Thus, the output of the level shift was sent to the input of the filter. Pin 3 represents ground. Pin 4 is VDD, which represents the positive supply input, which for the specific model operates at 3V. Thus the design team attached the 3.3V power supply to pin 4, along with a 0.1μF decoupling capacitor for noise reduction. Although the specifications are for 3V, the filter can operate on voltages up to 3.6V, so the slight error in supply voltages can be acceptable. The design team accepted this error because it was desired to minimize the amount of voltage regulators on the IC, as they take up a lot of space, so the design team wanted to maximize the use of the 3.3V power supply, rather than add an additional regulator that regulates to 3V. Pin 5 stands for OUT, which is the filter output, which will consequently go to the PIC32 input described in a later section. Pin 6 is OS, which stands for the offset adjust input. The datasheet directed us to connect OS to COM if no offset adjustment is needed and since the design team did not need an offset to the input, the design team followed the instructions of the datasheet. Thus, the design team connected the OS pin to the COM pin. Pin 7 is the SHDN, which stands for the shutdown input. The datasheet stated to drive the pin low to enable shutdown mode or connect to VDD for normal operation. The design team desired normal operation for the filter so the design team attached the SHDN pin to VDD as directed. Finally, pin 8 is CLK, which stands for Clock Input. Since the design team wanted to use the internal clock of the filter, rather than an external clock, the design team attached an eternal capacitor from CLK to GND to set the internal oscillator frequency. These directions were given from the datasheet along with the equations to calculate the capacitor. The following equations (E5.1.6.3-1) through (E5.1.5.3-2) show the calculations used to determine the capacitor value C_{osc}.

$$F_C = \frac{F_{CLK}}{100} \qquad \text{(E5.1.6.3-1)}$$

Where and F_c is the cutoff frequency and F_{CLK} is the oscillator frequency.

$$F_{CLK}(kHz) = K * \frac{1000}{C_{osc}} \qquad \text{(E5.1.6.3-2)}$$

Where $K = 34$ and C_{osc} is in pF. Since $Fc = 4\,kHz$, then

$$FCLK = F_C * 100 = 4(kHz) * 100 = 400\,kHz \qquad \text{(E5.1.6.3-3)}$$

and

$$400 = 34 * \frac{1000}{C_{osc}} \qquad \text{(E5.1.6.3-4)}$$

therefore

$$C_{OSC} = 34 * \frac{1000}{400} = 85pF \qquad \text{(E5.1.6.3-4)}$$

Thus, for the capacitor on pin 8, the design team chose an 85pF capacitor. Having figured out all the values for the components of the pins, the design team implemented the circuit. Figure 5.1.6.3-2 shows the implementation of the antialiasing filter which will be used in the IC.

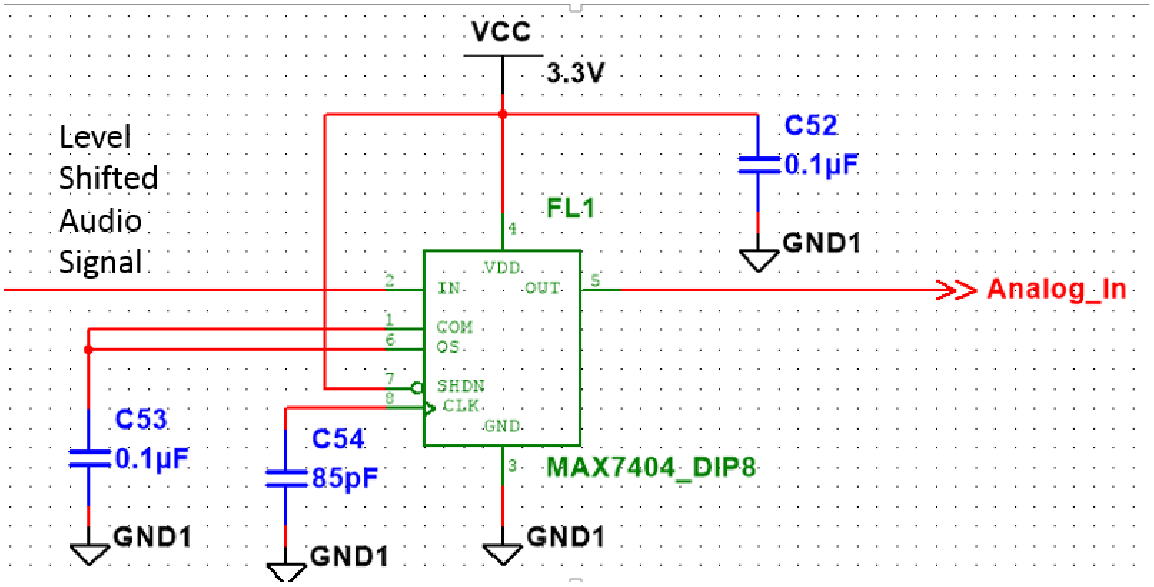


Figure 5.1.6.3-2: Antialiasing Filter Design Implementation

5.1.6.4 Nyquist Sampling

Sampling is the process by which a continuous time signal, like voltages or water levels, are converted into a discrete time signal. This is done by translating the signal into a voltage and then sending it through an Analog to Digital Converter to make the signal discrete. In signal processing, there is a concept called the Nyquist-Shannon Sampling Theorem, which is a fundamental and necessary theory that determines the sufficient condition for the sample rate of a signal that is needed to obtain all the information of a continuous signal, such as the analog audio signal. This sampling rate is needed to turn this analog signal into a discrete time signal that can be analyzed by the processor, without losing any information. The Nyquist-Shannon sampling theorem basically says that if you have a signal that is limited to frequency B, then you can obtain all the information about the signal in discrete times by setting the sampling rate by at least 2B. This is a very short but incredibly useful theorem in signal processing that the design team will directly apply in or design. As stated previously, the design team band limited the audio signal to 4 kHz due to design constraints. Thus by directly applying the Nyquist-Shannon sampling theorem, the design team determined that the design team needed to sample by at least 2 times the highest frequency. Thus, the design team needed to sample by at least 8 kHz to preserve all the information of the signal. Unfortunately, the Nyquist-Shannon Sampling Theorem is modeled for a signal that is perfectly band limited, but there is no such thing as a perfectly band limited signal. Fortunately for the design team though, perfection was not needed, and it is only required to have a design that is good enough to approximate the signal. We, the design team, decided to set the sampling rate at 10 kHz. Thus, the design team added 2 extra kHz to ensure that the entire signal is collected, while minimizing data constraints.

5.2 Bluetooth

Despite all of the restrictions that the standard places on Bluetooth devices, there are still plenty of options to choose from when picking a Bluetooth module for the spectrum analyzer. Since the application will use the built-in Bluetooth capabilities of whichever device it is running on, only one module had to actually be purchased and installed. Several modules were investigated to determine the best fit for the project. Table 5.2 provides a comparison of some of the options.

	HC-05	**TI CC2564**	**UCBT232B**	**BRBLU03-010A0**
Output Power	~2.5 mW	~8 mW	~200 mW	~4 mW
Sensitivity	-80 dBm	-95 dBm	-86 dBm	-84 dBm
Interface	UART	UART	RS-232	USB
Firmware	V2.0	V4.0 BLE	V2.1	V2.0
Transfer Rate	~700 kbps	~115.2 kbps	~115.2 kbps	~2100 kbps
Cost	$4	$14	$42	$46

Table 5.2: A comparison of Bluetooth module options

The development team decided to use the UCBT232B Bluetooth adapter on the microcontroller PCB. At first glance, this adapter might not seem to be anything special; it’s not the most sensitive receiver under consideration, nor the cheapest. It doesn’t use the most recent firmware and it doesn’t use the least power. The transfer rate is also fairly low. But all of those factors don’t matter very much for this project. The adapter will be mounted on a PCB with a power line on it, so output power is not much of a concern. The sensitivity is more than enough to reliably receive signals from a distance of 15 m. Also, the application will only be sending signals in the form of single characters, so the transfer rate doesn’t need to be very impressive either. The UCBT232B was chosen because it has one advantage over the other modules: it uses a DE-9 connector to interface with an RS-232 port. The low-power PCB on the spectrum analyzer was designed with an RS-232 port on board, so this adapter is able to be plugged right in without the need for additional equipment.

5.3 Microcontroller

A microprocessor is a small computer an integrated circuit. It contains a processor core, memory, and programmable input and output. It also contains a small amount of RAM. It is a programmable device that accepts data as input and processes it according to what the user dictates, and sends out an output. It requires little power to use and is cheap and small in size. These two qualities allow for programming in high level languages such as C or Java. Since the project requires a large amount of software programming, a microcontroller is an ideal choice for the project. This microcontroller will be the center of management of the spectrum analyzer. It is a hub for all of the connections of the project and is the most crucial part of the design. The processer will take in an analog filtered audio signal in the time domain as an input and will send out a digital quantized signal in the frequency domain. The controller will need to be able to interface with a computer, and will need the processing capacity to commit a Discrete Fourier Transform. In addition, it will have to do an Analog to Digital Conversion, along with having the quality of being in a DIP package so the design team can solder it onto the custom board. To do this, the design team desired to pick the best microcontroller for the project. The two ideal microprocessors the design team was to decide upon were determined to be between the MSP430 and the PIC32 microprocessor. The next sections document the analysis of the two, and how the design team decided which one to implement in the design.

5.3.1 Through-hole vs Surface Mount

DIP stands for Dual in-line package, which is an electronic component package that contains two rows of pins. They can be inserted into a socket, such as a breadboard, or they can be through hole mounted in a custom PCB (printed circuit board). This allows for easy replacement of a component and is safer from overheating during soldering. DIP packaging is used for through-hole technology. Nowadays, most industry PCBs no longer use through-hole technology, thus eliminating the use of DIP packages. Rather, they utilize surface mount technology, which is generally smaller, allowing much more pins to be used in an IC. In addition, SMT can acquire much higher circuit speed due to reduced size, allowing for optimal performance. Thus, Surface mount technology is much more efficient and compact than through-hole technology, making this type of implementation superior to DIP packaging. Unfortunately, it is much more expensive to implement, unlike DIP package supported technology. Since the project doesn’t require optimally compact space usage, and due to budget constraints, the design team decided that using DIP packages was the best choice for the project. Thus, when the design team looked for components for the PCB, the design team chose parts that were of the DIP variety.

5.3.2 MSP430 Microcontroller

The MSP430 Microcontroller is a family of low power microcontrollers developed by Texas Instruments. It uses a 16-bit CPU and is usually designed for low cost and low power consumption embedded systems. The MSP430 uses RISC as its architecture.

5.3.3 PIC32 Microcontroller

The PIC32 Processor is a High Performance 32-bit Microcontroller manufactured by Microchip Technology Incorporated. The PIC32 processor contains SRAM, Flash, UART, and a built in A/D converter. The project will need SRAM for memory storage, Flash to store the software code, an Analog to Digital Converter to digitize the audio signal, and UART to connect to a PC. Even more, this microcontroller is available in a DIP package, facilitating the soldering process. PIC devices are generally widespread due to their low cost and wide availability. Because of this, there is a very large user base for this processor, consequently resulting in a large amount of application notes and troubleshooting guides. This ease of use and readily available resources are a positive quality to have when implementing the processor for testing. In addition, The PIC32 Microcontroller operates using Harvard architecture. Figure 5.3.3 shows a block diagram of the PIC32 chip.

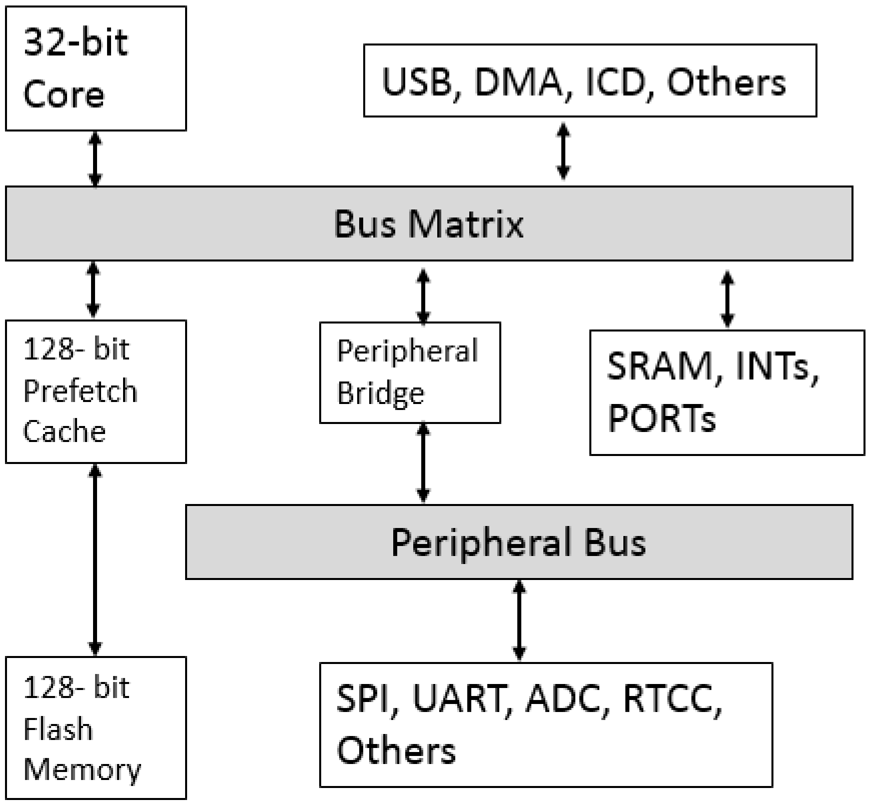


Figure 5.3.3: PIC32 Block Diagram

As shown above, the PIC32 utilizes a M4K 32-bit core from MIPS Technologies. Because it is a Harvard architecture based core, it contains separate Instruction and Data busses connected to the bus matrix, as shown in the figure. The core connects to the rest of the modules using the bus matrix. This bus matrix is a high speed switch that establishes connection between modules. As seen, the CPU core, USB, and DMA, and ICD modules connect to the SRAM, SPI, UART using the bus matrix and peripheral bus. In addition, the PIC32 can contain up to 128-bit wide flash memory, which allows a higher throughput and increased CPU performance. Also, as shown in the block diagram, the PIC32 uses a 128-bit Prefetch Cache module. This allows the PIC32 to maintain high performance, in the event that the CPU is running faster than the Flash memory speed.

5.3.4 Comparison and Decision

Table 5.3.4 shows the two microcontrollers containing the qualities that the design team analyzed to determine which processor to use. This table only analyzed the DIP package models.

Characteristics	**MSP430**	**PIC32**
Supply Voltage Range	1.8V-3.6	2.3-3.6V
Pin Count (DIP package)	20	28
Flash (KB)	16K	128KB
Analog to Digital Converter Channels	8	16
ADC bit type	10	10
Package Type	DIP	DIP
Architecture bit	16-Bit	32-Bit
Architecture type	RISC	Harvard
SRAM	512 B	32 KB
Maximum Frequency	16MHz	80MHz
SPI Modules	4	4
UART Modules	1	6
System Clock	20MHz	50 MHz

Table 5.3.4: Relevant Comparison of PIC32 and MSP430

Both microcontrollers include the properties needed for the design to function properly. Compared to the PIC32 Processor, the MSP430 required a lower supply voltage and included all of the periphery buses that were necessary for the project. On the other hand, the MSP430 had a much lower Flash memory capacity, lower SRAM, a lower maximum frequency, and less A/D converter channels. This project needs to have a processor that contains a lot of memory since there will be signal processing, along with a decent amount of software programming. Since the PIC32 contains a much larger memory capacity, Microchips microcontroller proved to be the best choice for the design, and the team decided to use the PIC32 microcontroller as the processor for the spectrum analyzer.

5.3.5 Analog to Digital Conversion

In order for the audio signal to be displayed on a water pump, the signal must be converted from a continuous quantity to a discrete one. Thus, it must be converted from an analog to digital format, so that it can be properly sampled. This process is done by a device called an Analog to Digital Converter. This will effectively represent the signals amplitude. For the project, it was of utmost importance that the design team ensures that the Analog to Digital conversion worked properly, and that the signal could be reconstructed. A/D converters come in several types of architectures, and there are many variations of their configurations. The type of A/D converter configuration depends on the specific process. These architectures include Integrating, Sigma-Delta, Successive Approximation, Sub ranging, and Flash. The distinctions that separate each type are their corresponding range of sampling rates (in samples/second) and Resolution (bits). Some devices such as high speed oscilloscopes require a high sample rate but not as much resolution, while other applications require high resolution but don't require high sampling rates. The design team had already determined the required sampling rate for the audio signal which was determined to be 10 kHz, and high resolution is not needed as much for the project. The design team determined that the perfect type of ADC needed was a Successive Approximation converter. The PIC32 processor has a built in 10-channel Analog to Digital converter, and utilizes the Successive Approximation Architecture. Thus, the design team decided to use the converter internal to the processor to implement the conversion; it has a 10-bit resolution and 16 input channels.

5.3.5.1 A/D Logic

A successive approximation (SAR) analog to digital converter converts a continuous analog signal to a discrete quantized signal utilizing a binary search algorithm. This type of converter is ideal for devices that use a relatively low sampling speed, low cost, and low power consumption. In addition, it has a middle resolution that varies from 10-16 bits normally. The A/D converter the design team is using has a specific variation of SAR architecture.

First, the filtered input audio signal will enter the processor directly to the internal ADC. The signal will then be sampled. In this step, acquisition, the analog input pin is attached to a Sample and Hold Amplifier (SHA). It is then sampled, and the sample voltage is the same as the input voltage. The sampling can be controlled automatically or manually. For the project, the design team will let the processor automatically set it. In this mode, the next sampling will start automatically after the previous sample has finished converting

The next part, conversion, converts the analog sample voltage into a binary representation. the ADC consists of four individual components: the SHA which holds the input voltage, an internal reference digital analog converter (DAC) that takes an input of the reference voltages and SAR output, an analog voltage comparator that compares the input voltage in the SHA to the output of the internal DAC, and finally the actual Successive Approximation register circuit that supplies an approximate digitization of V_{in} to the DAC. A block diagram in Figure 5.3.5.1-1 shows an elementary dataflow of the SAR ADC conversion process.

V_{ref+} is set to 3.3V and V_{ref-} is set to 0V which are the rails of the processor. The SAR register is at first initialized so that the MSB (most significant bit) is set to logic 1 and the rest of the bits are set at logic 0. This will force the DAC output (V_{DAC}) to be $\frac{V_{ref+}}{2}$. This is then sent to the comparator to compare if V_{in} is less

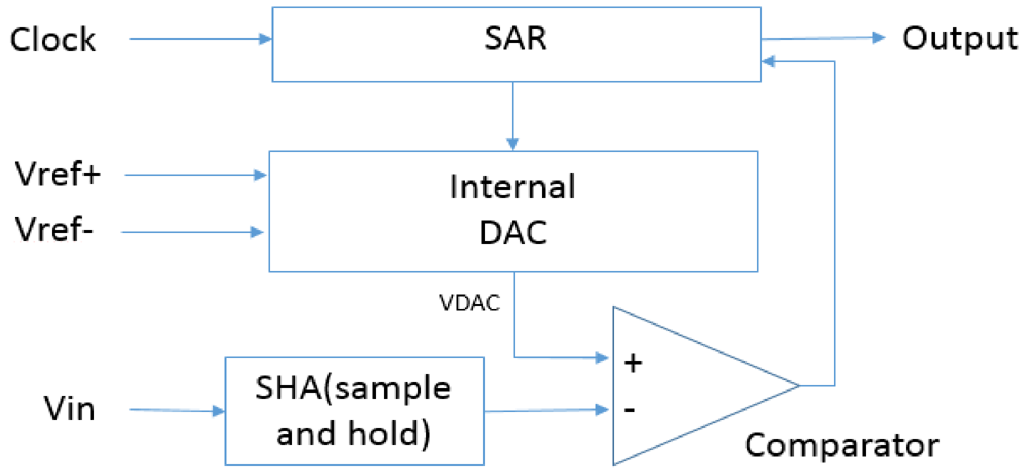


Figure 5.3.5.1-1: Block Diagram Successive Approximation AD Conversion

than or greater than V_{DAC}. If V_{in} is greater, the comparator output is 1. If V_{in} is less than V_{DAC}, the output of the comparator is logic 0. This is an application of the binary search algorithm. The SAR then moves to the next bit, sets it to 1, and the algorithm is repeated. This is done until the LSB is converted. Consequently, this 10-bit word is stored in the register and the A/D conversion is finally complete. The conversion speed of this ADC is 1 Msbps. A detailed block diagram of the ADC is shown in figure 5.3.5.1-2 and shows where each data bit is sent.

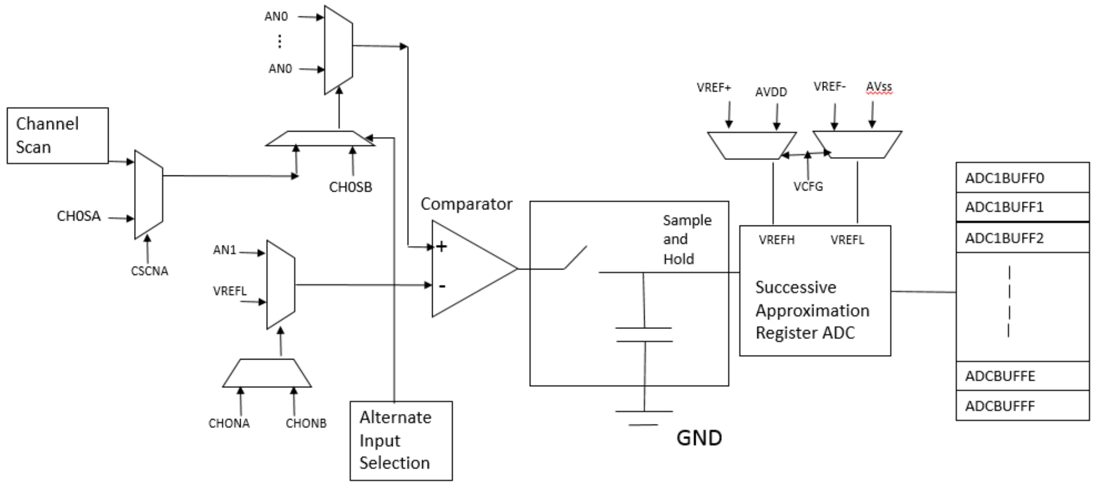


Figure 5.3.5.1 2: Block Diagram of PIC32 ADC

The PIC32 10-bit ADC can have up to 16 analog input pins, named AN0 through AN15. Also, there are two additional analog input pins for external voltage reference pins if the design team needs to use them. The actual number of analog input pins and external voltage reference inputs will depend on the specific PIC32 model. These analog inputs are connected by utilizing two multiplexers, in which the outputs are sent to the SHA. These multiplexers can be switched between the two sets of analog inputs. The control register determines which analog input channels will be converted. In addition, the 10-bit ADC is connected to a 16-word buffer. Each 10-bit result is converted to one of the eight 32-bit output formats when read from the buffer.

5.3.6 PIC32 Core Architecture (Harvard)

All Central Processing Units are based off of some sort of computer architecture that determines its set of rules to describe its functionality, organization, and implementation. Every CPU must have specific components to operate properly as a system. These components are the processor, which houses the Control Unit, which decodes instruction bits to control instruction execution, and the Arithmetic Logical Unit to calculate arithmetic and logic functions, memory to store values represented as bits, an input and output, and finally, a data path that performs data processing operations, and includes registers and memory busses. An architecture can be a programming model, a specific ISA (Instruction Set Architecture), or a type of logic design.

The architecture of the PIC32 microcontroller uses is Modified Harvard Architecture. This specific architecture is structured in that the instruction and data memory are physically separate. In addition, the signal pathways are separate as well. Thus, there is no need to make the instruction and data memory share characteristics, and their internal structures can differ. For example, timing, memory addresses, and word widths do not have to be the same. In addition, this unique architecture additionally allows the instruction memory to be accessed as if they were data memory. Because of this unique structure, the central processing unit of the PIC32 can both read an instruction and perform data memory processes simultaneously. This can be done without the need for a data cache. Consequently, the CPU can be much faster due to the fact that there is not a single memory path. Figure 5.3.6 shows a block diagram of the basic architecture of the PIC32 processor.

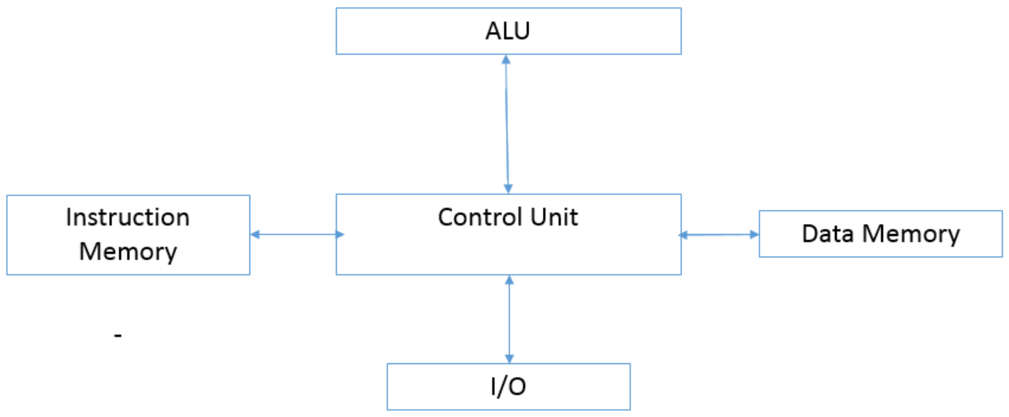


Figure 5.3.6: Computer Architecture of PIC32 Processor

Since the PIC32 contains flash and SRAM memory, this microcontroller takes advantage of modified Harvard Architecture by processing instruction and data access in parallel. Since there is separate storage, the program and data memories can use different word widths, along with performing instruction prefetching and other processes simultaneously. This allows the processor to be high speed, which is necessary for the signal processing algorithm.

5.3.7 PIC32 Schematic

The design team picked the PIC32MV250F128B model of the microcontroller for the project. It is a 28 Pin dip package. Figure 5.3.7-1 shows the pin mapping of the PIC 32, and the section below describes the functions of each pin. This custom template was used to design the schematic for the design.

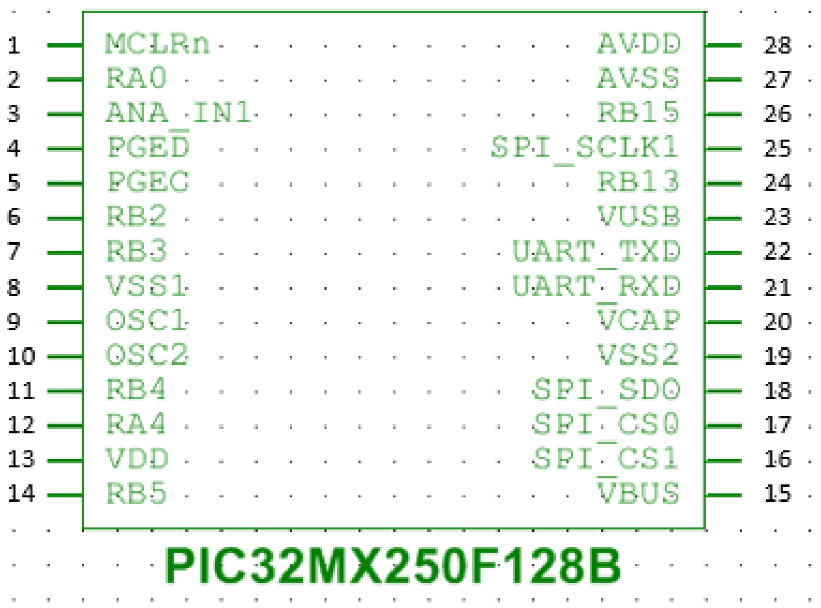


Figure 5.3.7-1: PIN mapping of PIC32

Pin 1, MCLRn, or the master clear pin, provides two functions: device reset, and device programming and debugging. If set to logic low, the device is reset. This logic is driven by the programmer. This pin is required to have a resistor and capacitor attached, and the specific values of these components are adjusted based on the PCB requirements. Figure 5.3.7-2 shows the hardware team's design of the sub circuit attached to the MCLRn pin.

This is the identical configuration of the sub circuit recommended by the datasheet. R2 serves as a current limiting resistor in case of MCLRn pin breakdown, and C1 is sized to prevent unintentional resets from glitches. S2 is a button and will serve as a reset button on the PCB. This button is how the programmer manually resets the device. As shown, Vcc is the 3.3V power supply used to power the chip. In addition, the Reset_n, wire goes to the DAC to the CLR pin. Thus, if the microcontroller is reset, the DAC will reset as well.

All of the RA and RB pins, which are pins 2, 6, 7, 11, 12, 14, 24 and 26 can be custom designed pins. The hardware team only utilized several of these pins, and the unused ones, RB13, RB5, and RB15 were unconnected. Pins RB1, RB3,

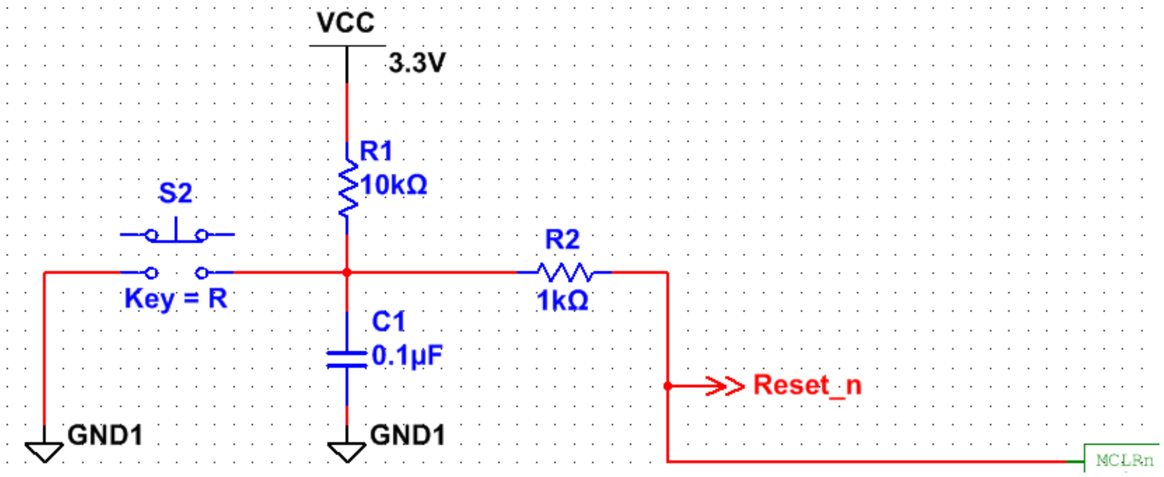


Figure 5.3.7-2: MCLRn PIN Sub Circuit

and RB4 were designated to be the connection between the chip and the LED shift register. More specifically, RB2 is designated for the LED data, RB3 is the latch enable, which latches data into the output shift register, and RB4 is the LED shift register clock, which shifts at every rising edge of the internal clock. Finally, RA0, or pin 2 was programmed to contain an LED. The LED's purpose is to always remain on when the chip is powered on. The LED will be placed on the board and serves as a visual aid when troubleshooting. Figure 5.3.7-3 is the schematic for the sub circuit coming out of pin 2.

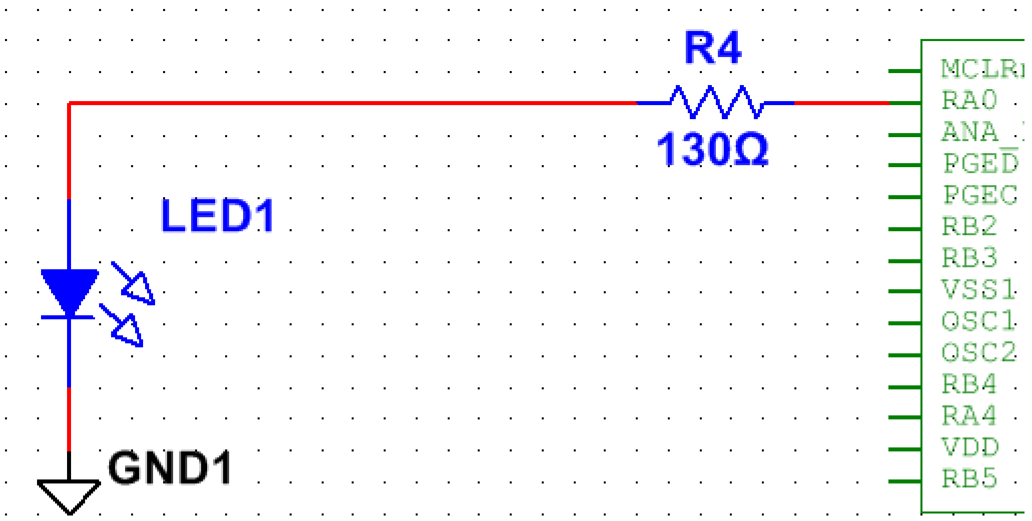


Figure 5.3.7-3: PIN 2 Sub Circuit

LED1 is the notification LED to determine if the chip is powered, and R4 is a current limiting resistor. Since the voltage coming out of the microcontroller is 3.3V, and most LEDs operate at around 2 V, to ensure the LED didn't overheat and break, the voltage had to be reduced to operate at LED voltages. Thus, the current limiting resistor's value was chosen so that there would be a voltage drop of 1.3 V to bring down the voltage to 2 V. According to the datasheet, the current coming out of the processor was estimated to be 10mA and consequently, the correct resistance was given by Ohm's law (E5.3.7-4).

$$R = \frac{V}{I} \tag{E5.3.7-1}$$

If $V = 1.3V$ and $I = 0.01A$,

$$R = \frac{1.3}{.01} = 130\ \Omega \tag{E5.3.7-2}$$

Thus, the value of R_4 was chosen to be 130Ω. Next, Pin 3 was assigned to be the analog input pin. Thus, the analog audio signal output of the level shifter will be connected to this pin on the microcontroller.

Pins 4 and 5, or the PGED and PGEC pins, are used for in circuit serial programming (ICSP) and debugging purposes. These pins will be connected to a Programming Debug Header. This is where the development kit for debugging will be connected to interface with the chip. The design team implemented a header so that the debugging port could be placed on an easy to access location on the PCB. .Figure 5.3.7-4 shows the schematic for the debug header which is connected to the PGED and PGEC pins.

Thus, the Programming Debug Header is connected to a Ground wire (GND1), a 3.3V power supply pin (VCC), and the PGED and PGEC pins on the microcontroller. This implementation will allow for an easy to access connector on the PCB.

Pins 8 and 19, or V_{SS} and V_{SS2}, are used as the ground references for logic and I/O pins. It must be connected at all times. As ground references, these pins will be connected to ground wires. Pin 27, or AVSS, is the ground reference for analog modules. This is used for the internal A/D converter. Likewise, this pin will be connected to a ground wire.

Pins 9 and 10, or OSC1 and OSC2, are the oscillator crystal input and output, respectively. These are the external oscillator pins for the microchip. The oscillator serves as the internal clock of the chip, and its configuration and component values must be determined manually. Figure 5.3.7-5 shows the configuration of the oscillator sub circuit recommended by the PIC32 datasheet.

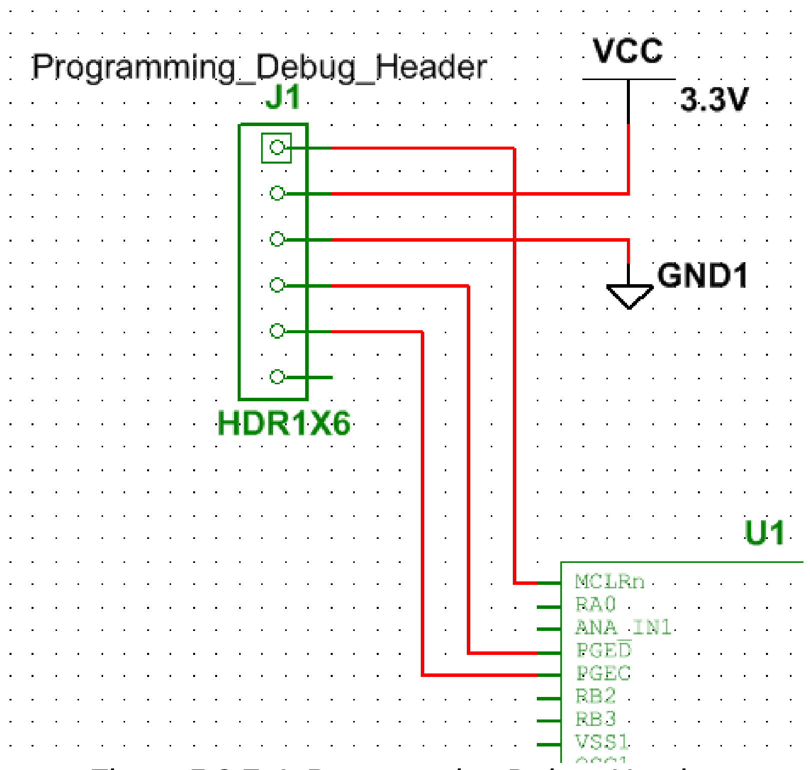


Figure 5.3.7-4: Programming Debug Header

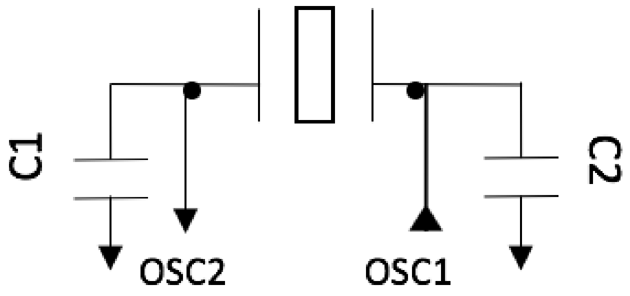


Figure 5.3.7-5: Crystal Oscillator Recommended Configuration

C1 and C2 are the loading capacitance values, which were determined to be 15pF. This is due to the following equation (E5.3.7-7) given in the datasheet

$$CLOAD = \left\{\frac{([CIN + C1] * [COUT + C2])}{[CIN + C1 + C2 + COUT]}\right\} + estimated\ stray\ capacitance \qquad \text{(E5.3.7-7)}$$

Using datasheet specified values,

- CIN = PIC32_OSC2_Pin Capacitance = ~5pF
- COUT = PIC32_OSC1_Pin Capacitance = ~5pF
- If C1 = C2 = XTAL, recommended loading capacitance = 15 pF
- Estimated PCB stray capacitance = 2.5 pF

$$CLOAD = \left\{ \frac{([5 + 15][5 + 15])}{[5 + 15 + 15 + 5]} \right\} + 2.5$$

$$CLOAD = \left\{ \frac{([20][20])}{[40]} \right\} + 2.5$$

$$CLOAD = 10 + 2.5 = 12.5\,pF \approx 15\text{ pF}$$

Thus, the loading capacitor values were determined to be 15pf. Figure 5.3.7-6 shows the schematic for the oscillator circuit connected to pins 9 and 10.

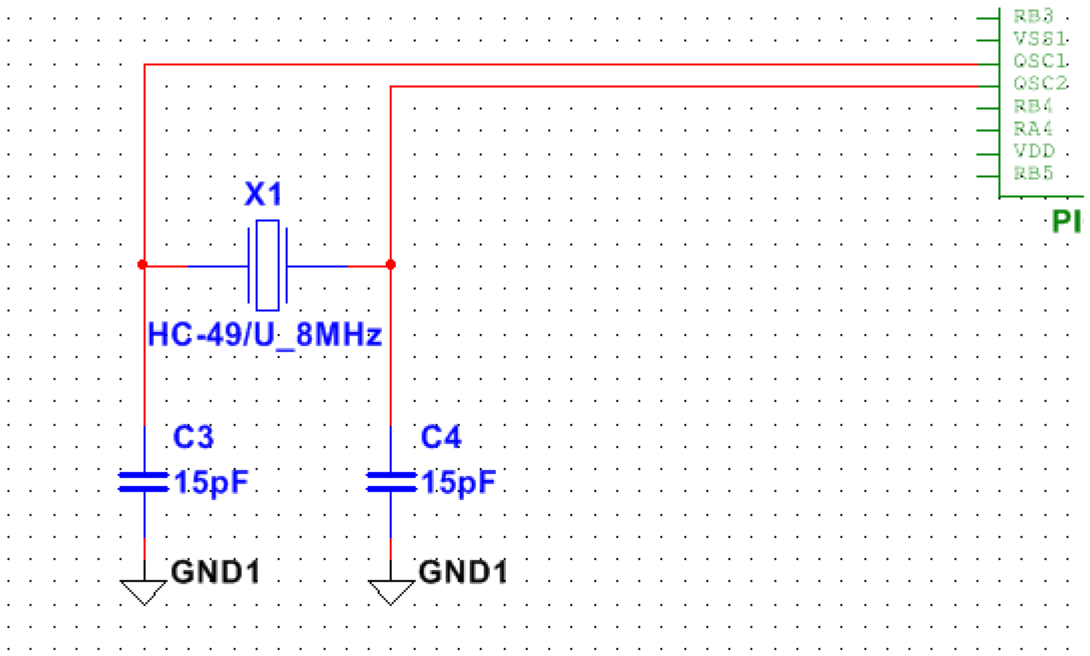


Figure 5.3.7-6: Crystal Oscillator Configuration

Pin 13, or V_{DD}, is the pin that connects to the positive power supply for peripheral logic and I/O pins. Pin 28, or AV_{DD}, connects to the positive power supply for analog modules, such as the A/D converter. These pins are both going to be connected to the 3.3V power supply.

Pin 15, or VBUS, which is the pin connected to the USB bus power monitor. Since this project does not require this component, this pin will not need to be used, and pin 15 will remain open.

Pins 16, 17, 18, and 25 were assigned to be the SPI outputs of the processor and consequently the input of the DAC. Thus, pins 16, 17, 18, and 25 were named SPI_CS1, SPI_CS2, and SPI_SD0, and SPI_SCLK1 respectively. SPI_SD0 connects to the data input of the DAC, meaning the audio signal will be the output of this pin. SPI_CS1 and SPI_CS2 are the select pins of each of the 2 DAC devices. Thus, if only the first DAC needs to be used, the CS1 pin will activate, and vice versa. The SPI_CLK1 pin will become the DAC Serial Interface Clock Input. This is to ensure that the DAC and PIC32 are synchronized properly.

Pin 20, or VCAP, is the pin that connects to the capacitor for the internal voltage regulator. A low ESR capacitor is required on this pin either ceramic or tantalum, and is used to maintain stability of the internal voltage regulator. Thus, a 10uF capacitor was attached to pin 20. This value was chosen because this was the recommended value as specified by the datasheet, as the typical voltage on the VCAP pin is 1.8V.

Pins 21 and 22 were assigned to be the connection between the chip and the TTL to RS232 converter. Thus, these pins were named UART_RXD and UART_TXD, respectively. The UART_TXD pin is the TTL output of the PIC32 which goes to the input of the TTL to RS232 converter. The UART-RXD pin connects the output of the TTL to RS232 converter to the input of the processor.

In order to minimize noise on the chip, decoupling capacitors must be placed on the V_{DD}, V_{SS}, AV_{DD}, and AV_{SS}. The values for the capacitors are 0.1 μF and they are recommended to be of the ceramic type, with low ESR ratings. Thus, the design team ensured that these capacitors were placed on the schematic.

Figure 5.3.7-7 shows the fully completed schematic of the PIC32 microcontroller with all of its correct connections between the other components of the design.

5.3.8 SPI

Serial Peripheral Interface, or SPI, is a type of synchronous serial interface used for communication with external devices. The PIC 32 chip uses SPI as its communication interface specification. Since the PIC32 utilizes an SPI module, it includes features such as master and slave modes, four different clock format types, SPI protocol support, user configurable 8, 16, and 32 bit data width, and programmable interrupts. Since SPI is the interface the PIC32 uses, the output of the PIC32 device will leave the device in SPI format. Thus, the next stage of the design, the component that will take in the frequency binned audio signal, will have to support SPI as an input to transfer correctly. It is imperative to ensure that the communication between these two devices work properly, and so SPI compatibility is a requirement for the input device of the next stage of the design, which in this project, is the Digital to analog converter.

5.3.9 UART

The Universal Asynchronous Receiver Transmitter, or UART, is a type of serial I/O module that is supported in the PIC32 Microcontroller. UART is a computer hardware device that can translate between serial and parallel forms. It is an

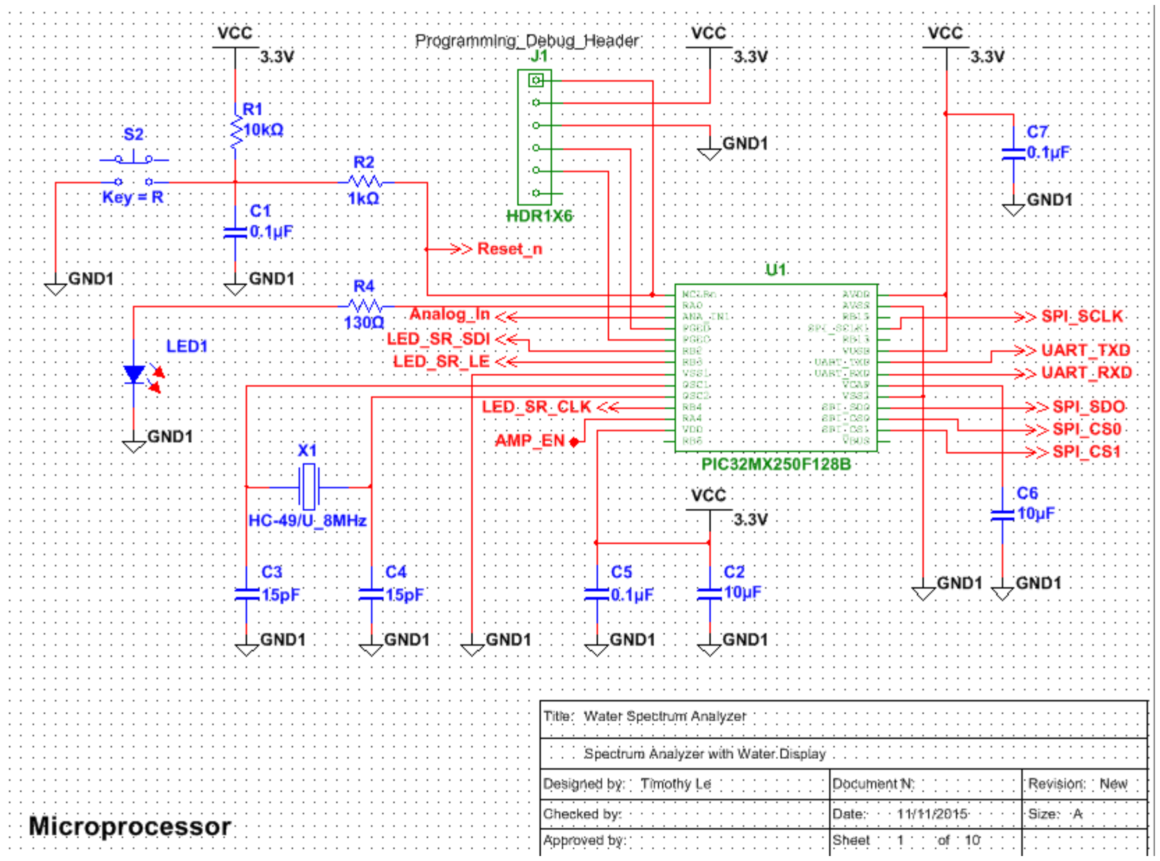


Figure 5.3.9-1: Completed PIC32 Processor Schematic

asynchronous communication channel that can communicate with external devices, including personal computers. The UART module in the PIC32 consists of 3 hardware elements: the baud rate generator, an asynchronous transmitter, and an asynchronous receiver and IrDA encoder/decoder.

In order to transmit software code onto the PIC32 microcontroller, the design team will utilize the internal UART module to interface with an external personal computer. In other words, the software team will test and write the programming code using a personal computer, and will consequently transfer the code to the PIC32 chip via UART. But in order to do this, it must first be converted into another form using a protocol.

5.3.10 RS-232

RS-232 is one of the standards for serial communication of data. It is commonly used in computer serial ports and older personal computers. The standard RS-232 is defined by: electrical signal characteristics (voltage levels, signal and slew rates, etc.), mechanical characteristics of its interface, functions of each circuit in the connectors, and subsets of interface circuits. In modern technology, RS-232 has been largely replaced by USB, another industry standard. USB boasts higher transmission speeds, more compact connectors, and lower voltage swings, which makes it a far superior choice than RS-232. But, the design team decided to use RS-232 regardless. The design team decided this because USB is much more complex, and requires more software. It is also much more difficult to provide low level interfaces for. On the contrary, RS-232 is intended to be used for hardware control, and is consequently easier to provide a low level interface for. Since the design is a relatively simplistic one, there is no need for the superior speed of USB. Another important reason why RS-232 was chosen was the fact that RS-232 and UART are very commonly used in conjunction in serial communication. Due to its common pairing, the design team determined it would be best to use something that is familiar. Thus, the use of RS-232 was justified, and it was then implemented for the computer to microcontroller interface.

5.3.11 TTL to RS232 Conversion

In order to correctly create a connection between the computer and microchip, there would need to be a line driver to successfully generate a high enough voltage level. The Analog Devices ADM3202 line driver. This device was ideal for the design due to its functionality, along with its ability to be powered from a single 3.3V power supply. This was perfect for the design, due to the fact that the project uses a 3.3V power supply as the low power voltage source. The ADM3203 component is a 16 pin DIP package, and Figure 5.3.11-1 shows the pin locations on the chip, which will be mapped on the actual PCB. Table 5.3.11-1 describes the functions of each of the 16 pins.

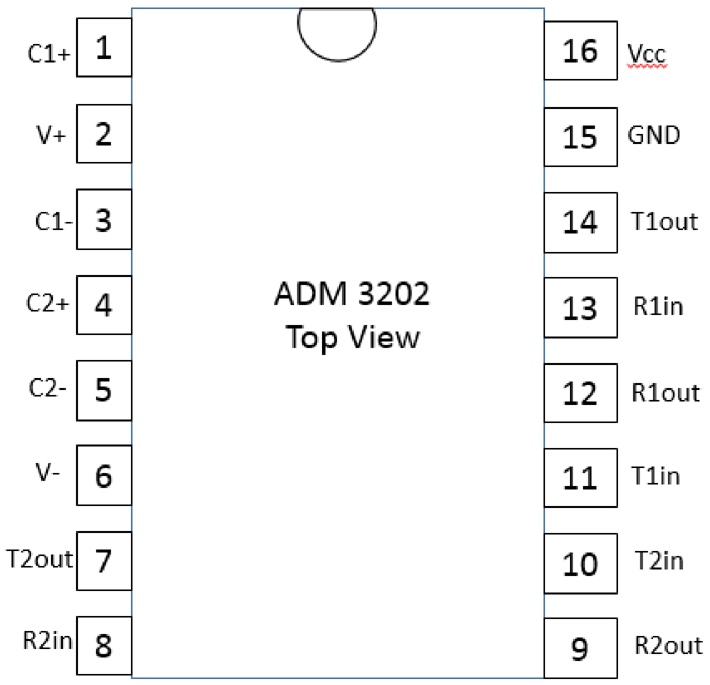


Figure 5.3.11-1: PIN Mapping of ADM3202

Pin Number	Name	Function
1	C1+	External Capacitor 1 is connected between pin 1 and pin 3. A 0.1 μF capacitor is recommended
2	V+	Internally Generated Positive Supply
3	C1-	External Capacitor 1 is connected between pin 3 and pin 1. A 0.1 μF capacitor is recommended
4	C2+	External Capacitor 2 is connected between pin 4 and pin 5. A 0.1 μF capacitor is recommended
5	C2-	External Capacitor 2 is connected between pin 4 and pin 5. A 0.1 μF capacitor is recommended
6	V-	Internally Generated Negative Supply
7	T2OUT	Transmitter (Driver) Outputs. These are RS-232 signal levels
8	R2IN	Receiver Inputs. These inputs accept RS-232 signal levels. An internal 5 kΩ pull-down resistor to GND is connected on each input.
9	R2OUT	Receiver Outputs. These are CMOS output logic levels.
10	T2IN	Transmitter (Driver) Inputs. These inputs accept TTL/CMOS levels.
11	T1IN	Transmitter (Driver) Inputs. These inputs accept TTL/CMOS levels.
12	R1Out	Receiver Outputs. These are CMOS output logic levels.
13	R1IN	Receiver Inputs. These inputs accept RS-232 signal levels. An internal 5 kΩ pull-down resistor to GND is connected on each input.
14	T1OUT	Transmitter (Driver) Outputs. These are RS-232 signal levels
15	GND	Ground Pin
16	Vcc	Power supply input

Table 5.3.11: PINs and Functions of ADM3202

According to each pin and function, the design team created a schematic to implement this line driver. Figure 5.3.11-2 shows the schematic, designed by the hardware team.

The design team created the schematic by designing each pin properly to emulate the recommended configuration. The T1IN and R1OUT pins were connected to the UART pins in the PIC32, respectively. The four capacitors were placed as recommended by the datasheet and were consequently connected in series from the C1+ to C1- pins, and from the C2+ to C2- pins. Since there was no internally generated positive and negative supply, they were grounded with a 0.1uF decoupling capacitor. The VCC pin was connected to the 3.3V power supply and attached to a 0.1uF decoupling capacitor to reduce noise. Finally, the

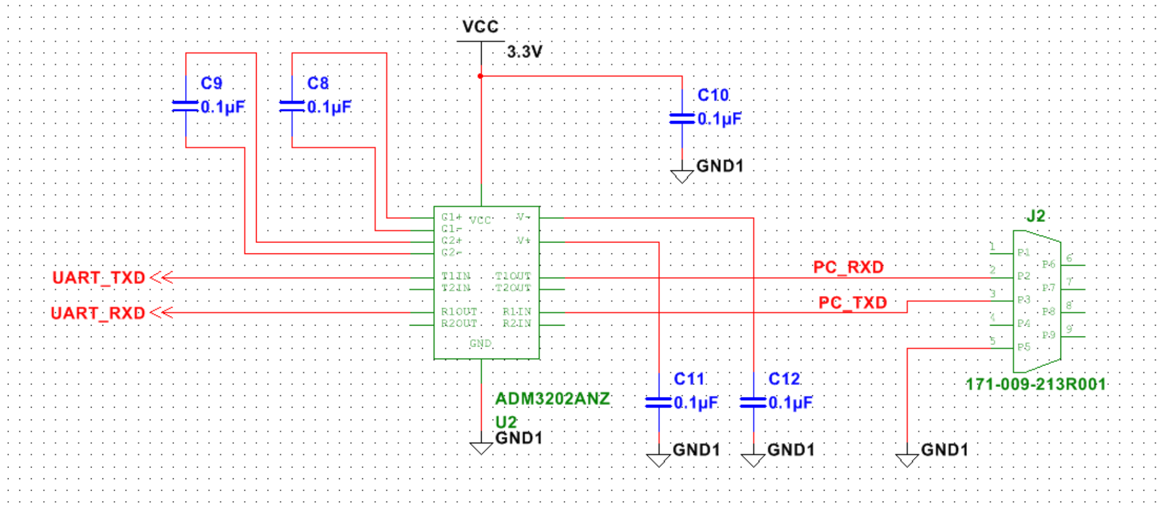


Figure 5.3.11-2: UART to RS232 Conversion Schematic

T1OUT and R1IN pins (which accept RS-232 levels) were connected to a standard connector. This would then be connected to a pre designed RS-232 to USB converter, so that it could connect properly to a personal computer. The design team did not possess any personal computer that contained an internal RS-232 port, so the converting device was necessary. Thus an interface from a computer to the microchip was successfully designed.

5.3.12 Memory

The PIC32 processor features 4GB of virtual memory address space. This address space includes program and data memory, special function registers, and configuration registers. The program and data memories can be partitioned, and the data memory can be made executable. This results in the ability of the PIC32 to be executed from data memory. The device also includes 32-bit data width, separate user and kernel mode address spaces, flexible RAM and Flash partitioning, and cacheable and non-cacheable address regions. In addition, there is a separate boot flash memory, which allows the user to store and protect code for the device.

The PIC32 microcontroller implements two types of address spaces, virtual and physical. All of the hardware resources such as the program and data memories, and peripherals are located in the physical address spaces. The Virtual address space is used only by the CPU to fetch and execute instructions. The Physical address spaces are used by the peripherals and access memory separately and independently from the CPU. This is where the code of the project will be stored. Figure 5.3.12-1 shows the physical memory map of the microcontroller, along with the memory addresses associated with each part.

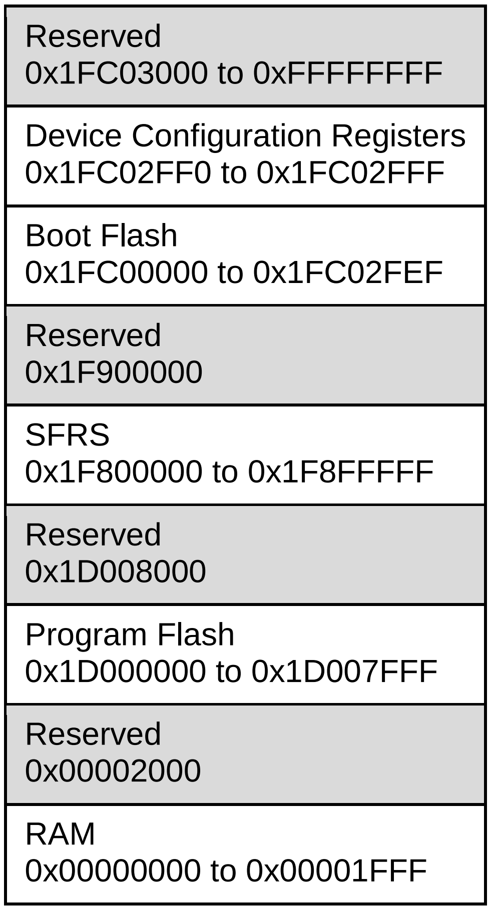


Figure 5.3.12-1: Physical Memory Map of PIC32

Thus, the RAM uses memory addresses 0x00001FFF to 0x00000000, the flash can use addresses 0x1D007FFF to 0x1D000000, the special function registers can use addresses 0x1F8FFFFF to 0x1F800000, the Boot flash can use addresses from 0x1FC02FEF to 0x1FC00000, the device configuration registers utilize the space from 0x1FC02FFF to 0x1FC02FF0, and the rest of the addresses are reserved.

The PIC32 virtual address space is quite large, at 4 GB. This virtual address space is divided into two equal regions, the user and the kernel space. Each space is reserved for its respective mode. Kernel mode is reserved for the lowest level and most trusted functions of the processor, since it has complete and unrestricted access to the hardware. The user mode cannot do this, and all of the code for the design will be in user mode. This is beneficial in that the code in this project will require troubleshooting, and crashes in user mode are easily recoverable, while crashing in kernel mode will be catastrophic. Figure 5.3.12-2 shows the memory map of the user/kernel address segments.

Kernel Segments (KSEG0,1,2,3) 0x80000000 to 0xFFFFFFFF
User/Kernel Segment (USEG/KUSEG) 0x00000000 to 0x7FFFFFFF

Figure 5.3.12-1: Memory Map of User/Kernel Address Segments

The lower 2 GB of space is called the USEG/KUSEG, and all user mode applications will be stored and executed in this segment. The upper 2GB of space is called the Kernel segment, and all kernel mode applications reside here. The kernel space is divided into four 512 MB segments, called KSEG0, KSEG1, KSEG2, and KSEG3. Only the kernel mode applications can access these segments. This also includes all of the peripheral registers. The PIC32 only uses the KSEG0 and KSEG1 segments, and the boot flash memory, program flash memory, and the data RAM memory are accessible from either KSEG0 or KSEG1, and the peripheral SFRs are accessible from only KSEG1. The KSEG0 and USEG/KUSEG segments are cacheable, while the rest are not.

5.3.12.1 SRAM (Memory Storage)

SRAM is needed for this project to perform the calculations from the Fast Fourier Transform code and having SRAM allows these calculations to be done quickly. Without SRAM, these calculations will have to be done through internal registers, which is much more limited and will not work for this design. The PIC32 contains 32 KB of SRAM, according to its datasheet, which is more than enough memory required for this project.

The RAM memory is divided into four partitions: Kernel Data, Kernel Program, User Data, and User Program. These partitions start from the base of the data RAM, and in order to execute from data RAM, the partitions must be defined manually. In order to do this, the following registers, BMXDKPBA, BMXDUDBA and BMXDUPBA, must be programmed. Figure 5.3.12.1-1 shows a partitioning diagram for the RAM, which will be modified to fit the design's needs.

The Grey areas are unused addresses that can be partitioned. The coding for this specific project will be done in kernel mode. This is because In addition, there is one application (the user) being added to the processor, and it will be beneficial to have unrestricted access to the hardware. In addition, user mode must deal with overhead layers, and directly interfacing with kernel mode will increase efficiency. Due to these advantages, the code will be done in kernel mode, not user mode. Consequently, since all of the memory will be allocated for

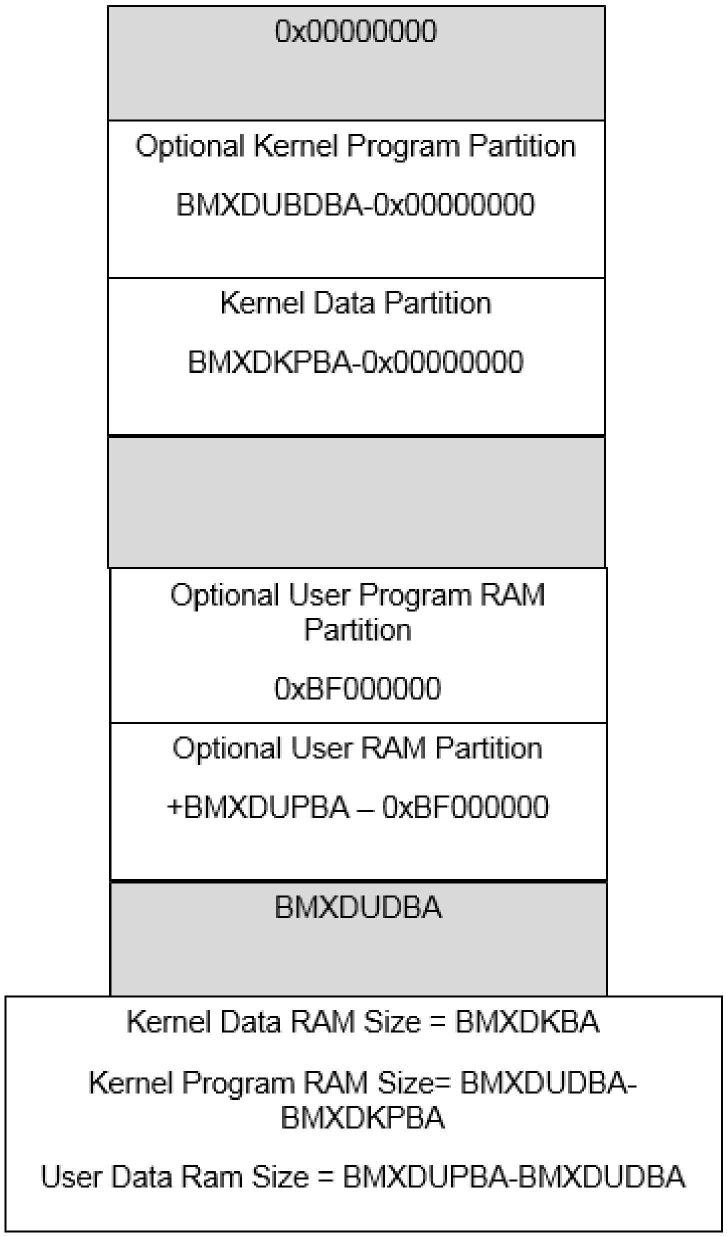

0x00000000
Optional Kernel Program Partition
BMXDUBDBA-0x00000000
Kernel Data Partition
BMXDKPBA-0x00000000
Optional User Program RAM Partition
0xBF000000
Optional User RAM Partition
+BMXDUPBA – 0xBF000000
BMXDUDBA
Kernel Data RAM Size = BMXDKBA
Kernel Program RAM Size= BMXDUDBA-BMXDKPBA
User Data Ram Size = BMXDUPBA-BMXDUDBA

Figure 5.3.12.1-1: RAM Partitioning

kernel mode, the partitioning will maximize the size of the Kernel data and program partition, and the user program RAM partition will be minimized. This will be done through the modifying of the BMXDKPBA, BMXDUDBA and BMXDUPBA registers. To do this, one of these registers will be set to zero. According to the datasheet, doing this will assign the entire RAM to kernel data RAM, which is the desired outcome. Thus, the memory partition will look like Figure 5.3.12.1-2 when visualized.

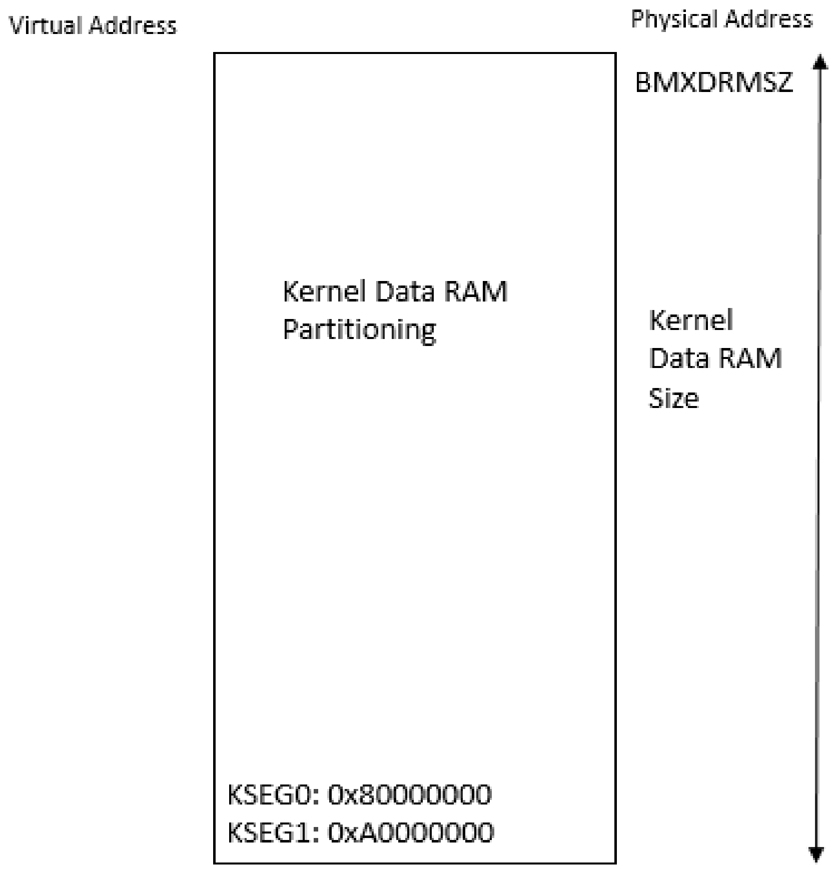


Figure 5.3.12.1-2: Modified RAM Partition for Custom Design

As the figure shows, all of the user mode partitions have been eliminated, and the entire RAM is designated for the kernel data.

5.3.12.2 Flash (Software Storage)

Flash memory is needed for this project to store the actual Fast Fourier Transform algorithm code. This code will need to stay in the processor even if the device is reset. Since flash memory is nonvolatile, meaning that the device can still retrieve the stored information after it has been turned off and back on, flash memory is a necessity for this project.

Like the SRAM, the program flash memory can be partitioned for User and Kernel mode programs. Figure 5.3.12.2-1 shows the partitioning map for the program flash.

As seen in Figure 5.3.12.2-1, the flash partition is divided into two sections, the Flash Partition for Kernel Program, and the Flash Partition for the User Program. Like the SRAM section, the flash partition will be dedicated to the kernel program, since User mode will not be used. Thus, all of the flash memory will be

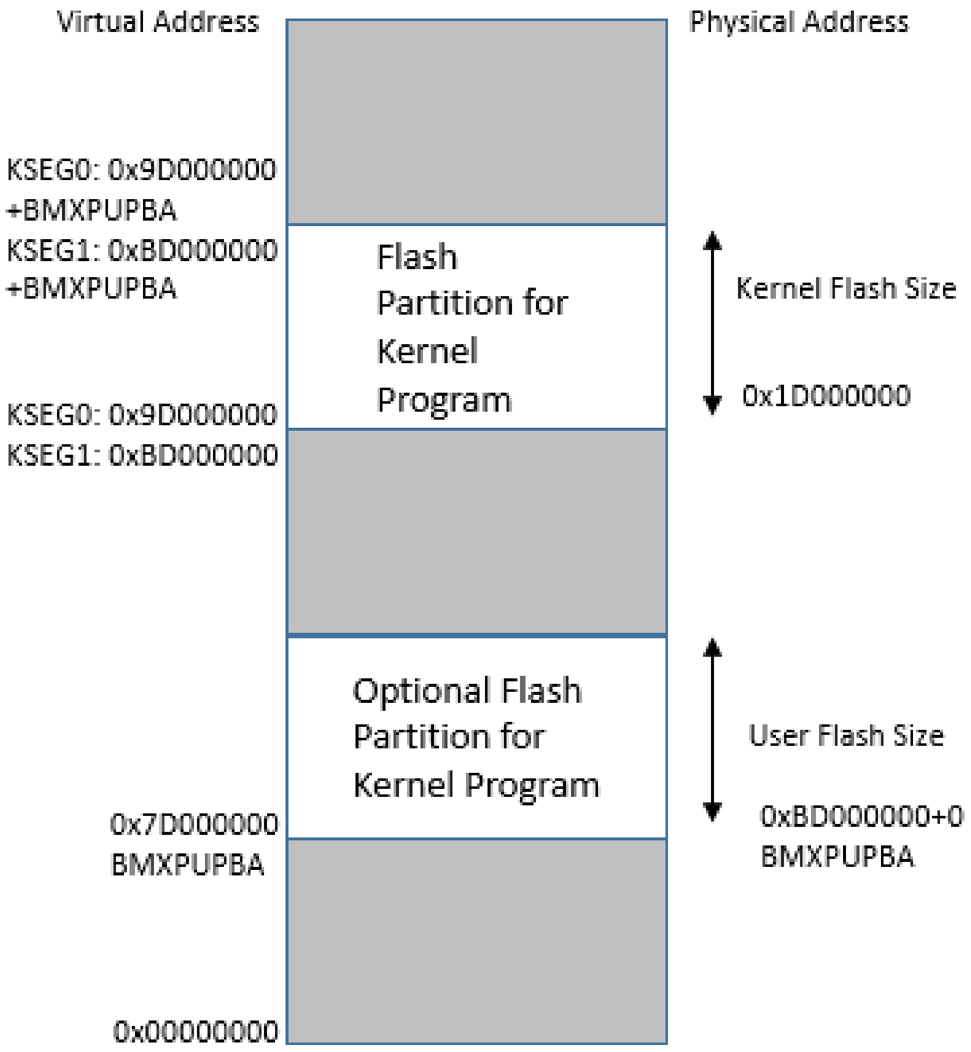


Figure 5.3.12.2-1: Program Flash Partitioning

designated to the Flash Partition for the Kernel Program. To do this, the register BMXPUPBA will be set to 0. According to the datasheet, setting this register to 0, the entire program flash memory will be mapped to kernel mode program space.

5.4 Buttons

The following section describes the buttons of the project. These buttons described below are on the smartphone application for user modifications of the display. These include pump and LED settings.

5.4.1 LED Settings

Some of the LED settings that the team wishes to have are the ability to control the colors, the rate at which they flash, whether or not they will be changing colors, patterns, and the power to turn them on and off. This way, it is customizable and users have complete jurisdiction over the lighting output. The group plans to have buttons on the mobile application that will allow these settings.

5.4.2 Pump Settings

A few of the pump settings that the group hopes to have are controlling pump flow rate, up ramp, down ramp, and turning them on and off. These functions will also be accessible on the mobile application.

5.5 Digital-to-Analog Converter

Once the audio signal has been digitally processed and converted to the frequency domain, the signal must be sent to the water pumps to display. In addition, there must be 16 separate independent outputs, each corresponding to a water pump. Each output is assigned to a specific frequency range, and they are all uniform. The signal must be converted back to an analog one in order to be effectively displayed on the pumps in continuous time. Since the output of the microcontroller is a digitized signal, it must be externally converted to an analog signal. Thus, the design team needed to design or acquire a Digital to Analog Converter device with 16 outputs. Hence, the design team will need 16 individual converters preferably compacted to one or two devices. Like all the other components, the DAC must be in a DIP package.

5.5.1 LTC1665 Micro power DAC

For the design, the design team decided to use a pre-designed Digital to Analog Converter for the spectrum analyzer. This device was required to have a functionality to produce 16 outputs, along with being able to function at low power. Thus, there will need to be 16 individual converters. In addition, the DAC must support SPI, since that is the format of the output of the audio signal from the processor. In the search, the design team found an ideal DAC that fit all of the specifications. This is the Linear Technology LTC 1665 Micro-power Octal 8-bit DAC. This device integrates eight individual digital to analog converters that are serially addressable (SPI). The following table 5.5.1 shows all of the relevant specifications of the device for the project.

LTC1665	
Supply Voltage	2.7-5.5 V
Output Pins	8
Supply Current (per DAC)	56 uA
Maximum Junction Temperature (C)	125°
Input Interface	SPI
Packaging	DIP

Table 5.5.1: Relevant Specifications for LTC1665

The 2.7 – 5.5V supply voltage fits with the 3.3V power supply for the low power specifications. Since there are 8 DACs per device, the design team will need two of them to allow all outputs to be displayed.

Pin 1 represents the Ground. This pin will obviously be set to ground. Pins 2-5 and 12-15 are the outputs of each Digital to Analog Converter in the LTC1665. The output voltage range for these will be $\left(\frac{255}{256}\right) * V_{ref}$ .These pins will thus be the output of the device and consequently be sent to the power drivers for the motors. Pin 6, REF, is the pin for the reference voltage input. This ranges from 0 to the supply voltage, which in the case is 3.3V. Pin 7, CS/LD, is the serial interface Chip Select/ Load Input. If set at logic 0, the SCK pin is enabled for shifting data on the DIN pin. If set to high, SCK is disabled and the data is loaded from the register to the corresponding DAC register, consequently updating the output which is now in analog. In other words, if CS/LD is set to low, it will convert, if it is set to high, the signal will not be sent through. Pin 8, SCK, stands for the Serial Interface Clock Input. Pin 9 stands for Din. This is the Serial Interface Data Input. Data on this pin is shifted into the register (16-bit) on every rising edge of the Serial Clock (SCK). Pin 10, Dout, stands for the Serial Interface Data Output. Data on this pin will appear after 16 positive SCK edges after application to Din. This pin will be daisy chained to the second LTC1665, so that all outputs can be represented. Pin 11 is the CLR pin, which is the Asynchronous Clear Input. On the falling edge of this signal, all of the internal registers are cleared. This is the reset pin. Finally, Pin 16 is Vcc, which is the Supply Voltage Input. For the project, Vcc will be 3.3 Volts, and this will come from the low voltage part of the power supply.

5.5.2 DAC Schematic Design

As stated before, the design team is utilizing two DACs in parallel so that all 16 outputs can be represented. In order to represent all the outputs correctly, the design team will daisy chain the DACS and will use a select input to set which DAC will activate. Figure 5.5.2 is the schematic design for the digital to analog conversion.

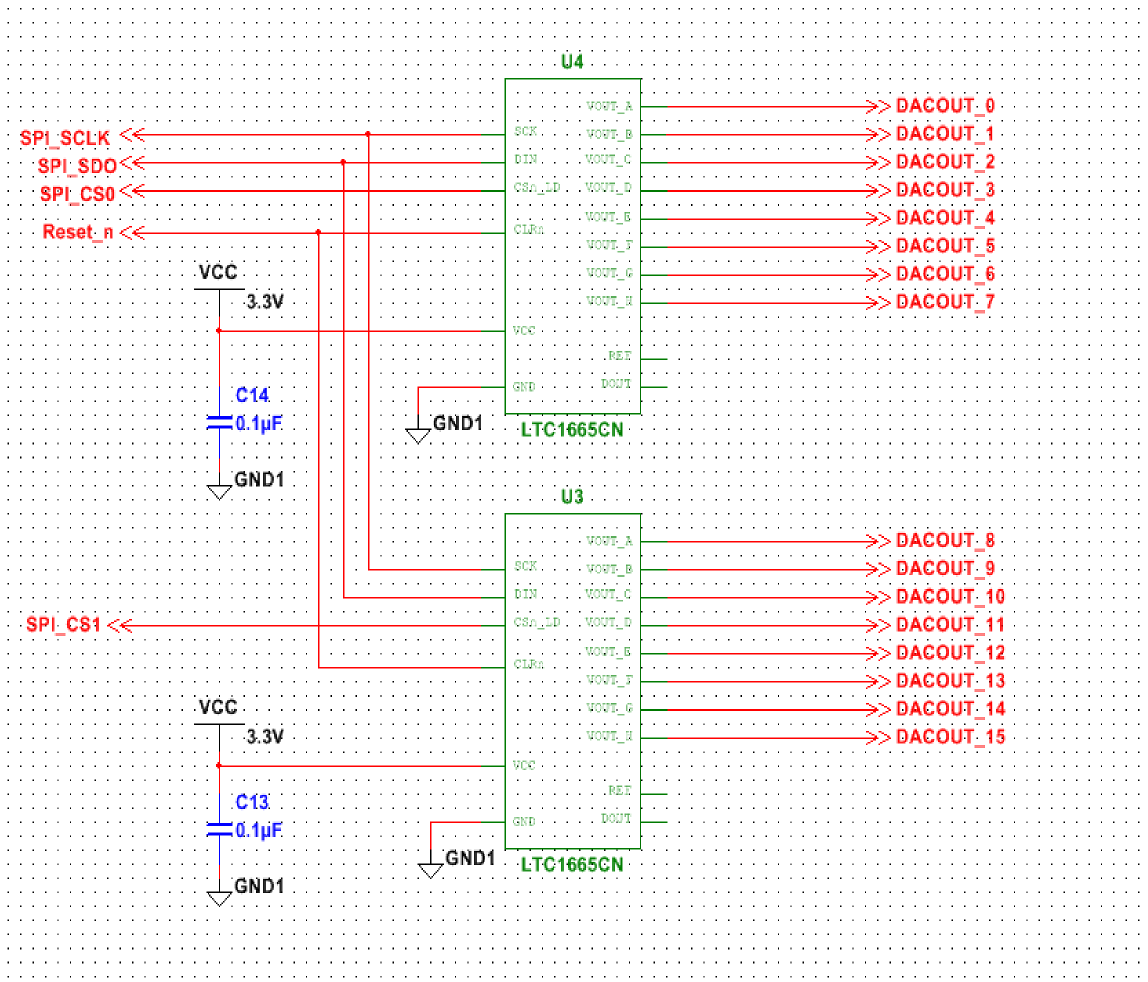


Figure 5.5.2: LTC1665 16 bit DAC Schematic

The first DAC will convert the outputs 0-7 and the second will convert outputs 7 through 15. These sequential outputs are then sent to each corresponding motor driver. The SPI CLK, Din, and CLR pins will all share the same input. The inputs will come from the output of the PIC32. That is, the SCLK, Din, and CLR inputs will come from the PIC32 Pins 25, 18 and 1, respectively. The SCLK inputs are shared because the clocks must stay consistent to function properly between the two. The Din inputs must be the same because the two devices must have the same signal to analyze. The CLR inputs must be shared, because if the design team were to reset the device, it is efficient to only have one reset. The CS/LD pin is the select pin, and selects which device to use. Hence, the two inputs must be independent. Both supply voltages will be supplied by the 15V power source, regulated down to 3.3V. This is the same source used for all of the low power devices. The design team attached a 0.1uF capacitor as a decoupling capacitor to stabilize the Vcc. This completes the design of the DAC process. The output is then sent to the next stage, which is the power op amp.

5.5.3 Voltage Regulation

The voltage requirement for the PIC32 microcontroller and the D/A converters is 3.3V. The initial project idea was to use a LM317 linear regulator to supply the required voltage. This idea was changed for a more efficient switching regulator because the linear regulators require heatsinks. This section will discuss the design of the LM317 linear regulator and the TPS54239 switching regulator.

5.5.3.1 LM317

The PIC32 microcontroller and the D/A converters require 3.3V to operate. A LM317 will be used to provide the 3.3V and 1.5A to the main circuit board containing the microcontroller and the converters. The LM317 is a linear regulator made by Texas Instruments. This linear regulator has an adjustable output voltage from 1.25V to 37V with a guaranteed output current of 1.5A. The input voltage to the LM317 is from another LM317 in the power supply board. The power supply LM317 will provide 15V and 1.5A to the main circuit board LM317. The desired output of the main circuit board LM317 is 3.3V and 1.5A. The linear regulator has a maximum input-output differential voltage of 40V. The input-output differential voltage according to the datasheet for a 1.5A output current must be less than or equal to 15V. The LM317 will have an input-output differential voltage of 11.7V and meets the requirements. The basic components and layout for the LM317 is provided from the Texas Instrument datasheet for the linear regulator and is shown in Figure 5.5.3.1-1. The resistors R3 and R5 determine the output voltage for the LM317. From the datasheet the resistor R3 has a recommended value of 240Ω. The resistor R5 is adjustable and is used to calculate the output voltage for the linear regulator in Equation E5.5.3.1-1.

$$V_o = V_{ref}\left(1 + \frac{R_5}{R_3}\right) + I_{adj}R_5 \qquad \text{(E5.5.3.1-1)}$$

The reference voltage is $V_{ref} = 1.25V$ for the LM317 according to the datasheet. The adjustable current I_{adj} is considered negligible because it typically has a value of 50µA. The equation to calculate the output voltage then becomes Equation E5.5.3.1-2.

$$V_o = V_{ref}\left(1 + \frac{R_5}{R_3}\right) \qquad \text{(E5.5.3.1-2)}$$

To provide enough power for the electrical components on the main circuit board the LM317 will need to supply an output voltage of 3.3V. The adjustable resistor R5 will be calculated form Equation 5.5.3.1-2 to have the output voltage equal to 3.3V. Setting the values of $V_{ref} = 1.25V, V_o = 3.3V,$ and $R_3 = 240\Omega$ in Equation 5.5.3.1-2 the value of R5 is 390Ω. A capacitor C15 is placed before the LM317 with a value of 100nF to smooth the input voltage. Improving the transient response of the linear regulator a capacitor C16 of value 47µF is placed on the output voltage. The LED in Figure 5.5.3.1 will be used to determine if there is output voltage going to the main circuit board. A resistor of 470Ω is used in series with the LED. Figure 5.5.3.1 is the finished circuit diagram with all the component values for a LM317 with an output voltage of 3.3V and output current of 1.5A.

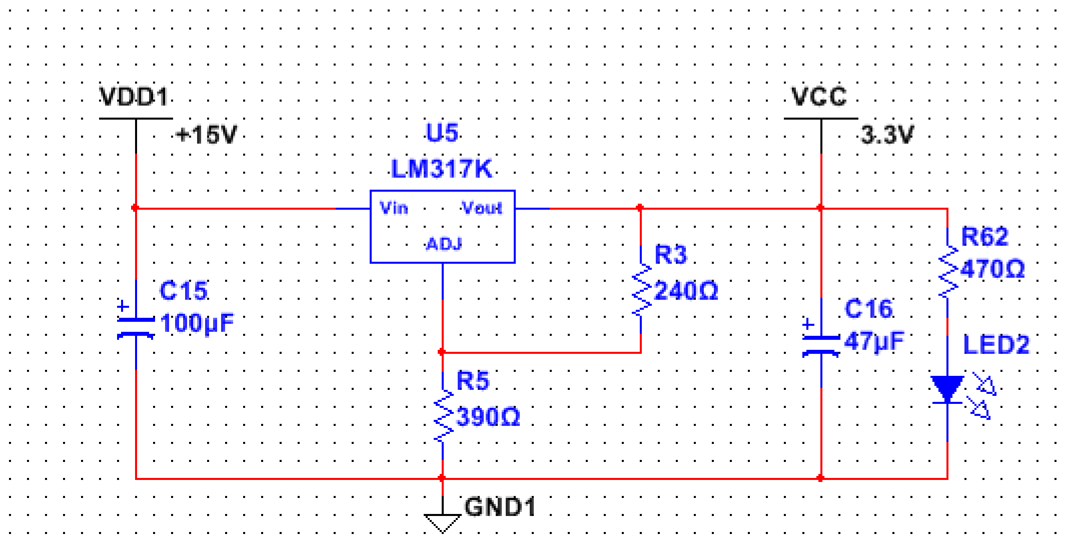


Figure 5.5.3.1: Linear Voltage Regulator LM317

5.5.3.2 TPS54239

The PIC32 microcontroller and the D/A converters require 3.3V to operate. Instead of using a linear regulator a TPS54239 switching regulator will be used because it does not require a heatsink. The TPS54239 switching regulator is made by Texas Instruments and will be designed to provide an output voltage of 3.3V and an output current of 1.5A to the PIC32 microcontroller and the D/A converters. According to the TPS54239 datasheet the switching regulator has an input voltage range from 4.5V to 23V. The input voltage for the TPS54239 is 15V from a TPS54332 switching regulator on the power supply circuit board and is within the specifications for the TPS54239. The output voltage range for this switching regulator is 0.76V to 7.0V and the required 3.3V is well within this range. Inside the switching regulator there is an integrated 140mΩ High-Side MOSFET capable of handling an output current of 1.5A. The switching frequency of the TPS54239 is 600kHz.

The Texas Instruments online design tool WEBENCH was used to create the circuit for the TPS54239 and is shown in Figure 5.5.3.2-1. The designed circuit will provide and output voltage of 3.3V and an output current of 1.5A. To set the output voltage for the switching regulator the resistors R_{fbt} and R_{fbb} are used. The TPS54239 datasheet recommends a value of 73.2kΩ for R_{fbt} but the WEBENCH design tool has this resistor with a value of 84.5kΩ. For the project the

WEBENCH resistor value will be used and R_{fbb} will be calculated in order to have the required output voltage of 3.3V. To calculate the value of resistor R_{fbb}, Equation E5.5.3.2-1 will be used.

$$V_o = V_{ref}\left(\frac{R_{fbt}}{R_{fbb}} + 1\right) \qquad \text{(E5.5.3.2-1)}$$

The recommended reference voltage is $V_{ref} = 0.765V$ from the datasheet. The project requires an output voltage of 3.3V for the PIC32 microcontrollers and the D/A converters. Evaluating Equation E5.5.3.2-1 with the values of $V_{ref} = 0.765V$, $V_o = 3.3V$, and $R_{fbt} = 84.5k\Omega$, the resistor R_{fbb} is 25.5kΩ. The topology for this switching regulator is a buck converter. The TPS54239 is 89.1% efficient and has a total power dissipation of 0.6W. Figure 5.5.3.2 is the complete circuit diagram for the TPS54239 switching regulator to have an output voltage of 3.3V and an output current of 1.5A.

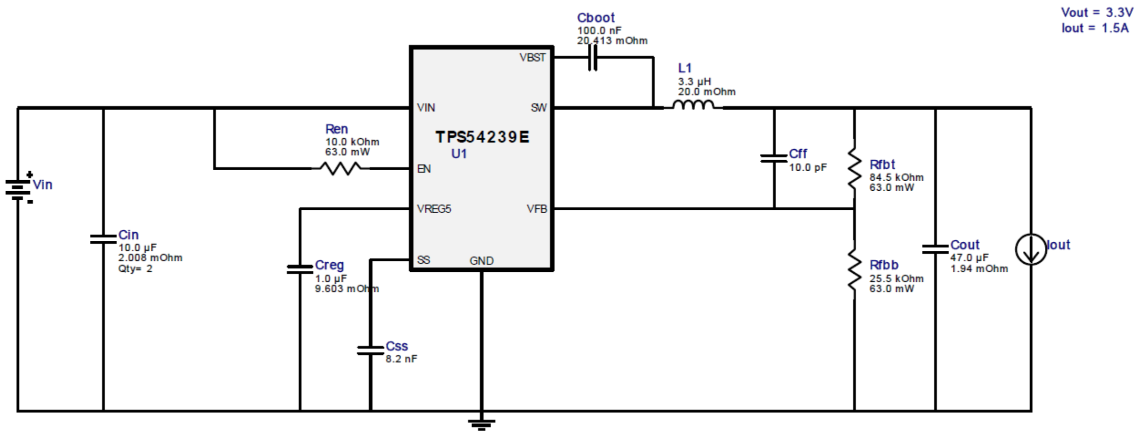


Figure 5.5.3.2: Switching Regulator with V_{out} = 3.3V and I_{out} = 1.5A

5.6 Current Driver

Once the audio signal has been converted to the frequency domain in the processor, and consequently converted to an analog signal from the LTC1665, the analog voltages must now be sent to a voltage amplifier to power the water pumps. This amplification can be done by several devices, and the design team had to establish which type of amplifier to use. The team eventually narrowed it down to two choices: using a MOSFET or an Operation amplifier. In order to decide which component to utilize, the design team researched and analyzed both options and then as a group decided on which to use.

5.6.1 PWM vs. Linear Amplifier

Typical high current DC motors are driven by a pulse width modulation (PWM) scheme for power efficiency. However, the DC motor chosen for the project includes built in circuitry that requires a stable input DC voltage. The PWM output is a pulse voltage with high ripple. Thus PWM is not applicable in this design, and a linear amplifier is required.

5.6.2 Current Amplifier using MOSFET

One option for a power amplifier involves using a MOSFET transistor as an amplifying device. A MOSFET, or metal -oxide-semiconductor field effect transistor is a three terminal device of which are the source, gate, and drain. A MOSFET has an advantage in that it requires very little current to turn on, and can amplify to a very high current. It is the most common transistor in present day circuitry, which will facilitate the troubleshooting of the device. The design team narrowed the selection of the MOSFET down to one device, the Fairchild Semiconductor FQP30N06L N-Channel MOSFET. The FQP30N06L features proprietary DMOS technology, and with it, boasts minimal on state resistance and superior switching performance. Table 5.6.2 shows all of the relevant specifications of this device.

FQP30N06L	
Drain Current(output current)	32A
Gate Source Voltage (supply range)	-20 to 20V
Drain Source Voltage	Up to 60V
Turn on Delay Time	15 ns
Turn on Rise Time	210 ns
Package	DIP
Pins	3

Table 5.6.2: Relevant Characteristics of MOSFET

This device can drive a high voltage, which is needed for the water pumps themselves. In addition, there is a very miniscule delay and rise times, ensuring that the water pumps will respond in adequate time. Furthermore, this component comes in a DIP package, which is required for the design. All in all, this MOSFET can be an effective component for the power drivers.

5.6.3 Current Amplifier using Power Op Amp

Another option that the design team considered when deciding on an amplifier is a Power Operational Amplifier. For this particular application, the design team narrowed the model of the component to one choice. This was the Texas Instruments OPA548 High Voltage, High Current Operational Amplifier. This amplifier is ideal for driving a wide variety of loads, and has the additional ability to operate on a single or dual power supply. This single supply quality is beneficial in that the design team will not have to supply an additional power supply, which would take up extra space, along with efficiency issues. In addition, the OPA548T comes with current limiting capabilities, to prevent overloading the circuit. This safety feature is an obvious benefit to the design, and will ensure the safety of the IC. Table 5.6.3 shows all of the relevant characteristics of this component.

OPA548	
Supply Voltage (single supply)	8-60V
Output Current (Continuous)	User defined 0-5A
Input Voltage	-.5 to 5V
Slew Rate	10V
Pins	7
Package Type	DIP

Table 5.6.3: OPA548 Relevant Characteristics

This device can operate with a supply voltage coming from the power supply at 15V. In addition, this device can operate with the input voltage of 3.3V coming from the DAC in the previous stage. Furthermore, this op amp comes in a DIP package, a requirement for all of the circuit components of the design. Figure 5.6.3 shows a schematic of the Op amp itself.

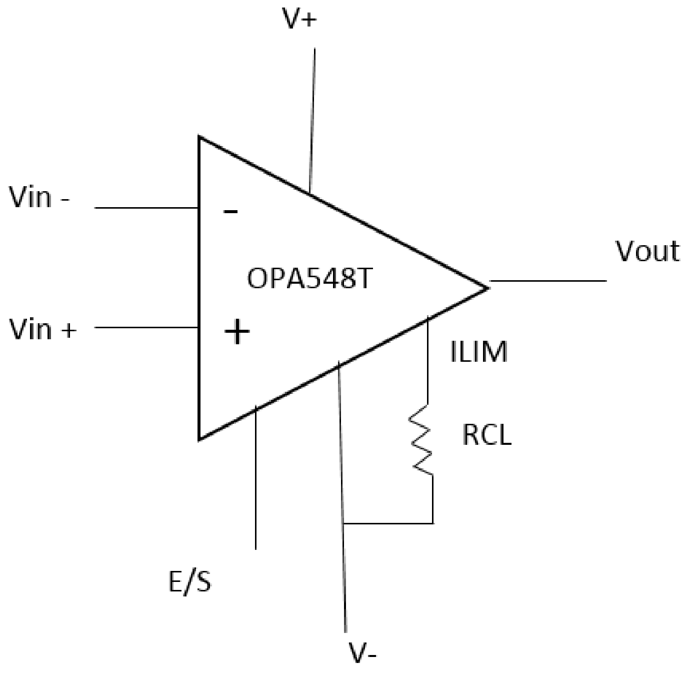


Figure 5.6.3: Schematic for Power Op Amp

V_{in-} is the non-inverting input, and V_{in+} is the inverting input. R_{CL} is the current limiting resistor which goes to the I_{LIM} pin which is the current limit set. V^+ and V^- are the power supplies, and the power op amp can operate on both or one of those. V_O is the output voltage of the amplifier and finally E/S is the enable/disable control input or the thermal shutdown status output. In all, there are 5 possible input pins one output, and one input/output pin.

As said before, the OPA548 features a user selectable current limit. This can be set from 0 to 5 A. Unlike other power op amps, which use a power resistor on the output of the op amp, the OPA548 detects the load indirectly. By setting R_{CL} to the correct value, the current can be limited to any value between 0 to 5A. This calculation is shown in Equation E5.6.3-1 below where ILIM is the desired current limit.

$$R_{CL} = \frac{(15000)(4.75)}{ILIM} - 13750\Omega \qquad \text{(E5.6.3-1)}$$

These qualities make the OPA548T an attractive component for an amplifier.

5.6.4 Comparison and Decision

Upon much consideration, the design team opted to use the Power Op amp for the final design. The design team determined that a power op amp was easier to implement, and the circuitry would be less complex to design for linear applications. Thus, due to simplicity purposes, the power op amp was chosen, and the design team utilized the OPA548s.

5.6.5 Operational Amplifier Design Schematic

Each power driver would power one of the water pumps. Thus, in the final design there will be 16 individual drivers, all identical. Figure 5.6.5 shows the design schematic for the power drivers.

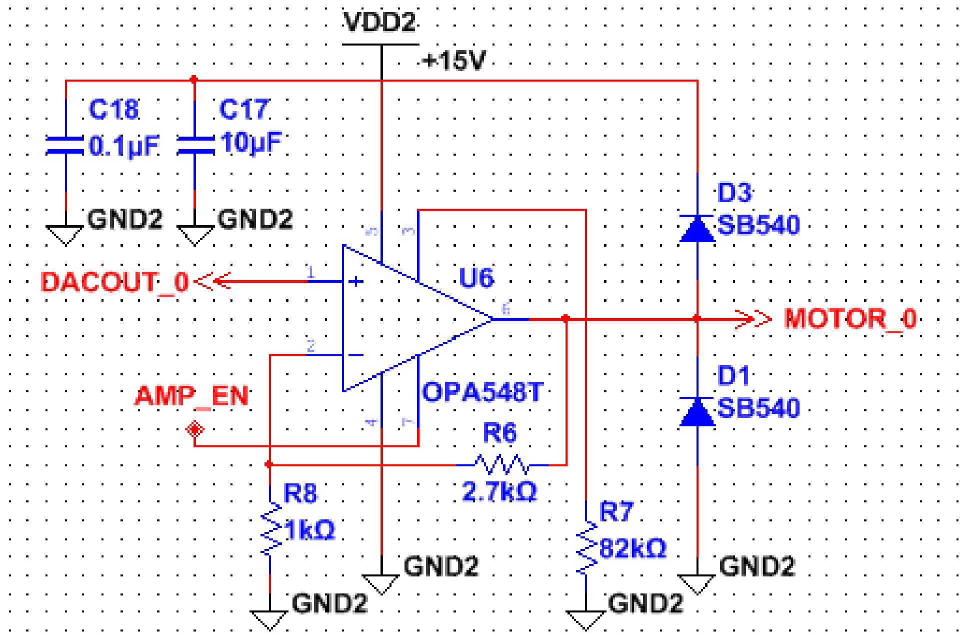


Figure 5.6.5: High Current Power Driver for Motors

Since there will be a positive gain for the drivers, the operational amplifier configuration will be of the typical non-inverting type. Thus, the input to the positive terminal will be the output of the DAC corresponding to the proper water pump. This will vary from 0-3.3V. The inverting terminal will be sent to the ground, with a resistor to achieve the desired gain. The supply pins will be attached to the 15V power supply, and will serve as the V_{CC} of the op amp. The required output voltage will need to be 12V at the maximum input (3.3V) because that is the operating max voltage of the water pump. Thus the gain is calculated in equations E5.6.5-1 through E5.6.5-5, which are the equations for general gain and the gain of a non-inverting op amp.

$$Gain = \frac{V_{out}}{V_{in}} \tag{E5.6.5-1}$$

If V_{in} = 3.3V and V_{out} = 12V,

$$Gain = \frac{12}{3.3} = 3.63 \tag{E5.6.5-2}$$

The gain of the power driver in Figure 5.6.5 can also be expressed as:

$$Gain = (1 + \frac{R_6}{R_8}) \tag{E5.6.5-3}$$

Setting E5.6.5-2 and E5.6.5-3 equal to each other,

$$3.636 = \left(1 + \frac{R_6}{R_8}\right) \tag{E5.6.5-4}$$

Solving for the resistance ratio yields

$$\frac{R_6}{R_8} = 2.636 \tag{E5.6.5-5}$$

Equation E5.6.5-5 can be satisfied by setting R_6 = 2.7kΩ and R_8 = 12.7kΩ. Thus, the resistor values were chosen for the circuit. In addition, to limit the current in the op amp to the pumps, using equation E5.6.5-6 supplied by the datasheet, the current limiting resistor R_7 was chosen. The design team chose to limit the current to 0.75A as that was the maximum current the motors could handle.

$$RCL = \frac{(15000)(4.75)}{I_{LIM}} - 13750\Omega \tag{E5.6.5-6}$$

Setting I_{LIM} to 0.75A,

$$RCL = \frac{(15000)(4.75)}{0.75} - 13750\Omega = 81250\,\Omega \tag{E5.6.5-7}$$

Thus the current limiting resistor was set to 82kΩ, as it is a close approximation to the calculated resistance value, which is not available to buy. The two capacitors, 0.1uF and 10uF, are to create noise immunity from the power supply for the circuit. Since the power amplifier switches a large amount of current, there are large transients along the power line, thus requiring a larger capacitor to minimize noise. Thus the 10uF capacitor was included in the design. The 2 diodes are to prevent back electromotive force, as described in the next section 5.6.5.1.

5.6.5.1 Back Electromotive Force

Electromotive force, or EMF, is the voltage generated by a spinning motor, such as the DC motor contained in the water pumps. All motors involve electromagnetic devices, and as such will follow the rules of electromagnetism. Thus, the motor itself will store energy when powered and when the supply is switched off, will generate its own force. Since it is generating its own voltage, the motor itself becomes a power source and will induce a current towards the original current driver, potentially causing a voltage spike. This is called back electromotive force, otherwise known as counter EMF or back EMF. It is a voltage that pushes back against a current that is driving it. If this back EMF is not suppressed or controlled, it can generate problems such as creating interference and can even damage the internal electronics. Thus, it is imperative that the design for the drivers suppresses this effect. Hence, the design team implemented two Shottky diodes as barrier rectifiers. These rectifiers will prevent current from going back into the op amp from the output of the circuit. Implementing these components effectively suppresses any effects that back EMF will have on the circuit, and will ensure the safety from damage of the design.

5.6.6 Connector

To simplify the physical design, the design team used connectors to bunch together all 16 driver outputs, so that they could be transferred over a single port. This port can be connected to a bundle of wires that can then be connected to the water pumps. This and allows the isolation of the PCB from the water pumps, in addition with simplifying the output to one area. Figure 5.6.6 shows the schematic for the connector.

5.7 PCB Design Schematic

In order to bring the digital schematics to a physical form, a custom printed circuit board must be designed, and then sent to a manufacturer to print out. Thus, the design team integrated all of the circuit components and designed the circuit board from scratch. Using Ultiboard software, the PCB was designed and

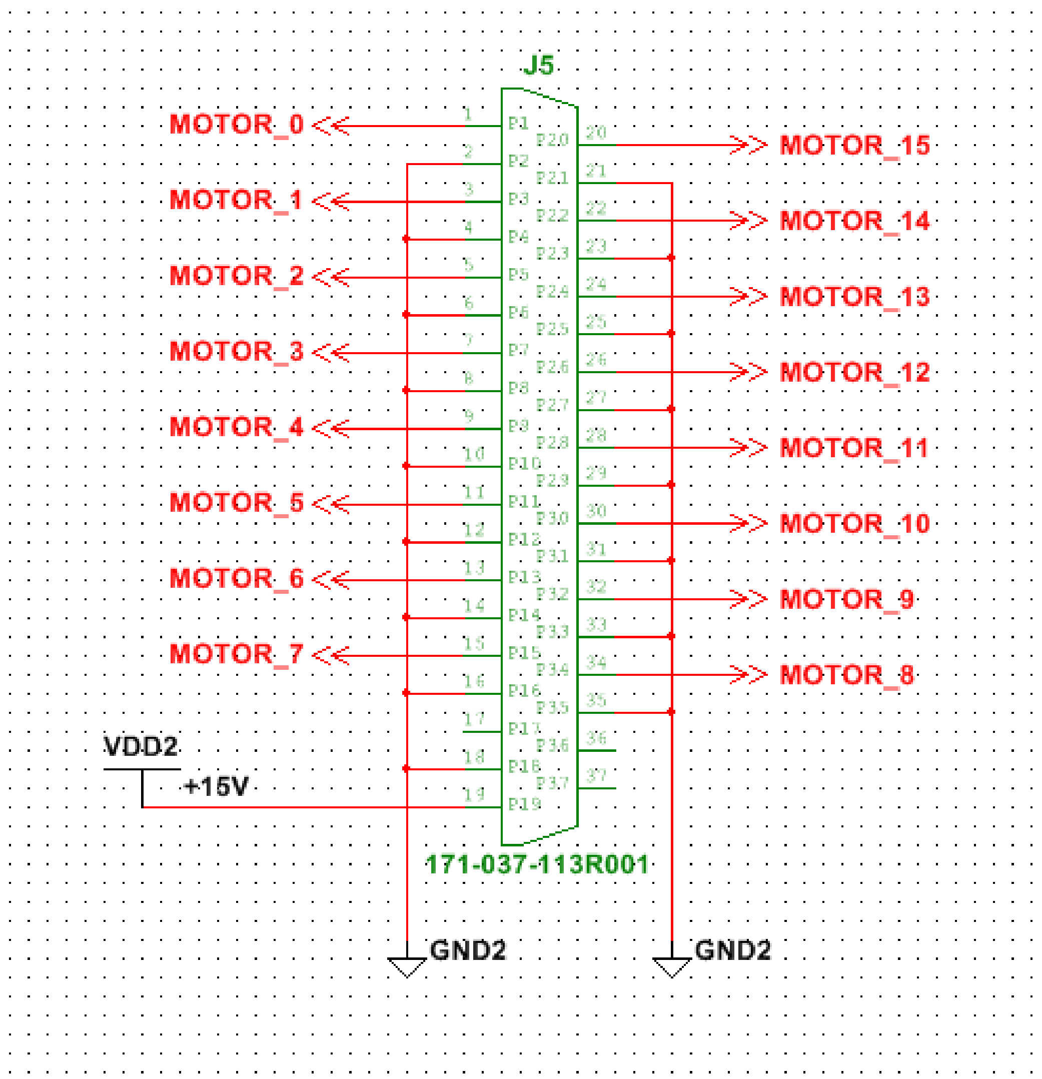


Figure 5.6.6: Connector from Drivers to Motors

checked for errors. Once this was done, the design team sent the design layout to a manufacturer for review, in order to determine whether the manufacturing corporation is capable of producing such a board.

The design of the board itself was straightforward. All of the connectors were placed at the edge of the board for easy access. The design team decided to use two PCBs in all, one for low power components, and one for the high power components. The low power board was biased at 3.3V and contained the processor and its neighboring components. The high power board was exclusively used for the power op amps, which required to be biased at 15V. The design team used two boards to facilitate testing. For every component that contained a decoupling capacitor, the capacitors were placed as close to their corresponding components as possible, so that they could function properly. The linear regulator for the 3.3V power supply was placed so that it was far apart from the other components, as the regulator would dissipate the most heat. Heat sinks on the op amps were required for the high power board due to high heat dissipation. The reset switch was placed on the bottom left of the board, for ease of access. In addition, the design group segregated the power supply traces, so that the high current traces were kept separate from the low current traces. The same segregations were also made for the ground lines, so that there were separate sets of ground traces for high current and low current. This was done so that the noise of the high power traces did not affect the noise of the low power ones. Although not shown, the design team picked to have two ground planes, on top and one on bottom instead of ground traces. This was done to reduce impedance of the ground signals. The power traces were made to be as large as possible, as there were not enough layers for a power plane. The design team decided to use through-hole technology for the PCB. Figure 5.7-1 shows the layout of the low power PCB and Figure 5.7-2 shows the layouts of the high power PCB. Figure 5.7-3 shows a more visual 3D model of the prototype PCBs.

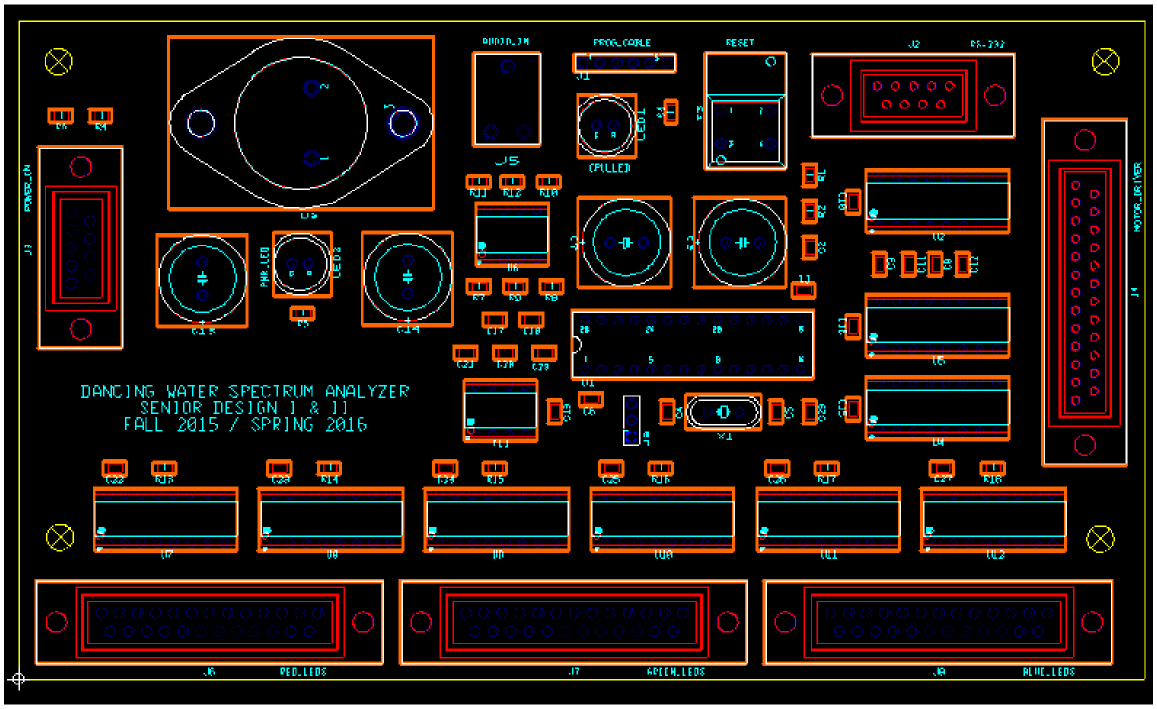


Figure 5.6.6-1: Low Power PCB

Figure 5.7-4 shows the design rule for the auto router. Due to the complexity of the boards and the large amount of traces, the design team utilized an auto routing software to rout all of the traces. Thus, no manual routing was done. The auto router used a 10 mil trace width, a standard width for typical boards. In addition, clearance between traces as well as clearance between traces to vias were set to 10 mil as well. These design rules are easy for manufacturers to fabricate PCBs. Although smaller clearances and trace widths can be done, the board would have cost much more, and the yield would not have been as good.

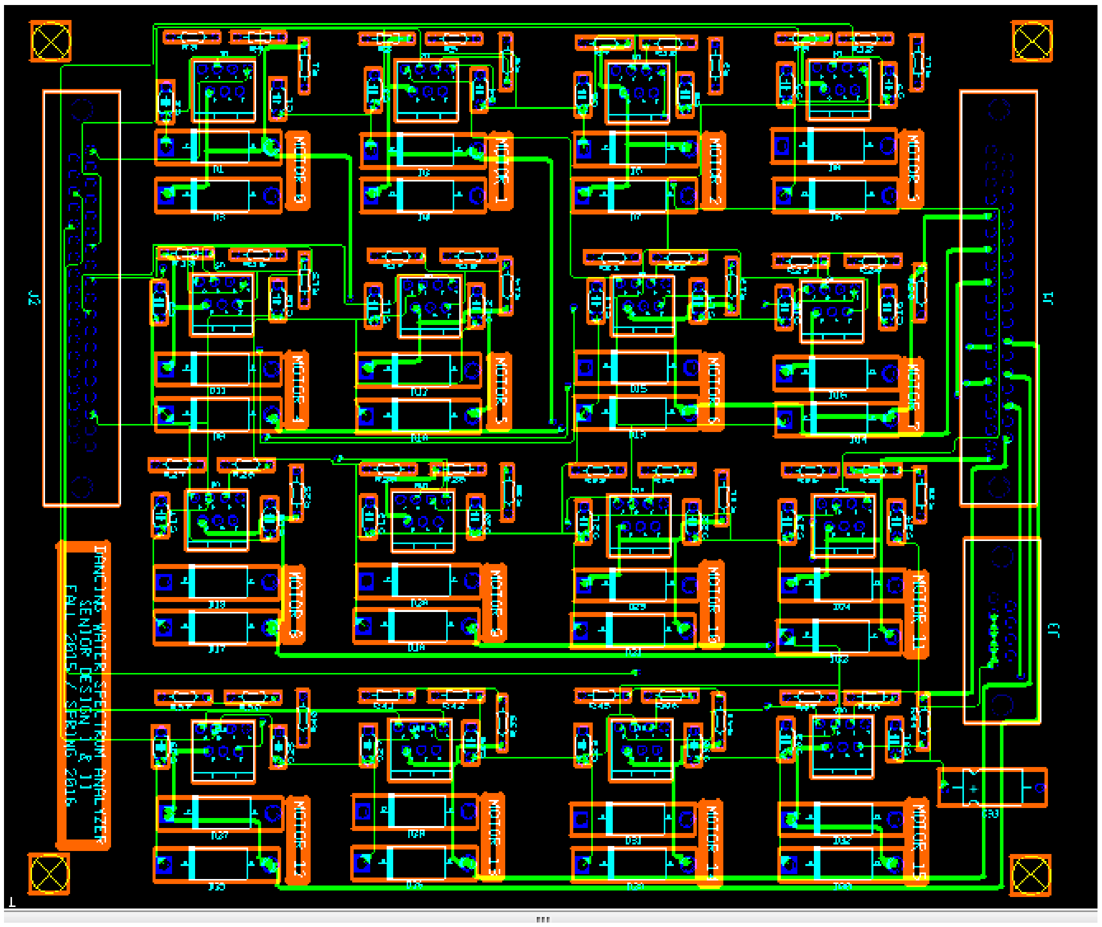


Figure 5.6.6-2: High Power PCB

Similar to trace width and clearances, the through-hole and via sizes had to be specified as well. Figure 5.7-5 shows the design rules for the through-holes and vias. It was decided that default design rules from the auto routing software would suffice. For the through hole pad annular ring, it was determined that 7 mil from the drill hole would be safe and easily manufactured. Via drill diameters and pad sizes were set to 23.6 and 39.37 mils, respectively. There were no micro-vias on this board, so the rest of the design rules were irrelevant.

Figure 5.7-6 shows the summary statistics for the PCBs after the auto router completed its work. At 657 pins, 93 vias, and 468 connections, these boards were relatively complicated.

Most of the traces on the boards were set to 10 mil traces and 10 mil clearance. However, for power traces, these were set to be much bigger, to reduce trace impedance. They were set to 100 mils for VDD1 (15V) and 130 mils for VDD2. These were the biggest trace widths the auto router was able to route without manual intervention after many trials and errors. For the traces from the high current power amps to the water pump connector, they were set to 30 mils.

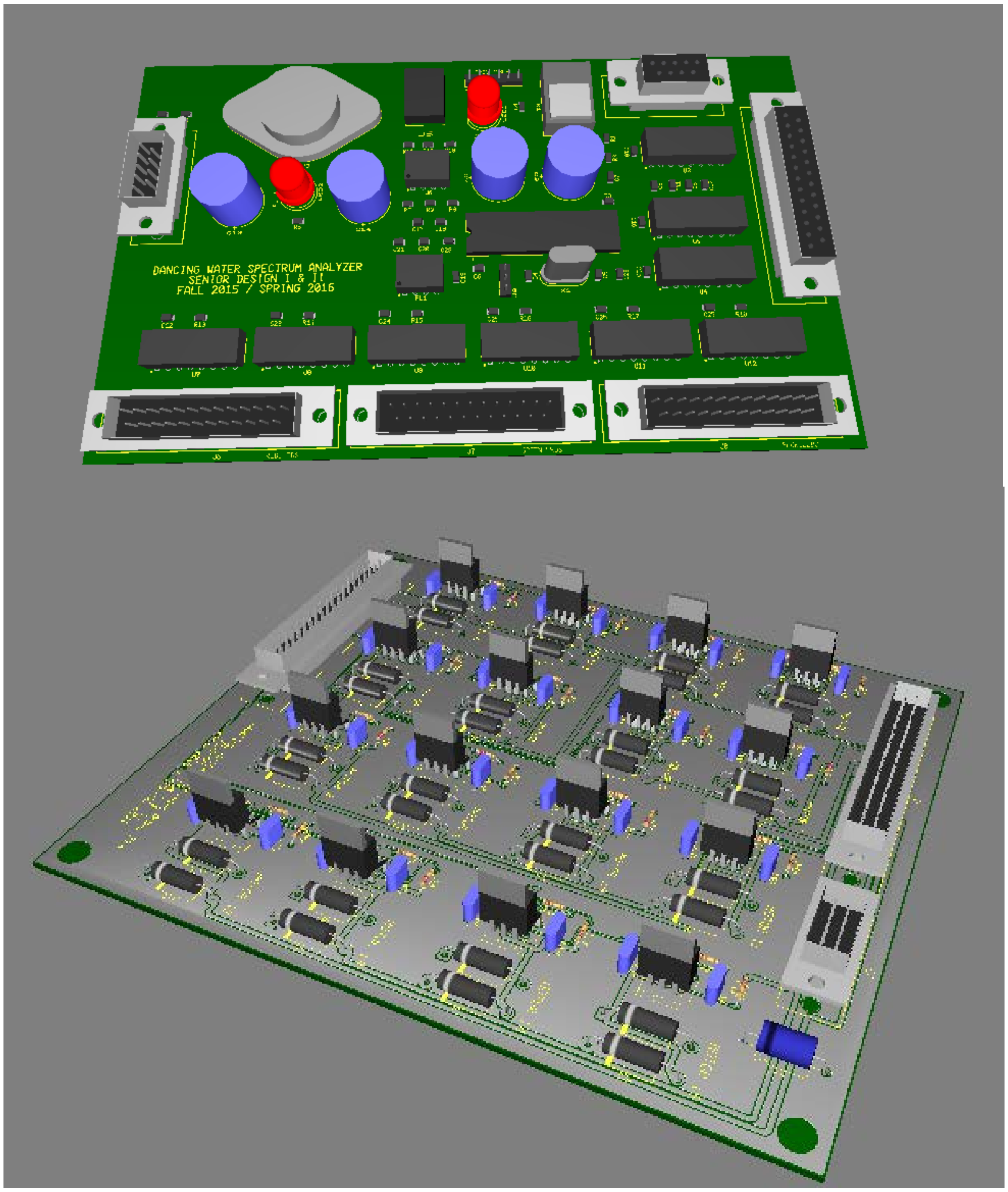
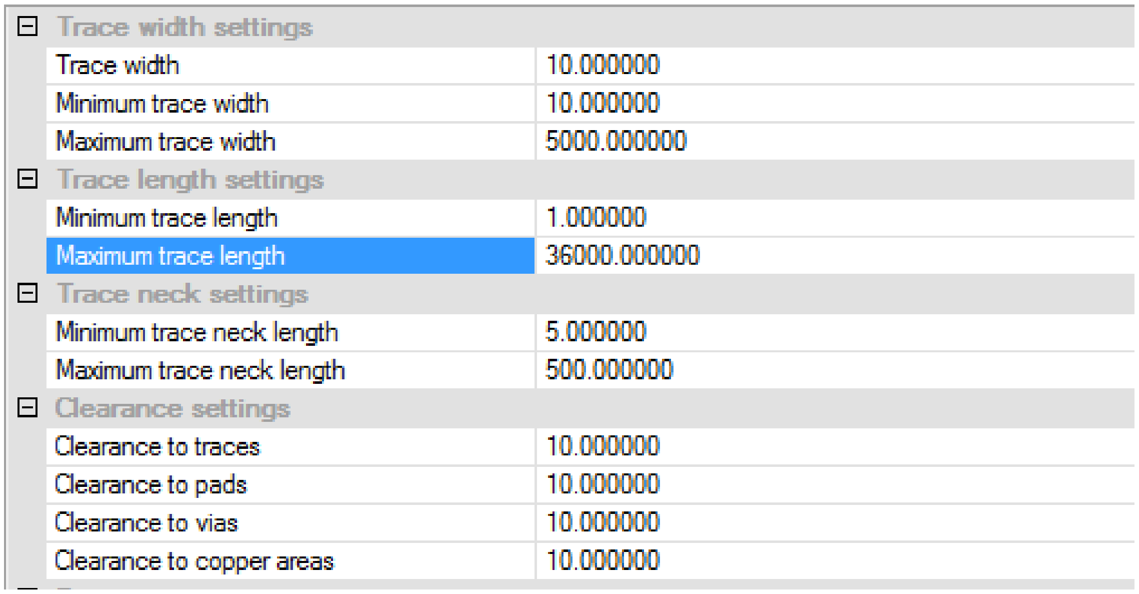


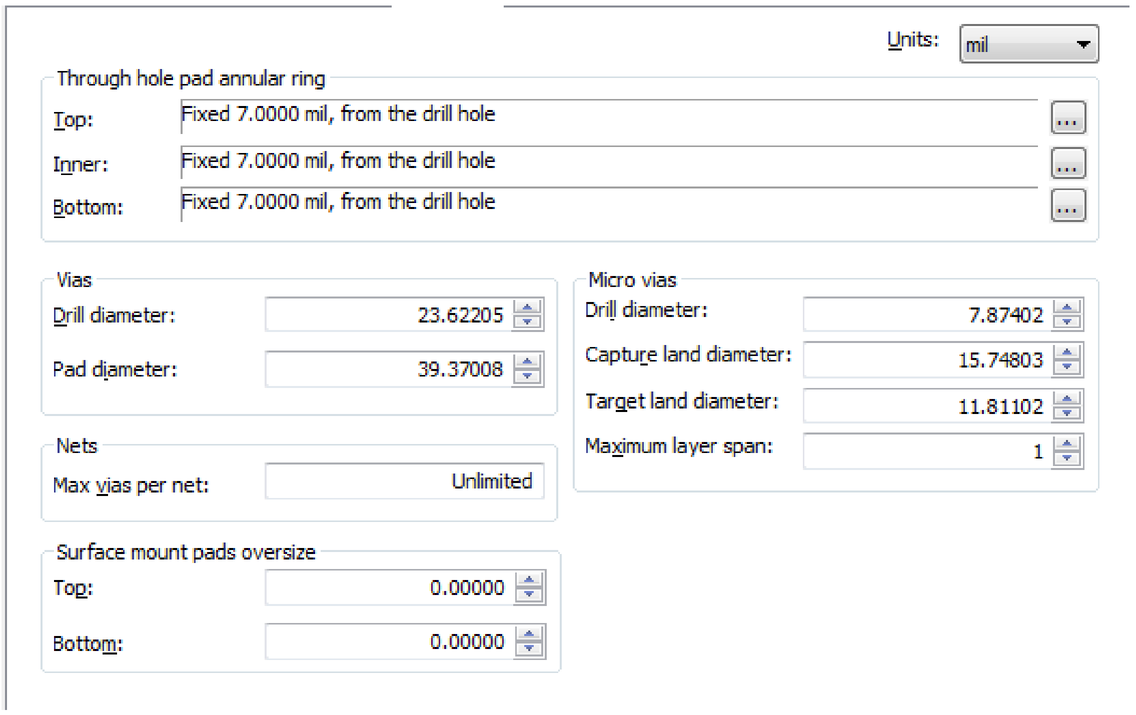
Figure 5.7-3: 3D Model of PCBs (Low power on top, high power on bottom)

Figure 5.7-7 shows the PCB board stack up. It is a two layer board with a copper top and copper bottom. Above the copper layers are the solder masks. These are to protect all the traces on the board when components are being soldered to the boards. Above the solder masks are the silk screen layers. This makes it easy for the user to assemble and debug the board. It lists all of the components and where they are located on the board.



Trace width settings	
Trace width	10.000000
Minimum trace width	10.000000
Maximum trace width	5000.000000
Trace length settings	
Minimum trace length	1.000000
Maximum trace length	36000.000000
Trace neck settings	
Minimum trace neck length	5.000000
Maximum trace neck length	500.000000
Clearance settings	
Clearance to traces	10.000000
Clearance to pads	10.000000
Clearance to vias	10.000000
Clearance to copper areas	10.000000

Figure 5.7-3: PCB Design Rule for Traces and Clearance



Units: mil

Through hole pad annular ring

Top:	Fixed 7.0000 mil, from the drill hole
Inner:	Fixed 7.0000 mil, from the drill hole
Bottom:	Fixed 7.0000 mil, from the drill hole

Vias

Drill diameter:	23.62205
Pad diameter:	39.37008

Micro vias

Drill diameter:	7.87402
Capture land diameter:	15.74803
Target land diameter:	11.81102
Maximum layer span:	1

Nets

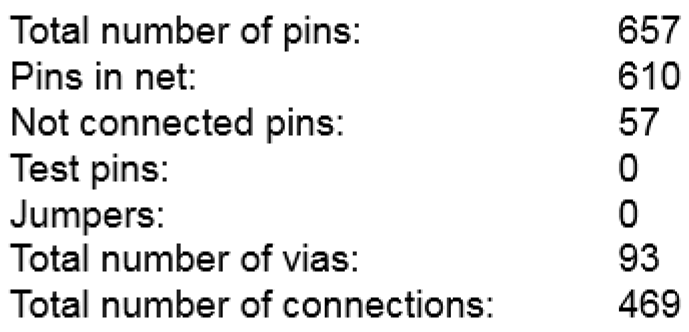
Max vias per net:	Unlimited

Surface mount pads oversize

Top:	0.00000
Bottom:	0.00000

Figure 5.6.6-4: PCB Design Rule for Through-Holes and Vias

A set of GBR type files were sent to a PCB manufacturer, along with its specs for confirmation. Table 5.7 shows the final job info specifications as confirmed by the manufacturer.



Total number of pins:	657
Pins in net:	610
Not connected pins:	57
Test pins:	0
Jumpers:	0
Total number of vias:	93
Total number of connections:	469

Figure 5.6.6: PCB Board Statistics

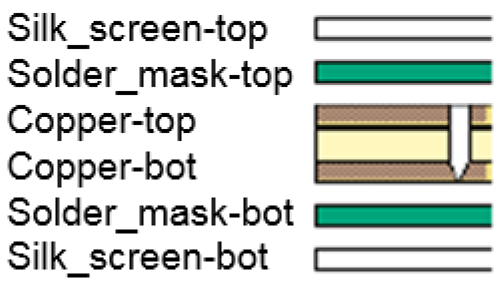


Figure 5.7-7: PCB Board Stack-up

Part Size (X,Y) inches	10.5 X 6.5
Thickness	62 mil
I/L Weight	0 Oz
O/L weight	1 Oz
Copper Layers	2
Drill Layer	1
Route Length	34 Inch
Soldermask Side	Both
Soldermask Color	Green
Soldermask Type	Photo image
Silkscreen Side	Both
Silkscreen Color	White
Impedance Tolerance	0%
Board Bow and Twist Percentage	0%
Gold Thickness	0 mil
Material	FR4
Finish Type	HAL

Table 5.7: Job Info

Thus the board size is relatively large at 10.5 by 6.5 inches. The board is of standard thickness at 62 mil. The board uses 1 ounce copper on all of the copper layers. The design team picked green for the board color, and white for the silkscreen color, for easy visualization. The board material was determined to be FR4. FR4 is a grade designation for PCBs, made of flame resistant fiberglass. This material was picked due to its inexpensiveness and ease of manufacturing.

5.7.1 Selection of Manufacturer

The design team selected Sunstone Circuits as the PCB manufacturer. This was mainly due to its cheapest cost of manufacturing, as the company has special prices for low volume, non-expedited PCB boards. For the design team's PCB board, the total cost was quoted from Sunstone circuits at $229.75. This was much cheaper than the standard price which was around $500. Sunstone also provides a free design for manufacturing analysis based on the GBR files submitted. Thus, the design team took advantage of this free analysis.

5.8 LEDs

Because a source of light is needed for the display, the simplest and least power-consuming solution is to go with the light emitting diodes, more commonly known as LEDs, which will be programmed so that they flash to the beat of the music. The market for LEDs can vary in cost, quality, brand, and design. They are composed of PN junctions that transmit light when put in forward-biasing mode. The team can find these in sizes of 3mm, 5mm, and 10mm. The desired size is 5mm and would typically require a voltage of approximately 2 volts. The goal for this aspect of the project is to have 16 LEDs to match the 16 water pumps that create a multicolor display from the roof of the casing.

5.8.1 Schematics

The schematic for the LED drivers are shown below. Rather than using discrete, individual drivers, the team chose to use a highly integrated LED driver IC, the STP08CP05 from ST Microelectronics. These IC's are available in DIP package and can operate down to 3V. This is critical since the team did not want to work with SMT packages, and also wanted parts that work at 3.3V. They used 2 of these IC's, STP08CP05, in a cascaded configuration to drive the 16 LED's in the system. Thus the Serial Out pin of the first device is connected to the Serial In of the second device. To simplify the system interconnects, they decided to bring all LED driver output wires to a connector. Thus they can easily connect/disconnect the processor PWB from the LED's. It also makes it easier for debugging/troubleshooting since the connector pins can be probed. C56 and C55 provide decoupling capacitance for the LED drivers. Figure 5.8.1 shows the schematic for the LED shift register.

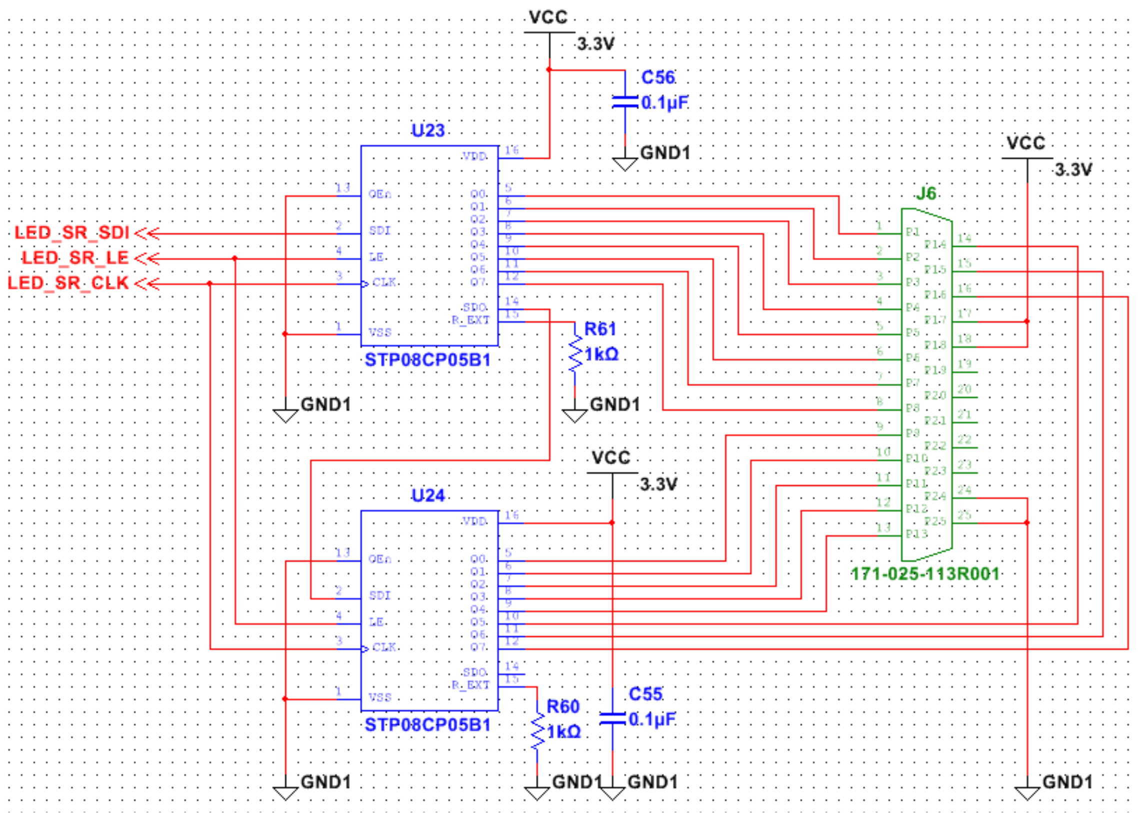


Figure 5.8.1: LED Drivers

5.8.2 LED Drivers

U23 and U24 are the LED drivers, providing a constant current source for the LED's. The STP08CP05 device is a monolithic, low voltage, low current, power 8-bit shift register designed for driving LED panels. This highly integrated device includes the serial-to-parallel shift registers, the high current MOSFET drivers, as well as the thermal protection circuitry. The STP08CP05 is low voltage and can be operated all the way down to 3V. In the application, it is used at 3.3V. Besides being a low voltage device, the STP08CP05 draws very little current. The supply current spec for the device shows only 13.5mA when fully on. At 3.3V supply voltage, this part is using only 44.5mW. Figure 5.8.2-1 is the internal block diagram of the STP08CP05 device.

The STP08CP05 device contains an 8-bit serial-in, parallel-out shift register that feeds an 8-bit D-type storage register. The serial shift register inputs are Serial Data In and Serial Data Clock. The group connects the Serial Data In and Serial Data Clock pins directly to the PIC32 microprocessor general purpose IO pin. Although the Serial Data Clock pin can be driven at up to 30 MHz, they will drive this pin at a much lower 1 MHz frequency. The human eye is not sensitive to this high level rate of change so a much slower clock is adequate for the application.

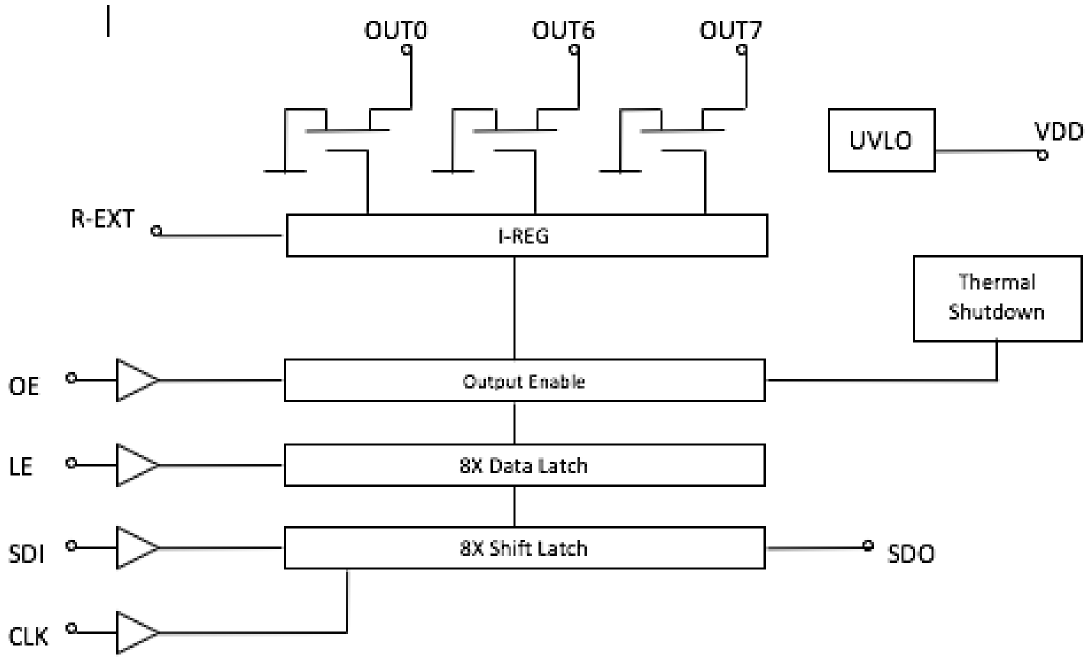


Figure 5.8.2: Block Diagram of Shift Register

After the data has been shifted into the 8-bit shift register, it will activate the Latch Enable pin on the STP08CP05 device. As with the Serial Data In and Serial Data Clock pins, the group connects the Latch Enable pin directly to the PIC32 microprocessor general purpose IO pin. Software will activate the Latch Enable pin once all the serial data has been loaded into the 8-bit shift register. Although the STP08CP05 device has an Output Enable pin to turn on or turn off all the outputs simultaneously, the team does not need this functionality, and thus have connected the Output Enable pin directly to GND, i.e., the device outputs are always enabled. They used the serial shift register data to individually turn on or turn off each individual LED. Since the project needed 16 LED drivers, they used two of the STP08CP05 devices, connecting the Serial Data Out of the first device into the Serial Data In of the second device. The Serial Data Clock are common to both device and thus, they effectively created a 16-bit logical shift registers out of the two STP08CP05 devices.

In the output stage, eight regulated current sources can provide 5mA to 100mA of constant current to drive the LEDs. The adjustable output current is programmed by setting an external resistor to a desired value. In this application, this resistor is set to 1kΩ. R60 and R61 are the current programming resistors in the schematic. The 1kΩ resistor value programs the STP08CP05 device to output a 20mA constant current output. This is the full turn on current for the LED's. The STP08CP05 device provides a 20V maximum output driving capability. In the design, the group will be using 15V, and thus this will allow us to drive multiple LED's in series to increase the overall brightness of the LEDs.

Beside the standard LED driver functionality, the STP08CP05 device also incorporates other useful functionality. The STP08CP05 device has built in Electro-static Discharge (ESD) protection of 2.5 KV to protect the device. STP08CP05 device also has a built in thermal shutdown protection. The STP08CP05 device will automatically shut down all outputs if the internal temperature of the device reaches 170 degrees Celsius.

Since the group is using a DIP package, this device can dissipate up to 1.4W at 25 degrees ambient temperature. If they rate this to 1W, and since they have 8 outputs, this means they can safely dissipate up to 1W / 8 outputs = 125 mW per output. Since the constant current output is programmed to 20mA, then the maximum voltage drop they can tolerate inside the STP08CP05 device is 125mW / 20mA = 6.25V. The group will design the LED voltage to be less than 6.25V at the STP08CP05 device.

5.8.3 Color Spectrum

The human eye perceives colored light through the primary red, blue, and green. Each of these colors requires 8 bits which have integer values varying from 0 to 255, which can display 256*256*256=16777216 possible colors. The group wants to keep it simple and hope to ultimately obtain the following color spectrum shown on Figure 5.8.3-1 by using single color LEDs.

Figure 5.8.3: Color Spectrum

5.8.4 Power distribution

Powering an LED or multiple LEDs differs from powering typical electronics. The majority of electronics demand a constant voltage source, whereas LEDs require a constant current source. The simplest way to power the LEDs would be to use a DC source with constant voltage that is already supplying energy to other components in the circuit.

A very common and inexpensive way to do this is to place an LED in series with a resistor so that current can be controlled. The only drawbacks to this is that power dropped across the resistor is lost as heat and the current cannot be controlled closely. These downsides are more evident when multiple LEDs are involved, so it might be better for them to be powered in parallel. The only drawback to putting them in parallel, however, is that the light output can vary from one LED to another. If one fails to “open”, the others will receive more current. This can potentially cause the other LEDs to burn out. A more efficient method would be to use a switching mode power supply which has the ability to boost the output voltage. This allows for more LEDs to be placed in series.

Below is the schematic of one of 16 external LED Displays which show how the LEDs are connected to the Processor PWB. There are 16 individual LED boards with each corresponding to a frequency spectrum. Each is identical in design. All channels are connected to the Processor PWB via a standard 25-pin connector.

The LEDs are all driven from the 15V power supply. The group has a series 330 Ohm resistor in-line with the LEDs. This resistor purpose is to reduce the voltage to the LEDs. Since the LEDs are driven by a 20mA constant current source, the resistor will drop 330 Ohm x 20 mA = 6.6V. The resistor will dissipate 330 Ohm x 20 mA x 20mA = 132mW. This is acceptable since the resistors are 250mW components.

The LED forward voltage drop specification is 3.4V. Since the design is using 2 LEDs in series, the total voltage drop across both LEDs is 3.4V x 2 = 6.8V. The available voltage at the LED is 15V – 6.6V = 8.4V, this is higher than the required 6.8V forward voltage drop to turn on the LED.

The voltage at the driver chip is the Power Supply voltage minus the voltage drop across the resistor minus the LED forward voltage drop. Thus, the voltage at the driver chip is 15V – 6.6V – 6.8V = 1.6V. This is much lower than the 6.25V maximum calculated from the above “LED Driver” section. Thus the driver IC will be well under the maximum power dissipation specification. Figure 5.8.4-1 shows a schematic of the LED display.

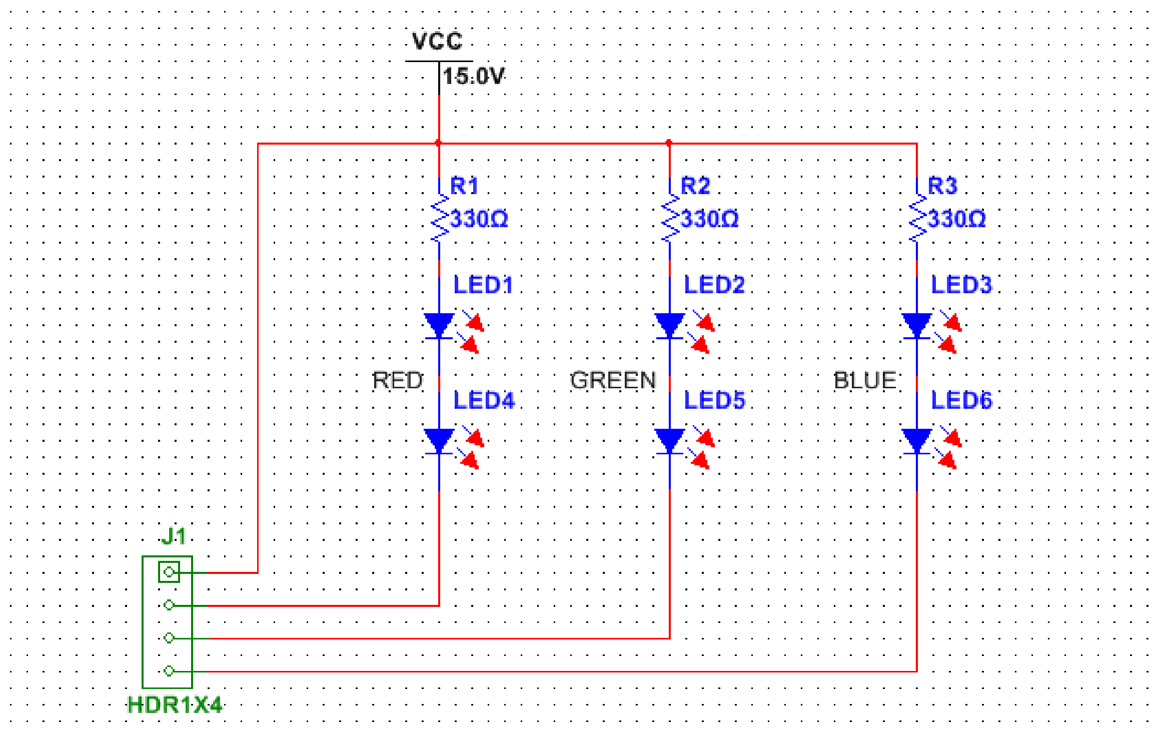


Figure 5.8.4-1: External LED Display

In order to simplify the assembly of the external LED components, the group decided to create a small PWB with 6 LEDs. This allows the LEDs to be easily soldered onto the PWB, and the PWB to be easily installed onto the water display assembly. Below on figures 5.8.4-2 and 5.8.4-3 are the layouts of the mini-LED Display Assembly as well as a 3D model of the small board. These were designed from Ultiboard.

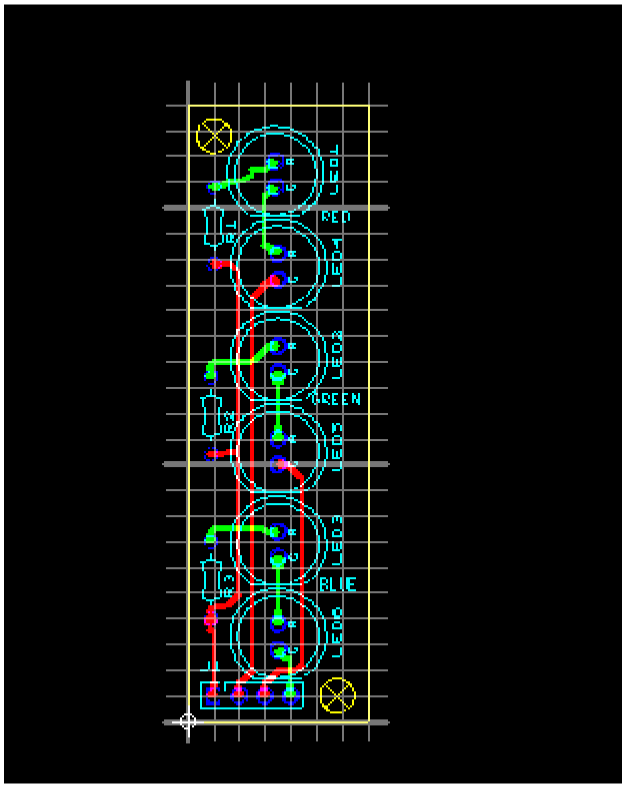


Figure 5.8.4-2: Top Down PCB Model of Individual LED

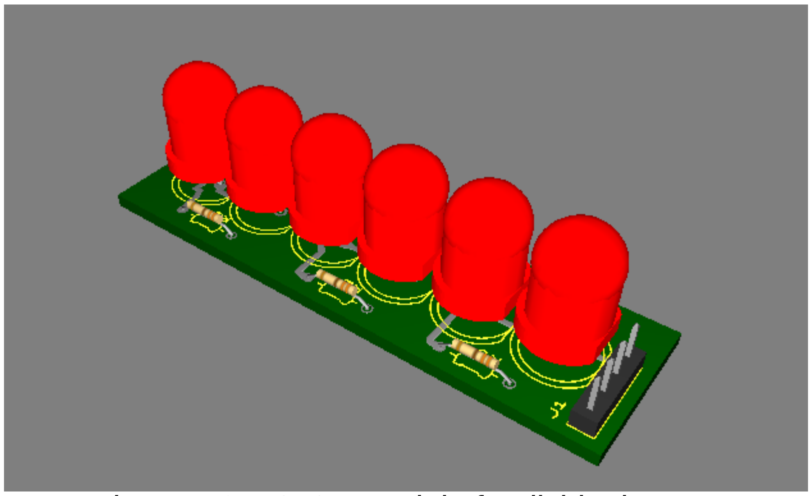


Figure 5.8.4-3: 3D Model of Individual LEDs

5.8.5 Customizability

LEDs can be controlled using logic devices. In pursuance of creating the colored display, the design team can choose one of two methods.

The first is to use single color LEDs that each emit one color. These only have a cathode and anode so they do not require much I/O control. The advantages of this is that there is a broader selection of packaging, more reliability, and better thermal dissipation. However, it requires extra wiring, less color blending, and is pricier. It will also result in each chamber having a different color, so there would be multiple colors flashing at the same time, which may not be very pleasing to the eye. Therefore, another option is to go with one consistent color.

The alternative method is to use RGB LEDs that contain three separate PN junctions for each color. This way, the group will have simpler wiring, a lower cost, and more prominent color blending and saturation that is more alluring to the eye, even though there is poor thermal dissipation and less package availability. Ideally, they want each chamber to be the same color at any given time so they have a consistent show. The challenge would be to get them to be synchronized and even the slightest deviation can throw off the illusion.

When it comes to shapes, LEDs can be either rounded or square. Round LEDs are the most common and although it won’t be completely necessary, it gives us a better viewing angle. The square based LEDs have a small dome on top and can be placed easily on a flat surface, so the viewing angle is not as good as the round LEDs but would not make a detrimental difference. Furthermore, the leads are much shorter and would entail more complex wiring.

5.8.6 LED Selection

The design team wanted an LED that will generate a bright output with minimal driving current. Since the LED light output has to go through the acrylic case as well as the water column, they expect a significant amount of light attenuation. The LED that was selected is the YSL-R547W2C-A13 from China Young Sun LED Technology Company. These are high output LEDs and are readily available from Sparkfun for a relative low cost of $0.95 per piece. They are available in multiple colors, and red, blue, and green single color LEDs were selected for the project. Below on table 5.8.6 are the specifications for the LEDs.

The luminous intensity specification for the LED is 8000 mcd minimum which is quite bright. In fact, the reviews for this LED indicated that “The LEDs are so bright that it hurts to look directly at them”. To ensure a bright LED display, looking through the water column, the design team decided to us two of them in series. The forward current specification for the LED is 20 mA. This is exactly

Items	Absolute Maximum Rating	Unit
Forward Current	20	mA
Peak Forward Current	30	mA
Suggestion Using current	16-18	mA
Reverse Current (V=5V)	10	mA
Power Dissipation	105	uA
Operation Temperature	40-85	mW
Storage Temperature	40-100	C
Lead Soldering Temperature	260 Degrees for 3 Seconds max	C
Forward Voltage	V	V
Wavelength	7000	K
Luminous Intensity	8000	mcd
50% Viewing Angle	10	deg

Table 5.8.6: LED Specifications

what they programmed the LED driver constant current source to generate (see previous section on the LED driver). The forward voltage specification for the LED is 3.4V maximum. Since they are using 15V to drive the LED, they will have more than adequate voltage to drive the 3.4V LED.

5.9 Water Pumps

In order to make this project a Dancing Water Spectrum Analyzer that uses water, different ways of displaying the frequency spectrum with water was explored. One option was to use a large water pump and a network of PVC pipes. The PVC pipes would have sixteen outlets for the water to act like a fountain. Each of the sixteen outlets would represent a specific audio frequency range. At the end of each outlet there would be a solenoid rotary shutter. The shutters would be controlled by the microprocessor to block or unblock the water flow. If the water was being blocked by the shutter that would mean that there was no audio frequency in that range. This set up would allow only active audio frequency ranges to be working. The result is an on or off Dancing Water Spectrum Analyzer. This is not what the group was looking for in the project. The goal of the project was to have water mimic a typical music spectrum analyzer that uses an LED display of various heights to signify the magnitude of the frequency in the specific frequency ranges. After more research the group came up with the idea of having sixteen individual water pumps. Each water pump would be individually controlled by the PIC32 microprocessor to have the water be various heights. This format with sixteen individually controlled water pumps would satisfy the goal the group wanted to display the magnitude of the audio frequency spectrum. There are various types of water pumps ranging from small outdoor fountain pumps to large pond pumps. The next section will discuss the method of determining what water pump would best fit the project.

5.9.1 Comparison of Various Pumps

This section will discuss various types of water pumps and the reasoning behind choosing the Magicfly DC30A-1230. As discussed in the previous section, the group's first idea was to have one large water pump with a network of PVC pipes to act like a fountain. When research was done in determining if this display method was appropriate, different large water pumps were taken into consideration. These pumps require an AC power source and can cost easily upwards of fifty dollars. Table 5.9.1-1 compares three pond pumps. The water pump with the highest water flow rate is the Atlantic WG TT1500. This pump also has the largest dimensions and is the most expensive. This high flow rate is not required for a small fountain and will be excluded. The Pondmaster Pond Mag 2 has the lowest water flow rate and is the smallest pump but its' cost is higher than the Aquascape Ultra Pump 400. The Aquascape Ultra Pump 400 would have been the chosen water pump if a single water pump was being used as the water display. This is because the Aquascape Ultra Pump 400 is less expensive than the Pondmaster Pond Mag 2.

	Atlantic WG TT1500	**Pondmaster Pond Mag 2**	**Aquascape Ultra Pump 400**
Voltage	110-120V AC	110-120V AC	110-120V AC
Frequency	60Hz	60Hz	60Hz
Power	101W	N/A	24W
Max Flow Rate	1640GPH	250GPH	370GPH
Dimensions	203.2x133.4x158.8mm	114.3x96.5x88.9mm	139.7x101.6x88.9mm
Weight	N/A	1.8kg	N/A
Price	$159.99	$78.19	$67.98

Table 5.9.1-1: AC Pond Pump Parameters

The large water pumps would not be used because there is no way to vary the water height due to a solenoid rotary shutter being needed to stop the water flow. The next group of water pumps are AC fish tank pumps in Table 5.9.1-2. These pumps are much smaller than the AC pond pumps in Table 5.9.1-1. With the small fish tank pumps, sixteen could be mounted in a line to form the water display. The PonicsPumps 12005 is a good choice because of its low power requirements. The downside to this pump is its price. Uniclife has three pumps of varying water flow rates. Of the three Uniclife pumps the UL-381 would be best suited for the project because it has the lowest power requirements and the smallest dimensions. Comparing the PonicsPumps 12005 and the Uniclife UL-381, the Uniclife UL-381 is the better option due to the lower price and smaller dimensions. Having the sixteen water pumps small in size will help make the entire display easily portable.

	PonicsPumps 12005	**Uniclife UL-381**	**Uniclife UL-401**	**Uniclife UL-403**
Voltage	120V AC	110-120V AC	110-120V AC	110-120V AC
Frequency	60Hz	60Hz	60Hz	60Hz
Power	6W	4W	15W	20W
Max Flow Rate	119GPH	80GPH	210GPH	320GPH
Dimensions	65x42x49.5mm	47x43x30mm	75x65x53mm	85x71x57mm
Weight	N/A	N/A	360g	500g
Price	$14.99	$7.99	$13.49	$15.99

Table 5.9.1-2: AC Fish Tank Pump Parameters

Using the AC fish tank pumps for the water display has the challenge of needing AC voltage to vary the heights of the water. Since the pumps are using AC voltage this is difficult because in order to control the voltage supplied to the pumps to vary the height of the water a PIC32 microcontroller is being used. The PIC32 microcontroller has DC output voltages. Having the water pumps use DC voltage instead of AC voltage would be beneficial instead of having to convert DC voltage to AC voltage for the pumps in Table 5.9.1-2. A few DC water pumps were researched and put into Table 5.9.1-3. The ZKSJ DC40E-1250 water pump has the highest flow rate and requires the most power of all the pumps in the table. This water pump is also the most expensive and has the largest dimensions. Due to the reasons stated before the ZKSJ DC40E-1250 will be excluded because the project does not need a pump with its capabilities. The TSSS Hmax AD20P-1230C is the least expensive DC pump but it weighs more than the Magicfly DC30A-1230 and Docooler DC30A-1230. One of the goals of the project is portability and having lightweight water pumps are ideal. This removes the TSSS Hmax AD20P-1230C from the choice of possible pumps. Now comparing the Magicfly DC30A-1230 and the Docooler DC30A-1230, the Docooler DC30A-1230 is slightly larger in dimensions, weight, power and price.

	Magicfly DC30A-1230	**Docooler DC30A-1230**	**ZKSJ DC40E-1250**	**TSSS Hmax AD20P-1230C**
Voltage	12V DC	12V DC	12V DC	12V DC
Power	4.2W	4.8W	13.5W	4.8W
Max Flow Rate	240l/h	240l/h	500l/h	240l/h
Dimensions	51x34x42.7mm	52x46x55mm	80x64.4x48mm	53.3x38.1x35.6mm
Weight	50g	65g	258g	105g
Price	$10.99	$11.69	$18.80	$8.22

Table 5.9.1-3: DC Water Pump Parameters

Taking all the discussed water pumps into consideration the Magicfly DC30A-1230 is the ideal DC water pump for the project. The Magicfly DC30A-1230 has adequate water flow rate, requires the least amount of power to operate, and has the lowest weight. Assembling sixteen of these water pumps to form the water display is feasible because of their small size and price.

5.9.2 Design

A closer look at the requirements and specifications of the Magicfly DC30A-1230 water pump will be discussed in this section. This is important in order to know the electrical and physical properties of the water pump and then assembling sixteen water pumps together to become the water display. The Magicfly DC30A-1230 is the DC water pump that will be used for the Dancing Water Spectrum Analyzer. Table 5.9.2-1 contains all the necessary information pertaining to the project's electrical and physical components. The water pump uses a 12V DC input with a max current of 0.35A and consumes 4.2W of power. The max flow rate of 240l/h is adequate to create the fountain display. Having small dimensions of 51x34x42.7mm is beneficial to mount sixteen Magicfly DC30A-1230's in a row. The water pump's light weight of 50g will help the project remain a portable display. A 140ºF temperature rating is within the limits of the project. The noise of the water pump being 35dB means that it is the same volume as a quiet conversation between people. The Magicfly DC30A-1230's noise will not interfere with the audio music being played by the project. A life span of over 30000hrs will ensure that the water pumps will not need replacing unless one breaks for the duration of this project. A price of $10.99 will allow the team to purchase sixteen Magicfly DC30A-1230's within a reasonable price.

Magicfly DC30A-1230	
Voltage	12V DC
Max Current	0.35A
Power	4.2W
Max Flow Rate	240l/h
Dimensions	51x34x42.7mm
Weight	50g
Max Temperature	140ºF
Noise	35dB
Life Span	Above 30000hrs
Price	$10.99

Table 5.9.2-1: Magicfly DC30A-1230 Parameters

In order to build the display structure a closer look at the dimensions of the Magicfly DC30A-1230 is necessary. Figure 5.9.2-1 is the front view of the water pump. The Magicfly DC30A-1230 has a length of 34mm but to attach the water pump to the display structure a bracket is used. This bracket increases the length of the water pump to a total of 50.5mm. The height is unaffected and remains at 42.7mm from the base to the water output.

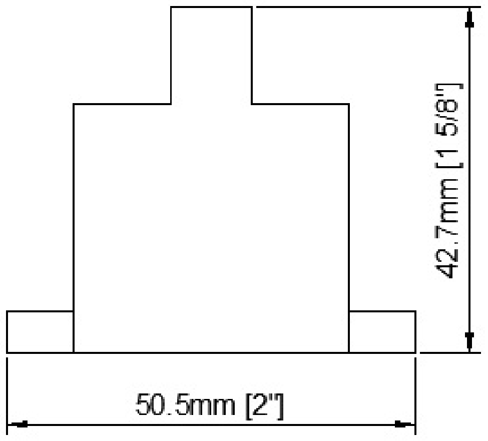


Figure 5.9.2-1: Magicfly DC30A-120 Front View

The next figure is of the Magicfly DC30A-1230's side view. Figure 5.9.2-2 shows that the width of the water pump is 51mm from the back to the water intake.

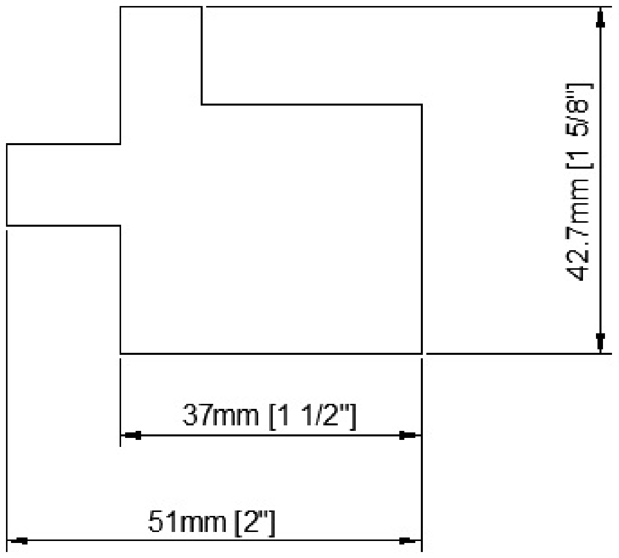


Figure 5.9.2-2: Magicfly DC30A-1230 Side View

Sixteen of Magicfly DC30A-1230 will be used for the water display. Table 5.9.2-2 lists the requirements of having sixteen water pumps being used in unison. The required voltage for the water pumps are connected to controlling op-amps. These op-amps are connected in parallel and require 15VDC total. The max current that is required to be supplied for sixteen water pumps is 5.6A. The Magicfly DC30A-1230 may have a startup current that draws more than 0.35A each. Each water pump will have 0.75A available to compensate for any excess startup current. This increases the total supplied current to 12A. The maximum power for all sixteen water pumps is 67.2W. Once all the water pumps are mounted the total length will be 808mm or about 32in. The combined weight of sixteen Magicfly DC30A-1230's is 800g or 1.76lbs. This low weight is a great start to having the Dancing Water Spectrum Analyzer be a portable water display. The total cost of the water pumps will be $175.84 and is within the projects budget.

Magicfly DC30A-1230 (x16)	
Voltage	12V DC
Max Current	5.6A
Power	67.2W
Length	808mm / 32in
Weight	800g / 1.76lbs
Price	$175.84

Table 5.9.2-2: Sixteen Magicfly DC30A-1230 Parameters

Figure 5.9.2-4 is a visual representation of the sixteen Magicfly DC30A-1230's. This is the front view with the water pumps mounted side by side. Each water pump has a mounting bracket that increases the length of each individual water pump. Combining all the water pumps in line will have a total length of 808mm or about 32in and a height of 1.625in.

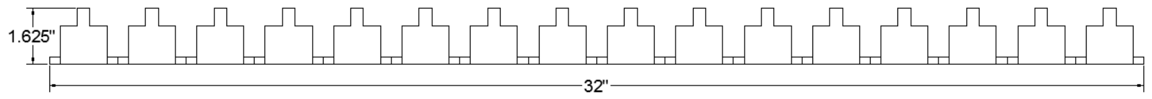


Figure 5.9.2-3: Sixteen Magicfly DC30A-1230 Front View

5.10 Speakers

The speakers for the Dancing Water Spectrum Analyzer will be purchased or borrowed. The speakers will use a standard wall plug for power. The input to the Dancing Water Spectrum Analyzer is an analog jack. A computer or Ipod or another electronic device will supply the song to be analyzed and displayed. The speakers will be connected to the computer or electronic device and provide the audio that goes with the water display. The volume of the speakers will be adjustable by the user.

5.11 Power Supply

This section will discuss two methods of designing the power supply for the Dancing Water Spectrum Analyzer. The group's initial design for the power supply was using a non-switching power supply. This method once analyzed turned out to be large and expensive. The linear regulators used would also require large heatsinks. Upon further research the group decided on a switching power supply because of its efficiency and small size. Once the switching power supply was designed the cost of purchasing a PCB and parts was over the groups budget because two PCBs were already being purchased for the microprocessor and the controlling op-amps.

The group was unable to build a power supply for the project and one had to be purchased in order for the project to be completed. Table 5.11-1 shows the specifications for the bought SP320-15 power supply. It takes an input voltage of 120VAC from the wall outlet and provides and output voltage of 15VDC at 20A. This is the correct output voltage needed to power the controlling op-amps. The total power provided by the power supply is 300W and the price is $64.37.

SP320-15	
Input Voltage	120VAC
Output Voltage	15VDC
Output Current	20A
Output Power	300W
Price	$64.37

Table 5.11: Power Supply Specifications

The Dancing Water Spectrum Analyzer will use sixteen Magicfly DC30A-1230’s water pumps to accomplish the water display. In order for these components to function properly adequate power must be supplied. Each Magicfly DC30A-1230’s has a maximum voltage of 12V and a maximum current of 0.35A when it is at full power. In order to compensate for any extra startup current from the water pumps, double the amount of current will be available to each Magicfly DC30A-1230. This means that each water pump needs to have access to 0.7A of current. Since there are sixteen water pumps that will be supplied with 0.7A, a total of 11.2A needs to be available from the power supply.

5.11.1 Non-Switching Power Supply

The initial design idea for the power supply was using linear regulators and a full wave bridge rectifier. The voltage and current will be regulated by four LM350 linear regulators connected in parallel. Each LM350 will have 15V and 3A output. A single LM350 will provide the voltage and current needed to power four of the op-amps that will be controlling the voltage provided to the Magicfly DC30A-1230 water pumps. The four controlling op-amps will be connected in parallel to the LM350 and each op-amp will have a single Magicfly DC30A-1230. Since the LM350 will provide a current of 3A the current available to each op-amp will be 0.75A. Providing 0.75A to each of the water pumps will ensure that it will function even if it is required to turn on and off multiple times. The PIC32 microcontroller, D/A converters and LED’s will receive their power from a LM317 linear regulator. The single LM317 will provide 15V and 1.5A to the main circuit board. The LM317 is also connected in parallel with the four LM350’s. The input voltage for the four LM350’s and the LM317 is 18.6V. This input voltage is within the LM350’s and LM317’s application requirements. The main power supply is connected to a wall outlet that is 115VAC and 60Hz. The wall outlet voltage is lowered by a transformer to 20VAC. After the transformer a full wave bridge rectifier is used to convert the 20VAC to 18.6VDC. The 18.6V direct current is used as the input voltage for the LM350 and LM317 linear regulators. Once the voltage and current is regulated by the linear regulators it is sent to the main circuit board that holds the water pump controlling op-amps by a 15pin connector cable. Figure 5.11.1 is the wire diagram from the wall outlet to the connector cable as described.

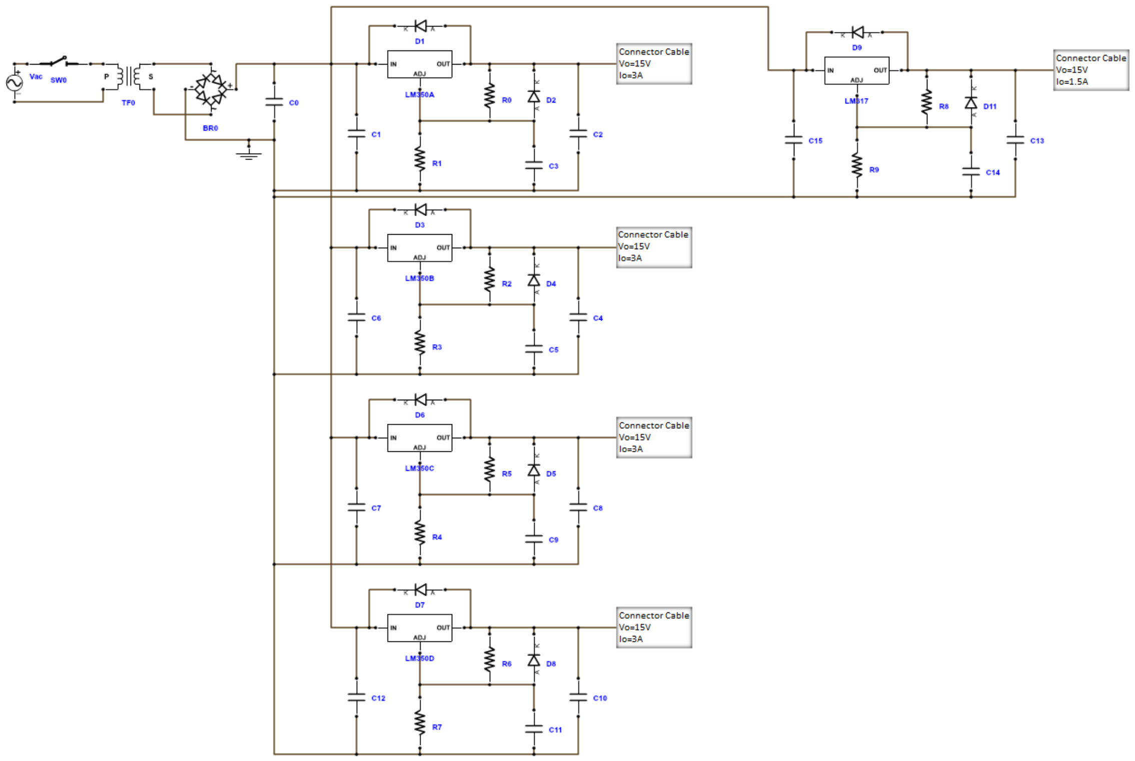


Figure 5.11.1: Wire Diagram of Non-Switching Power Supply

5.11.1.1 Power Source

This section will go into further detail on the components used for the non-switching power supply. A wall outlet of 115V at 60Hz will be the source of the power supply. From there a single pole single throw (SPST) switch will be used as an on-off switch. This will prevent any unwanted usage of the Dancing Water Spectrum Analyzer and will provide a means of cutting off the power to the project if any errors occur. After the switch there will be a transformer to lower the voltage to 20V and the current to 10A. A full wave bridge rectifier will convert the voltage from AC to DC to be used by the linear regulators.

The on-off switch will be a simple SPST switch for easy installment and usage. Table 5.11.1.1-1 is a list of switches that were taken into consideration. Each switch has a voltage rating of 125VAC because the switch will be located before the transformer lowers the voltage or the full wave bridge rectifier converts AC to DC voltage. The maximum current that can pass though the switch will need to be at least 10A because that is what is required for the sixteen Magicfly DC30A-1230 water pumps to operate. This requirement means that the NKK Switches CWSC11JCACS switch will have to be excluded. The switches are SPST for easy connections and usage. The CW Industries GRS-4011-1600 switch has no markings for on-off and could confuse the user if the Dancing Water Spectrum Analyzer has power or not and will be taken out of consideration. The project is most likely going to be used in a dark or dimly lighted area in order to see the LEDs that will be shown on the water display. This means that having a switch that lights up is beneficial in the user being able to locate it. The Cherry CRE22F4DBBNE switch does not light up and will be excluded for this project. The remaining E-Switch RSC241D1023-135 switch does light up in a green color to be easily identifiable. It also has on-off marking so that if the room is dark or bright the user can clearly see if the Dancing Water Spectrum Analyzer is powered on or not. The E-Switch RSC241D1023-135 dimensions and snap-in mounting make it easy to install on the power supply box. This switch is also the lower price of the two backlighted switches.

	NKK Switches CWSC11-JCACS	**E-Switch RSC241D-1023135**	**Cherry CRE22F4-DBBNE**	**CW Industries GRS-4011-1600**
Voltage	125VAC	125VAC	125VAC	125VAC
Max Current	9A	20A	20A	16A
Switch Type	SPST	SPST	SPST	SPST
Markings	None	On-Off	On-Off	None
Color	Red	Green	Black	Black
Lighted	Yes	Yes	No	No
Dimensions	19.2x12.9mm	27.2x12.1mm	28.5x12.1mm	19.2x12.9mm
Mount Type	Snap-In	Snap-In	Snap-In	Snap-In
Price	$4.49	$2.10	$1.20	$0.70

Table 5.11.1.1-1: Switch Parameters

After the switch a transformer is used to lower the voltage from 115VAC to 20VAC. Another consideration when choosing a transformer is the amount of current that it can handle. Four transformers and their parameters are in Table 5.11.1-2. The voltage across the LM350's and the LM317 is an input-output differential of 35V. The output voltage of the linear regulators will be 15V and having an input voltage a few volts above 15V will help to keep the temperature of the linear regulators low. The linear regulators will need a heatsink for either case but having a low input-output voltage differential would be ideal. Before the voltage can reach the linear regulators it must pass a full wave bridge rectifier. The full wave bridge rectifier contains four diodes that consume a small amount of voltage. The voltage for each diode will be considered to be 0.7V for this analysis. Only two diodes are operational at time during the full wave bridge rectifier so a voltage drop of 1.4V must be taken into account when finding the voltage supplied to the linear regulators. The linear regulators must have at least an input voltage of 16V. Adding the voltage drop of the diodes the voltage required is about 18V. To provide adequate voltage to the linear regulators the transformer needs to have a secondary voltage close to 20V. This initial analysis when compared to the transformers in Table 5.11.1.1-2 means that the Hammond Manufacturing 165S18 transformer is taken out of consideration because its secondary voltage is 18V. Each transformer has an output current of 10A to provide enough current for the Magicfly DC30A-1230 water pumps to operate. The dimensions of the transformers will not inhibit the decision on what transformer to use for the project. The voltage difference is only 1V when comparing the Hammond Manufacturing 165S25 transformer to the Triad Magnetics F-401U transformer but the price difference is $24.2. The Hammond Manufacturing 165S25 transformer will be taken out of consideration due to its cost. This leaves the two Triad Magnetics F-259U and F-401U transformers. The main difference between these two transformers the secondary voltage of either 20V or 24V. The 20V Triad Magnetics F-259U is the chosen transformer because the secondary voltage of 20V meets the design requirements for the project.

	Triad Magnetics F-259U	**Triad Magnetics F-401U**	**Hammond Manufacturing 165S25**	**Hammond Manufacturing 165S18**
Primary Voltage	115V	115V	115V	115V
Secondary Voltage	20V	24V	25V	18V
Output Current	10A	10A	10A	10A
Max Power	200W	240W	250W	180W
Dimensions	87.3x88.9x 104.8mm	87.3x95.3x 104.8mm	95.5x101.6x 79.5mm	95.3x88.9x 79.5mm
Price	$42.48	$42.48	$66.68	$59.81

Table 5.11.1.1-2: Transformer Parameters

Now that the switch and the transformer are chosen for the power supply Figure 5.11.1.1-1 is the Multisim circuit diagram with the full wave bridge rectifier and smoothing capacitor. Before finding the value of the smoothing capacitor the voltage ripple needs to be in an acceptable range. With the Triad Magnetics F-259U transformer and two diodes the voltage after the full wave bridge rectifier is 18.6Vrms and the peak voltage is 26.3Vp. The voltage ripple cannot be lower than 16V for the LM350's and the LM317 to operate normally. For this analysis of the project a voltage ripple of 5V is assumed. With the maximum voltage being 26.3V and the ripple voltage at 5V the minimum voltage is 21.3V. This minimum voltage is adequate for the linear regulators. In order to find the value for the smoothing capacitor Equation (5.11.1-1) is used with the values determined previously.

$$C = \frac{I_o}{2*f*V_{ripple}} \tag{E5.11.1-1}$$

Calculating the value for the smoothing capacitor with $I_o = 10A,\ V_{ripple} = 5V$, and $f = 60Hz$, the capacitor value is $C = 16700\mu F$. A higher capacitance smoothing capacitor will be used at a value of $C = 22000\mu F$ to lower the ripple voltage even further. Figure 5.11.1.1 is the complete circuit from the AC wall outlet to the DC voltage for the linear regulators. The AC source is representing the wall outlet at 120Vrms and 60Hz. U1 is the Triad Magnetics F-259U transformer. Next the four diodes is the full wave bridge rectifier and C1 is the smoothing capacitor. V_{out} is the voltage being sent to the four LM350's and the LM317 at 18.6Vrms.

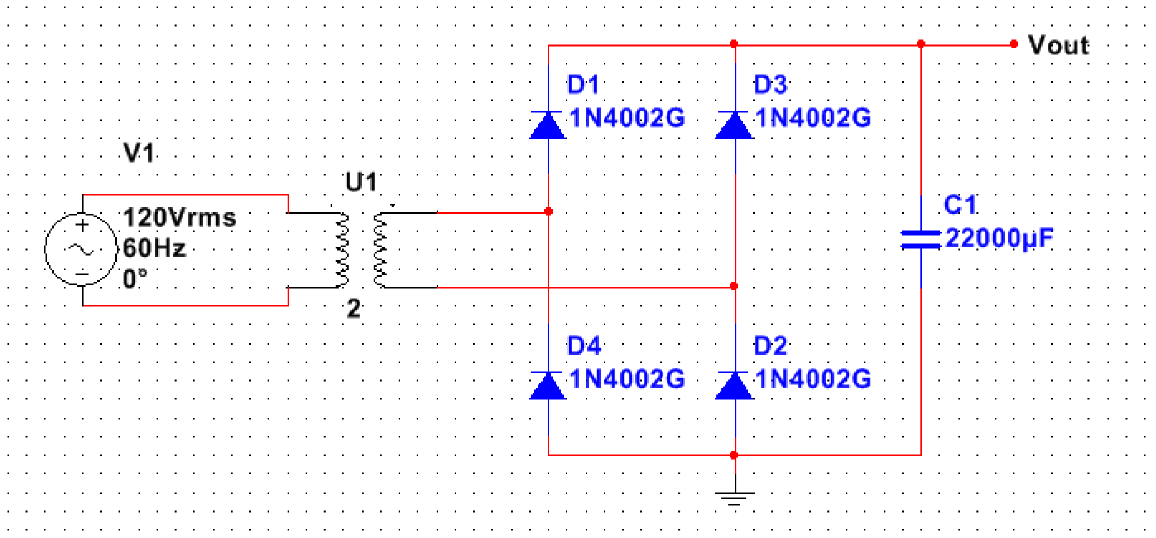


Figure 5.11.1.1: AC to DC Voltage Conversion

5.11.1.2 Voltage Regulators

The Dancing Water Spectrum Analyzer will use four LM350's and one LM317 to regulate the voltage and current being sent to the main circuit board and the op-amps controlling the Magicfly DC30A-1230 water pumps. This section will discuss the components used by the Texas Instruments LM350 and LM317 linear regulators.

5.11.1.2.1 LM350

The four LM350 linear regulators are being used to supply 15V and 3A to sixteen op-amps that control the water pumps for the Dancing Water Spectrum Analyzer. Each LM350 will be regulating the voltage being sent to four controlling op-amps. The linear regulator LM350 has an adjustable output voltage from 1.2V to 33V and a guaranteed output current of 3A. The LM350 is made by Texas Instruments. From the AC to DC voltage conversion the voltage being sent to the LM350 is 18.6Vrms. The desired output voltage from the linear regulator is 15V and output current is 3A. The LM350 has an input-output differential voltage of 35V. To have an output current of 3A the input-output differential voltage must be equal to or less than 10V. The projects input-output differential is 3.6V and meets the requirements to achieve a 3A output current. The Texas Instruments LM350 datasheet provides the basic components and layout for the LM350 as shown in Figure 5.11.2.1. The resistors R_3 and R_5 are required in order to set the output voltage of the linear regulator. Having R_3 at a value of 120Ω is recommended for the LM350. The resistor R_5 is the adjustable resistor that is calculated using Equation E5.11.2.1-1 so that the linear regulator has the desired output voltage.

$$V_o = V_{ref}\left(1 + \frac{R_5}{R_3}\right) + I_{adj}R_5 \qquad \text{(E5.11.2.1-1)}$$

The reference voltage for the LM350 is $V_{ref} = 1.25V$. The adjustable current I_{adj} is considered negligible because it typically has a value of 50µA. The equation to calculate the output voltage then becomes Equation E5.11.2.1-2.

$$V_o = V_{ref}\left(1 + \frac{R_5}{R_3}\right) \qquad \text{(E5.11.2.1-2)}$$

Since the project requires an output voltage of 15V, the resistor R_5 needs to be solved from Equation E5.11.2.1-2. Solving for the resistor R_5 with the values of $V_{ref} = 1.25V,\ V_o = 15V,$ and $R_3 = 120\Omega,$ R5 becomes 1320Ω. The capacitor C2 at 10µF is recommended to smooth the ripple rejection by preventing the output ripple voltage from being amplified. A 100nF capacitor, C_{15}, is used as a voltage smoothing capacitor. The C_{16} capacitor of value 1µF is recommended to improve the transient response of the circuit. The diode D_1 is acting as a protection diode and will prevent the capacitor C_{16} from discharging over the output voltage. The diode D_2 is another protection diode for the C_2 capacitor to stop it from also discharging into the output voltage. The finished circuit diagram with all component values is shown in Figure 5.11.1.2.1 for a LM350 with an output voltage of 15V and output current of 3A.

5.11.1.2.2 LM317

A single LM317 linear regulator is being used to supply the voltage and current to the main circuit board containing the PIC32 microcontroller, D/A converters, and LED's. These electronic components will be provided with 15V and 1.5A. The LM317 is a linear regulator made by Texas Instruments. This linear regulator has an adjustable output voltage from 1.25V to 37V with a guaranteed output current of 1.5A. The voltage being supplied to the LM317 by the full wave bridge rectifier is 18.6Vrms. The output voltage from the LM317 will be 15V and have an output

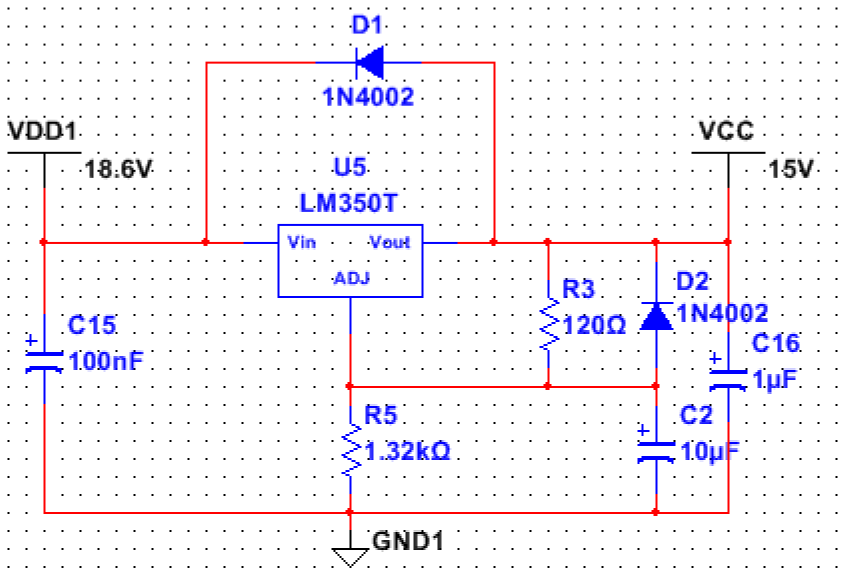


Figure 5.11.1.2.1: Linear Voltage Regulator LM350

current of 1.5A. The maximum input-output differential voltage for the linear regulator is 40V. Since the desired output current is 1.5A, the input-output differential voltage must be less than or equal to 15V. The input-output differential voltage is 3.6V for the LM317 and meets the requirements to achieve a 1.5A output current. The datasheet for the Texas Instruments LM317 provides the basic components and layout for the linear regulator as shown in Figure 5.11.2.2-1. The output voltage for the linear regulator is determined by the resistors R3 and R5. The datasheet for the LM317 recommends that resistor R3 has a value of 240Ω. The adjustable resistor R5 is used to calculate the output voltage of the linear regulator in Equation E5.11.2.2-1

$$V_o = V_{ref}\left(1 + \frac{R_5}{R_3}\right) + I_{adj}R_5 \qquad \text{(E5.11.2.2-1)}$$

The reference voltage is $V_{ref} = 1.25V$ for the LM317. The adjustable current I_{adj} typically has a value of 50µA and is considered negligible. The equation to calculate the output voltage then becomes Equation E5.11.2.2-2.

$$V_o = V_{ref}\left(1 + \frac{R_5}{R_3}\right) \qquad \text{(5.11.2.2-2)}$$

The required output voltage for the LM317 to provide enough power to the main circuit board is 15V. Equation E5.11.2.2-2 will be used to solve for R5 to have an output voltage of 15V. When $V_{ref} = 1.25V, V_o = 15V$, and $R_3 = 240\Omega$ in Equation 5.11.2.2-2 the value of R5 is 2640Ω. In order for the output voltage ripple to not be amplified a capacitor C2 with a value of 10µF is used. The capacitor C15 with a value of 100nF is acting as a smoothing capacitor for the input voltage. To improve the transient response of the circuit the C16 capacitor of value 1µF is recommended. In order to prevent the capacitor C16 from discharging on the output voltage, a protection diode D1 is used. Another diode D2 is also acting as a protection diode for capacitor C2 to stop it from discharging across the output voltage. Figure 5.11.1.2.2 is the finished circuit diagram with all the component values for a LM317 with an output voltage of 15V and output current of 1.5A.

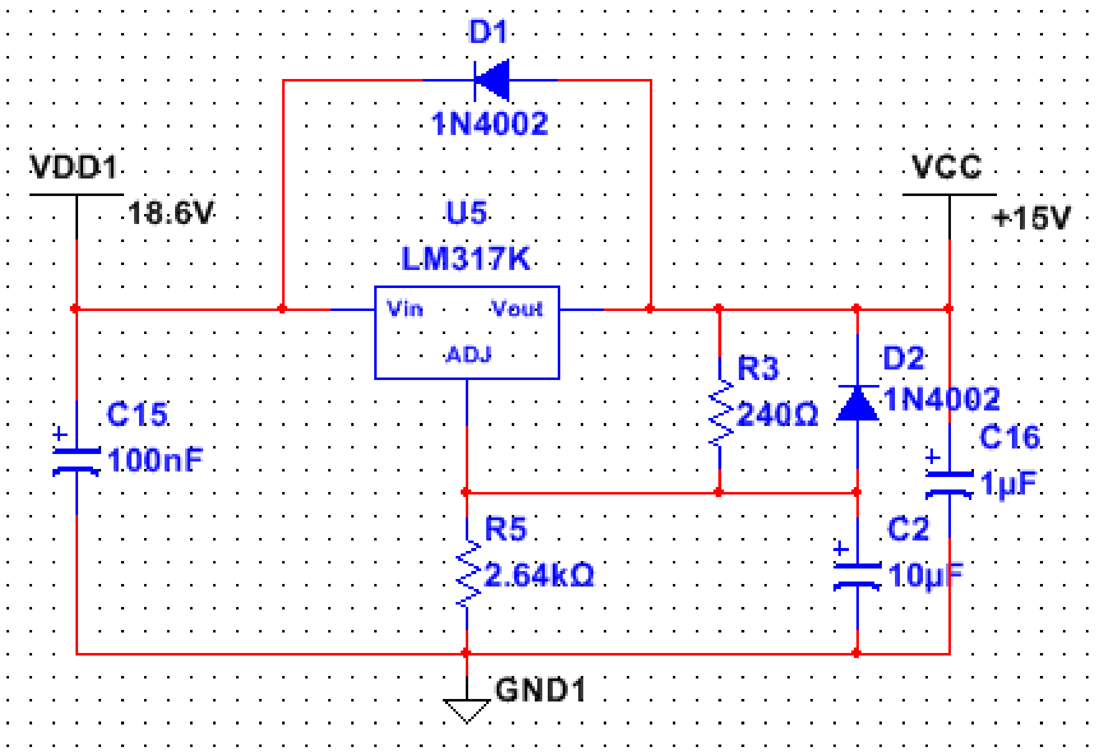


Figure 5.11.1.2.2: Linear Voltage Regulator LM317

5.11.2 Switching Power Supply

With further research a switching power supply was chosen to provide power to the Dancing Water Spectrum Analyzer. The voltage and current will be regulated by four TPS54531 switching regulators connected in parallel. The input to the switching regulators will be from an AC to DC power converter that will supply 20V and 12A. The output of the TPS54531's is designed to be 15V and 3A. One of the switching regulators will be used to supply power to four of the op-amps that will be controlling the voltage to the Magicfly DC30A-1230 water pumps. The op-amps will be connected in parallel with the switching regulator and be supplied with 15V and 0.75A each. This will provide adequate power for the water pumps to function properly. A single TPS54332 switching regulator will provide the voltage and current to the main circuit board to power the PIC32 microcontroller, D/A converters and LEDs. This switching regulator will supply 15V and 1.5A and it is also connected in parallel to the four TPS54531's. The power supply will convert a wall outlet voltage of 115VAC and 60Hz to 20V and 12A to power the switching regulators. After each switching regulator the output voltage and current will be sent to the main circuit board by a 15pin connector cable. The wire diagram from the wall outlet to the connector cable is shown in Figure 5.11.2.

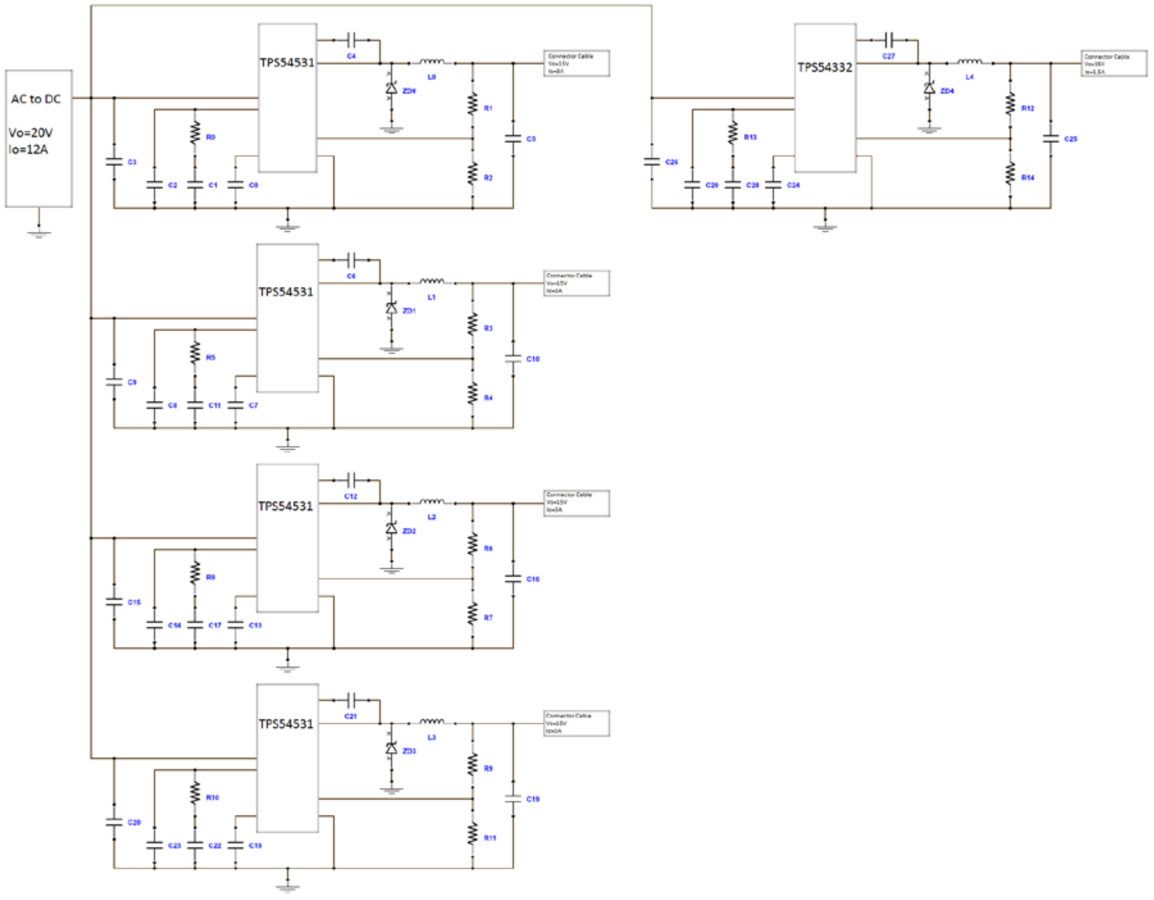


Figure 5.11.2: Wire Diagram of Switching Power Supply

5.11.2.1 Power Source

The switching power supply was created using the web design tool from Poweresim. This section will go into further detail on the components used for the switching power supply. The maximum input for the design is $V_{in,max} = 120V_{RMS}$, $V_{inmin} = 100V_{RMS}$, $V_o = 20V$, and $I_o = 12A$. The circuit layout is shown in Figure 5.11.2.1-1. This schematic uses a wall outlet of 115V at 60Hz as the source of the power supply. A switch will be added to the schematic after the fuse in F1 and will act as a kill switch incase if any errors occur in the Dancing Water Spectrum Analyzer. A pulse width modulator U1 will control the amount of power being supplied for the rest of the circuit. Next a flyback converter is used to lower the voltage to 20V. Connected to the flyback converter is a feedback block to smooth the output voltage. The efficiency of the power supply in Figure 5.11.2.1 is about 82.1% with a power dissipation of 52.5W.

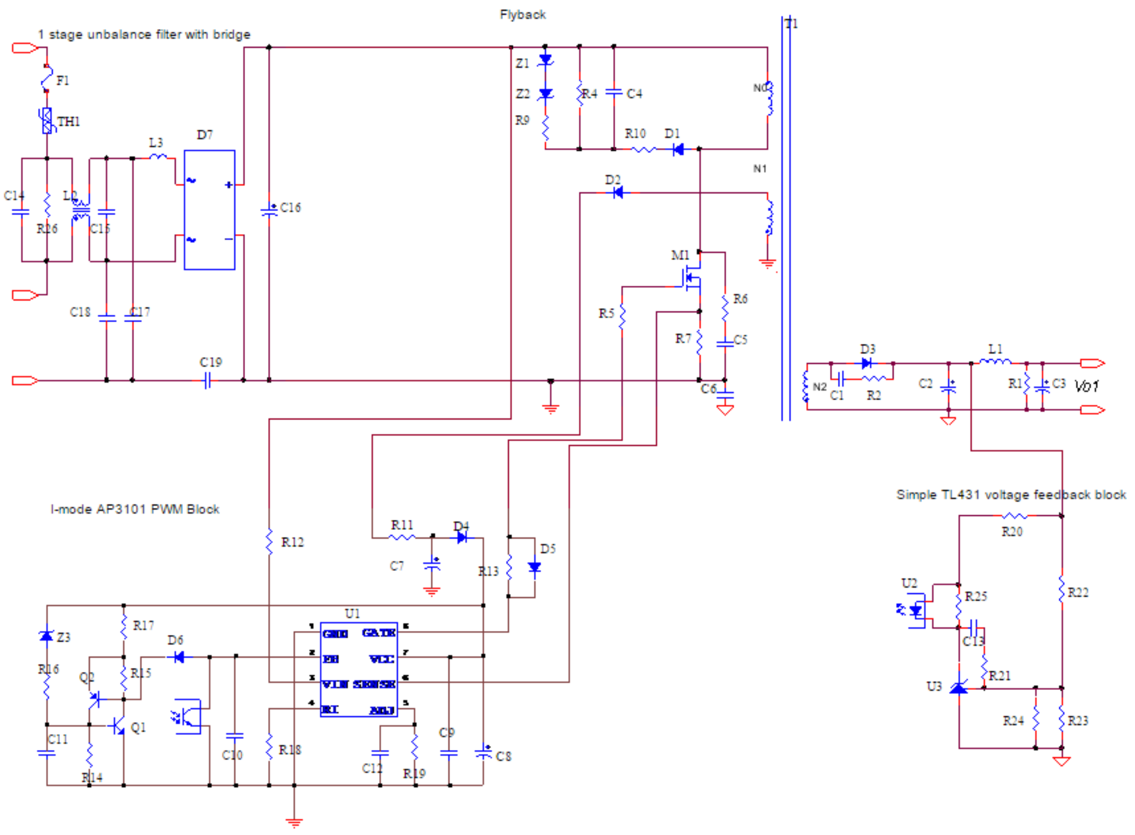

Flyback
1 stage unbalance filter with bridge
I-mode AP3101 PWM Block
Simple TL431 voltage feedback block
Vo1
U1
GND
GATE
FB
VCC
VIN
SENSE
RI
ADJ
T1
M1
D7

Figure 5.11.2.1: Switching Power Supply

All of the basic components and their values and descriptions for the circuit diagram of Figure 5.11.2.1-1 are listed in Table 5.11.2.1-1. The resistors R9 and R10 in the power supply are actually not there and are shorted. The two zener diodes, Z1 and Z2, are also shown in the schematic but are open. The diodes D4, D5, and D6 are switching diodes and act like a switch in the circuit.

Table 5.11.2.1-2 contains the more unique components and their descriptions. U1 is a pulse width modulation controller for the power supply. U2 and U3 also help to regulate the voltage of the circuit. The diodes Q1 and Q2 are three terminal diodes for switching. F1 is a safety fuse. A high voltage transistor is used in M1. T1 is a transformer with a flyback converter. L1, L2, and L3 are inductors with the specified values.

5.11.2.2 Switching Regulators

Four TPS54531 switching regulators will be used to regulate the voltage going to the Magicfly DC30A-1230 water pumps in the Dancing Water Spectrum Analyzer. In order to power the main circuit board and the controlling op-amps for the water pumps one TPS54332 switching regulator will be used. This section will discuss the components used by the Texas Instruments TPS54531 and TPS54332 switching regulators.

Ref.	Value	Ref.	Value	Ref.	Description
R1	330Ω	C1	3.3nF	D1	1A 400V DO41
R2	47Ω	C2	1.5mF	D2	1A 100V DO41
R4	10kΩ	C3	680uF	D3	20 A 100 V MBR20100CT-E1 BCD TO-220-3
R5	0Ω	C4	68nF	D4	0.6 A 85 V BAV222 INFINEON SC75
R6	100Ω	C5	100pF	D5	0.6 A 85 V BAV222 INFINEON SC75
R7	91mΩ	C6	2.2nF	D6	0.6 A 85 V BAV222 INFINEON SC75
R9	Short	C7	3.3uF	D7	15 A 400 V PEB15A400 BR154
R10	Short	C8	47uF	Z1	Open
R11	6.2Ω	C9	100nF	Z2	Open
R12	1.2MΩ	C10	10nF	Z3	27 V 0.5 W 5% 1N5254B Axial
R13	10Ω	C11	470nF		
R14	12kΩ	C12	27pF		
R15	12kΩ	C13	10nF		
R16	1kΩ	C14	1uF		
R17	1.5kΩ	C15	1uF		
R18	20kΩ	C16	560uF		
R19	4.7kΩ	C17	2.2nF		
R20	1.8k	C18	2.2nF		
R21	100	C19	2.2nF		
R22	10k				
R23	1.6k				
R24	13k				
R25	2k				
R26	180k				

Table 5.11.2.1-1: Power Supply Components

Ref.	Description
U1	AP3101M-E1 BCD SOIC-8
U2	PEO111 CTR=85% 5 kVac DIP4
U3	2.5 V 0.8% 20 ppm AZ431BZ-BTRE1 BCD TO92
Q1	40 V 0.6 A MMBT222ALT1 SOT23
Q2	40 V 0.6 A MMBT222ALT1 SOT23
F1	10 A 250 V 6.35x6.35x32.5 mm Slow_Blow GSA10
TH1	8 Ω 6 A 35 J 21.5x21.5x7 mm PEN60A80
HS1	U-Shape HeatSink
M1	0.16 Ω 650 V 20.7 A SPP20N65C3 INFINEON TO220
T1	87.9 uH EC70 H44 IFcores Flyback Type Transformer
L1	1.06 uH T50-52B 52 MICROMETALS Noise Filter Choke
L2	5.23 mH T25*15*9 HM5A ER Input Common Mode Choke
L3	400 uH T200-52 52 MICROMETALS Input Differential Mode Choke

Table 5.11.2.1-2: Power Supply Components

5.11.2.2.1 TPS54531

The switching regulator TPS54531 will need to be designed to regulate an output voltage of 15V and output current of 3A. The TPS54531 switching regulator is made by Texas Instruments and has an input voltage range from 3.5V to 28V. The input voltage of 20V is being supplied from the AC to DC power supply and meets the requirements of the switching regulator. Inside the TPS54531 switching regulator is an integrated 80mΩ High-Side MOSFET capable of handling a continuous output current of 5A. This switching regulator has a preset switching frequency of 570kHz. The project requirements are only 3A and is within the specifications of the switching regulator. Four TPS54531's will be used to power the sixteen water pumps. One switching regulator will provide 15V and 3A to four controlling op-amps and their water pumps. This will allow each controlling op-amp to have 0.75A of available current and 15V.

Figure 5.11.2.2.1-1 was created by using the Texas Instruments WEBENCH design tool. This schematic for the TPS54531 is the circuit diagram that will provide an output voltage of 15V and an output current of 3A. The resistors R_{fbt} and R_{fbb} are used in order to set the output voltage for the switching regulator. The use of a 10200Ω resistor for R_{fbt} is recommended by the TPS54531 datasheet. This means that the R_{fbb} resistor will be adjusted to acquire the desired output voltage. Equation 5.11.2.2.1-1 is used to calculate the resistor R_{fbb}.

$$V_o = V_{ref}\left(\frac{R_{fbt}}{R_{fbb}} + 1\right) \qquad (5.11.2.2.1\text{-}1)$$

The Texas Instruments datasheet for the TPS54531 recommends that the reference voltage is $V_{ref} = 0.8V$. One of the requirements for the project is to have an output voltage of 15V. Using the given values of $V_{ref} = 0.8$, $V_o = 15V$, and $R_{fbt} = 10200\Omega$, the resistor R_{fbb} becomes 576Ω. The topology for this switching regulator is a buck converter. The TPS54531 has an efficiency of 95.4% and a total power loss of 2.06W. The complete circuit diagram is in Figure 5.11.2.2.1 with an output voltage of 15V and output current of 3A.

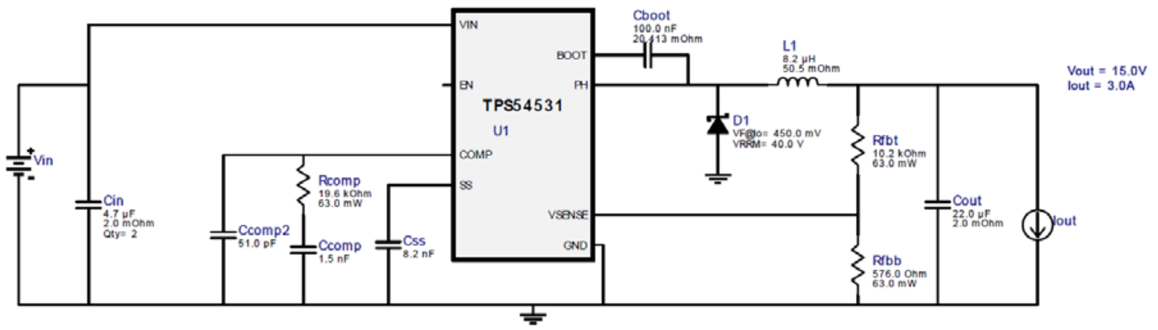


Figure 5.11.2.2.1: TPS54531 Switching Regulator V_{out} = 15V and I_{out} = 3A

5.11.2.2.2 TPS54332

In order to supply power to the main circuit board containing the PIC32 microcontroller, D/A converters and LED's a switching regulator TPS54332 will be used. These circuit components will be supplied with 15V and 1.5A. The switching regulator has an input voltage range of 3.5V to 28V and is manufactured by Texas Instruments. The input voltage range is compatible with the AC to DC power supply of 20V. An integrated 80mΩ High-Side MOSFET inside the TPS54332 is able to provide a continuous output current of up to 3.5A. The requirement for the main circuit board is 1.5A and is within the switching regulators specifications. The switching regulator has a preset switching frequency of 1MHz. A single TPS54332 will be used to regulate the power being sent to the main circuit board.

Using the WEBENCH design tool provided by Texas Instruments Figure 5.11.2.2.2-1 was created. This circuit diagram will provide an output voltage of 15V and an output current of 1.5A. To set the output voltage for the switching regulator the resistors R_{fbt} and R_{fbb} are used. The TPS54531 datasheet recommends a 10200Ω resistor for R_{fbt}. In order to get the required output voltage the resistor R_{fbb} will be adjusted. To calculate the value of R_{fbb} Equation E5.11.2.2.2-1 is used.

$$V_o = V_{ref}\left(\frac{R_{fbt}}{R_{fbb}} + 1\right) \qquad \text{(E5.11.2.2.2-1)}$$

A reference voltage of $V_{ref} = 0.8V$ is standard from the TPS54332 datasheet. The project requires an output voltage of $V_o = 15V$, and evaluating Equation 511.2.2.2-1 with $V_{ref} = 0.8$ and $R_{fbt} = 10200\Omega$, the resistor R_{fbb} is 576Ω. The switching regulators topology is a Buck converter. The switching regulators efficiency is 96.5% with a total power dissipated of 0.81W. The circuit diagram for the TPS54332 is shown in Figure 5.11.2.2.2 and has an output current of 1.5A and output voltage of 15V.

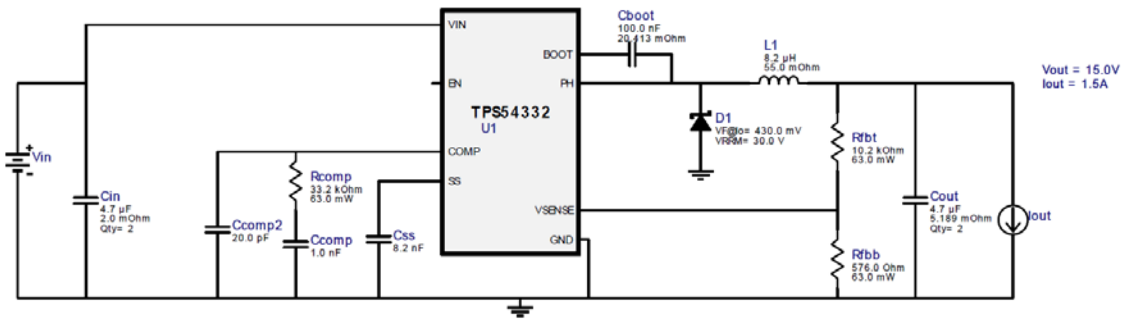


Figure 5.11.2.2.2: TPS54332 Switching Regulator V_{out} = 15V and I_{out} = 1.5A

5.12 Water Display and Reservoir

The Dancing Water Spectrum Analyzer will be a portable display. This section will cover the details of the materials used for the construction of the water display and power supply box. The dimensions of the water display will also be covered in detail along with building schematics.

5.12.1 Materials

Constructing the Dancing Water Spectrum Analyzer display will require various materials and tools. Table 5.12.1 lists all of the current materials and tools that are planned on being purchased to construct the display structure and the power supply box. The total estimated cost of all the building materials and tools is $61.89. This is within the budget and some expenditures may be further reduced by researching more material providers.

The 2x2 wood piece will be used as the mounting surface for the sixteen Magicfly DC30A-1230 water pumps. This will need to be cut to the length of the water pumps and mounted to the cover of the water reservoir. The water reservoir that the sixteen water pumps will be will be constructed from plywood and 2x4s from Lowes. The length of the sixteen water pumps is estimated at 32 inches long and will be able to fit inside the water reservoir. To make the water reservoir not leak, pond liner was used and acquired for free. Acrylic sheets will be purchased from Lowes to construct the clear display. The acrylic sheets will be cut to the desired length to enclose the fountain part of the display. To glue the acrylic together silicone sealing was used.

Item	**Price**	**Supplier**
7/16in x 4ft x 8ft Plywood	$9.35	Lowes
Acrylic	$31.76	Lowes
Silicone Sealant	$5.24	Lowes
Window Box	$10.67	Lowes
Miscellaneous	$4.87	Lowes
Total Cost	$61.89	

Table 5.12.1: Building Materials

5.12.2 Water Display Dimensions

This section will discuss in detail the dimensions of the Dancing Water Spectrum Analyzer display frame including the structure holding the sixteen Magicfly DC30A-1230 water pumps in the water reservoir and the clear acrylic display.

Figure 5.12.2-1 is the front view of the display frame and water reservoir. The water reservoir will be constructed of plywood and will have dimensions of

37”x9.25”x6”. The sixteen Magicfly DC30A-1230 water pumps have a total length of 32” and will be mounted to a 2x2 piece of wood. The length of the 2x2 wood will be 33”. The water reservoir will be lined with pond liner to prevent leaks. Figure 5.12.2-1 shows the length and the height of the display structure. It will be 38” long and 20.5” high at its maximums. The sixteen Magicfly DC30A-1230 water pumps will be positioned inside and just below the top of the box to ensure that they remain within the water reservoir. This means that the water pumps will need to be mounted 3” from the top of the box. The water pumps will be attached to the box cover to allow the water pumps to be removed from the reservoir. The cover of the box will be described later.

The fountain part of the Dancing Water Spectrum Analyzer will be housed in a clear acrylic display. Containing the water from the pumps will allow the Dancing Water Spectrum Analyzer to be positioned anywhere without the need to worry about water spilling from the fountain. The acrylic display will be made into a box with dimensions of 12”x33”x3.5”. This is the same length of the sixteen Magicfly DC30A-1230 water pumps to allow enough room for the water to be projected vertically. The max tested height of the water pumps was determined to be 13”. The group wanted the water from the pumps when at its maximum to hit the top of the acrylic display to make the project visually appealing. This is why the acrylic display is shorter than the maximum height the water from the pumps can reach. The only time the water will actually hit the top of the acrylic display would be when the magnitude of the specific audio frequency is at its maximum. Otherwise the water will not hit the top of the acrylic display.

A plywood box will be constructed to contain the LEDs for the display. This is located on top of the acrylic box with dimensions of 3”x34”x4.5”. The LED box will have two ledges 0.5” from the bottom to allow the box to rest on top of the acrylic display. It also has forms a lip around the top of the acrylic like that of a shoe box cover. This is so that the LED box does not slip or fall. The LEDs are mounted on the underside top of the box and pointing downward into the acrylic display.

The dimensions of the side of the Dancing Water Spectrum Analyzer are shown in Figure 5.12.2-2. The width of the structure holding the sixteen Magicfly DC30A-1230 water pumps 10.25” at its maximum. The water pumps will be mounted to the cover of the water reservoir by 2x4 wood 3” long. The cover to the water reservoir will have a lip around the bottom perimeter created out of plywood being 0.5” width and 1” down. This will allow the cover to act like a shoe box cover and will fit securely around the water reservoir. The LED box will be constructed like a shoe box also. The lip of the LED box will be 0.5” width and 1” down. This will sit on top and around the acrylic display. Two ledges will be attached to the inside of the LEX box to allow the box to rest on top of the acrylic display on either end. This will allow the LED box to fit security on top of the acrylic display.

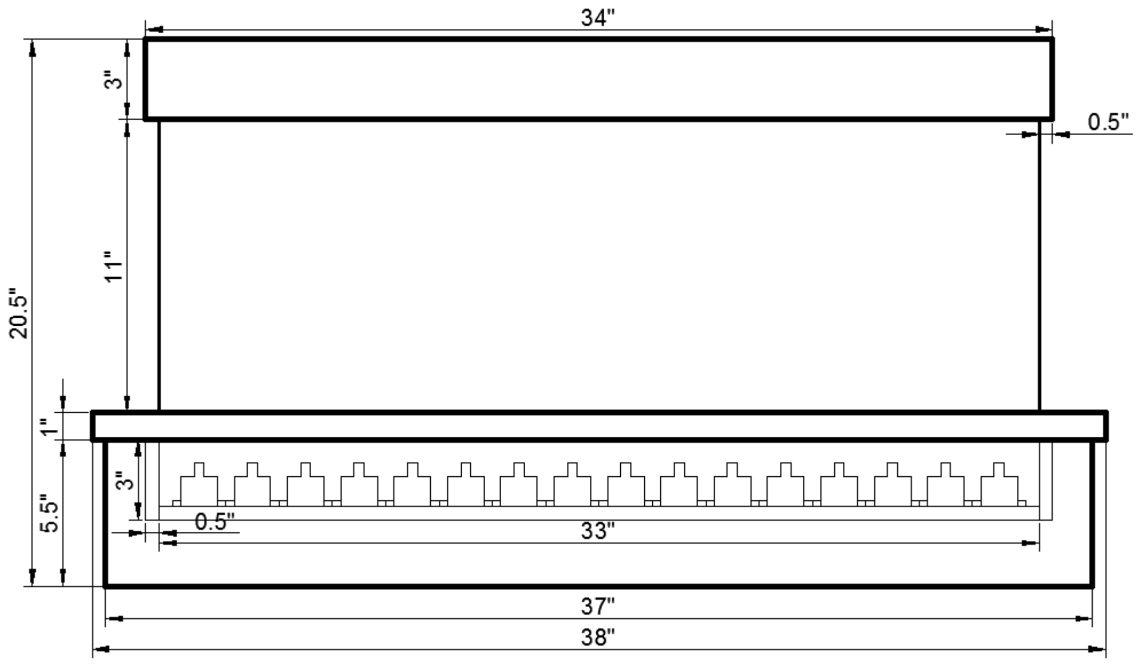


Figure 5.12.2-1: Display Front View

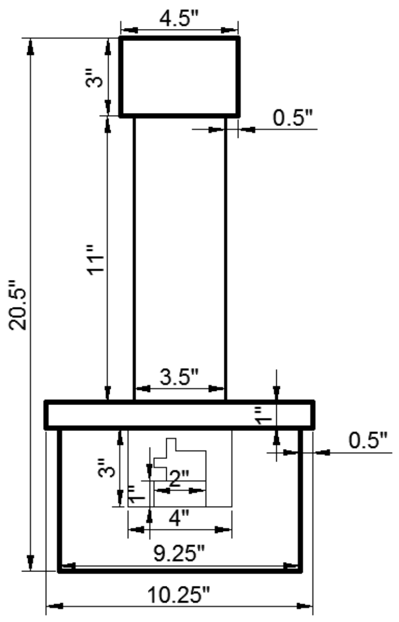


Figure 5.12.2-2: Display Side View

The lid of the Dancing Water Spectrum Analyzer was made out of plywood. The dimensions are 38”x10.25” as shown in Figure 5.12.2-3. The lid itself will have a lip of plywood around the perimeter to form a type of shoe box cover. This will allow the cover to be secure on the water reservoir. A cut in the center of the lid 33.25” long and 3.75” wide will be made to allow the Magicfly DC30A-1230 water pumps room to project the water vertically into the acrylic display. A ledge made of plywood around the hole for the water pumps will be made to hold the acrylic display in place. There will be another cut in the back of the lid 3”x2” for the water pump power cables to exit the display. The water pumps will be attached to the underside of the cover on either end of the hole.

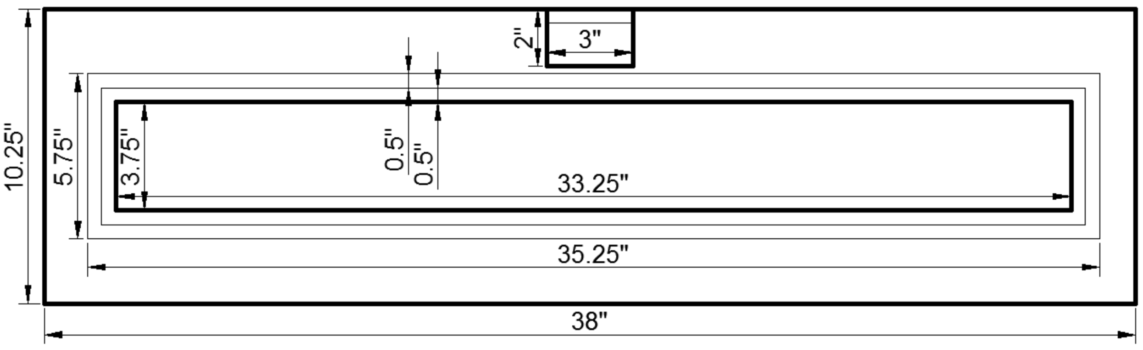


Figure 5.12.2-3: Display Top Cover View

5.12.3 Aesthetic Design

The reasoning behind the design of the water display for the Dancing Water Spectrum Analyzer is to provide a pleasant viewing experience to anyone watching. The water reservoir is being hidden inside a black box to conceal all the Magicfly DC30A-1230 water pumps and power cables. The LEDs will be mounted inside the LED box and pointing down into the water display so that the LEDs will not shine in the observers eyes. Having an acrylic box around the fountain will contain the water so that no matter where the display may be it will not leak. The cables for the water pumps and LEDs will have access on the rear of the display and it will be closed to prevent water leaking once the display is set up. Having the lid to the reservoir be removable will allow easy access to the water pumps should they need maintenance. The reservoir can be filled and emptied by removing the lid and adding or removing the water.

6.0 Project Software Design Details

This section of the report will be an overview of the mathematics that will be used in the software to control the water pumps in the Dancing Water Spectrum Analyzer. A quick version of the Fourier transform called the Decimation-in-Time Fast Fourier Transform algorithm will be used.

6.1 Decimation-in-Time Algorithm

The Dancing Water Spectrum Analyzer will utilize the Decimation-in-Time Fast Fourier Transform (DIT-FFT) to convert the music from time domain to frequency domain. This process will occur inside the PIC32 microcontroller. Once the inputted audio signal has been converted to frequency domain, specific frequency ranges will be grouped together and their magnitude value sent to the water pumps. Depending on the audio there will be different values for the set frequency ranges. This will allow the water display in the project to act like a spectrum analyzer.

The Decimation-in-Time algorithm is a short cut in the mathematics required when preforming the Fourier transform. This will allow the PIC32 microcontroller to be able to compute the required information without being slowed down by the complex calculations required in calculating the Fourier transform. The DIT-FFT algorithm is a series of multiplications and additions. The project will have sixteen water pumps that will act as sixteen outputs for the DIT-FFT.

The FFT works by decomposing an *N* point time domain signal into *N* time domain signals each composed of a single point. The second step is to calculate the *N* frequency spectra corresponding to these *N* time domain signals. Lastly, the *N* spectra are synthesized into a single frequency spectrum. The steps below show an example of the time domain decomposition used in the FFT. In this example, a 16 point signal is decomposed through four separate stages. The first stage breaks the 16 point signal into two signals each consisting of 8 points. The second stage decomposes the data into four signals of 4 points. This pattern continues until there are *N* signals composed of a single point.

1 signal of 16 points: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
2 signals of 8 points: [0, 2, 4, 6, 8, 10, 12, 14] [1, 3, 5, 7, 9, 11, 13, 15]
4 signals of 4 points: [0, 4, 8, 12] [2, 6, 10, 14] [1, 5, 9, 13] [3, 7, 11, 15]
8 signals of 2 points: [0 8] [4 12] [2 10] [6 14] [1 9] [5 13] [3 11] [7 15]
16 signals of 1 point: [0] [8] [4] [12] [10] [6] [14] [1] [9] [5] [13] [3] [11] [7] [15]

There are Log_2N stages required in this decomposition. Thus the 16 point signal requires 4 stages, and a 512 point signal requires 9 stages. This decomposition can be implemented in hardware very easily by using a bit reversal sorting algorithm, i.e. by sorting the samples according to a bit reversed order.

The next step in the FFT algorithm is to find the frequency spectra of the 1 point time domain signals. The frequency spectrum of a 1 point signal is equal to *itself*. This means that *nothing* is required to do this step.

The last step in the FFT is to combine the *N* frequency spectra in the exact reverse order that the time domain decomposition took place. Unfortunately, the bit reversal shortcut is not applicable, and the design team must go back one stage at a time. In the first stage, 16 frequency spectra (1 point each) are synthesized into 8 frequency spectra (2 points each). In the second stage, the 8 frequency spectra (2 points each) are synthesized into 4 frequency spectra (4 points each), and so on. The last stage results in the output of the FFT, a 16 point frequency spectrum.

The computational demand of the FFT is significantly less than the DFT. In fact the processing complexity is of order $NLog_2N$ instead of N^2, for the DFT. For example, a 512 point FFT is about *fifty* times faster than the standard DFT method. Thus the FFT is absolutely critical when performing frequency analysis. The FFT also has another advantage besides raw speed. The FFT is calculated more *precisely* because the fewer number of calculations are needed, and thus the results will have less round-off error.

Users of the PIC32 microprocessor have reported that a 512-point FFT can be executed in 410,000 cycles. Since the microprocessor runs at 40 MHz, this corresponds to 10.25 milliseconds. This is without using any compiler optimization. If the compiler is ran with maximum optimization level, this execution time reduces down to 146,000 cycles, or 3.65 milliseconds. The frame time is 20 Hz, and thus will have 50 milliseconds to process the data. The FFT processing will only require 3.65 out of the 50 milliseconds available and will have lots of margin in the processing timeline.

The output of the magnitude calculation block above is 256 real numbers corresponding to 256 positive frequencies in the spectrum. However, there are only 16 water pumps in the system. Thus it must be scaled from 256 frequency values into 16 frequency bands. For this project, the design team will perform a linear frequency binning. That is, we will add 16 adjacent frequencies into a single number and assign it to a single water pump. Thus each water pump will represent 16 adjacent frequency bins.

Similarly, the amplitude of each water pump output is in floating point number. Yet the DAC to control the water pump is only 10 bits, or 1024 levels. Thus the design team will need an amplitude dynamic range reduction method. For the project, the design team will first convert the floating point number into a 16-bit short unsigned integer (using the C "cast" function). Then the design team will shift the 16-bit number to the right 6 bits, effectively dividing the number by 64. Now the design team have a 10-bit number which can be sent to the DAC, and ultimately to the water pump.

Since the water pump response time is slow, the design team decided on a reasonable slow frame rate for the system. The design team settled on updating the water pump at a 20 Hz rate. This provides us with 50 milliseconds per frame to sample and process data. The ADC runs at 10 KHz, and thus the design team will have 500 samples in a 50 millisecond interval. The design team will have to process 512 samples in 1 frame time. This is completely doable since the most processing throughput intensive calculation is the FFT, and FFT processing will only take 3.7 milliseconds out of the 50 milliseconds available.

6.2 Embedded Software Design

The embedded software controls the microcontroller and ensures that it can fulfill its responsibilities, including those outlined in section 3.4. Primarily, the microcontroller needs to be able to analyze the inputted audio signal using a Fast Fourier Transform (FFT), determine the behavior of the LEDs and water pumps, and send out control signals to those components, all while interpreting and responding to wireless signals sent from the application.

In order to accomplish all of this, the development team decided to split each 100 millisecond period of time into two 50 ms blocks; the microcontroller would spend one 50 ms block collecting data, then one 50 ms block processing data, then 50 ms collecting data again, etc. Any spare clock cycles would be used to check for and process user input. In order to get the timing correct, the processor checks the clock often and utilizes interrupts. The following section will discuss the inner workings of the embedded software in more detail.

6.2.1 Embedded Code Details

The embedded software follows a pretty basic flow pattern, which is outlined in Figure 6.2.1 below. In Figure 6.2.1, the green boxes represent input and output devices, blue circles are internal functions (or groups of functions), and blue triangles are interrupts.

When the microcontroller is powered on, it first runs through an initialization procedure, then gets stuck in an infinite loop that continually checks for user input from the application. Every 50 ms, there is an interrupt that alternates between two courses of action: collecting data and processing data. When collecting data, a flag is set that enables the second interrupt to read information in from the analog to digital converter (ADC) and store it in a buffer. Since the development team wanted to sample ADC data at a rate of 10 kHz, this interrupt is set to occur every 100 μs. The microcontroller implements a 512-point FFT, so 512 reads are needed to fill the buffer, which will take just over 51 ms. The next time the 50 ms interrupt occurs, the processor turns its attention to processing data. There is a flag that stalls the data processing until the data buffer is full, since data collection will take about 1 extra millisecond.

Once the data collection is finished, the processor takes the full buffer and performs calculations on it. This consists of performing the FFT, removing noise, binning the frequencies, adjusting water pump levels, and calculating the beat power. This is also when the microcontroller will send out control signals to the LEDs and, through the digital to analog converter (DAC), to the water pump array. After control signals have been sent out, the software continues checking for user input until the 50 ms interrupt occurs again and sets the flag to begin collecting data from the ADC once more.

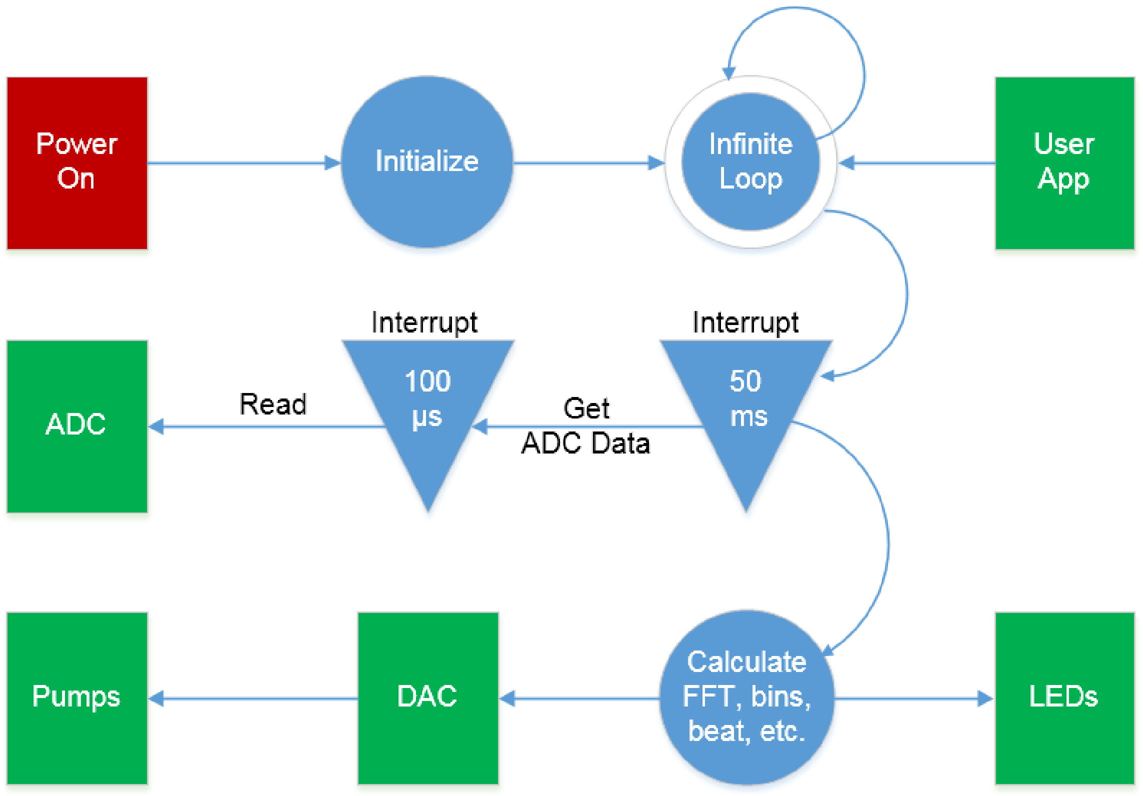


Figure 6.2.1: Overview of the Embedded Software

User input signals come in the form of single characters. A list of the commands that can be used to interact with the microcontroller is available in section 6.6. When the user sends a command from the application, it will be received by the Bluetooth module connected to the PCB's RS-232 port. The character is then compared to the acceptable commands in a large switch statement, and if there is a match, the microcontroller modifies the relevant settings, and the behavior of the system will update the next time the processor gets around to sending out control signals.

6.3 Application Software Design

The team decided to develop a mobile application to accompany the spectrum analyzer. A mobile application creates a pleasant user experience by allowing the user to control the display remotely with an easy-to-use graphical interface directly from their smartphone. Figure 6.2 shows an overview of the application.

The Home Screen welcomes the user when they open the application. From there, the user can establish a connection to the Bluetooth module on the

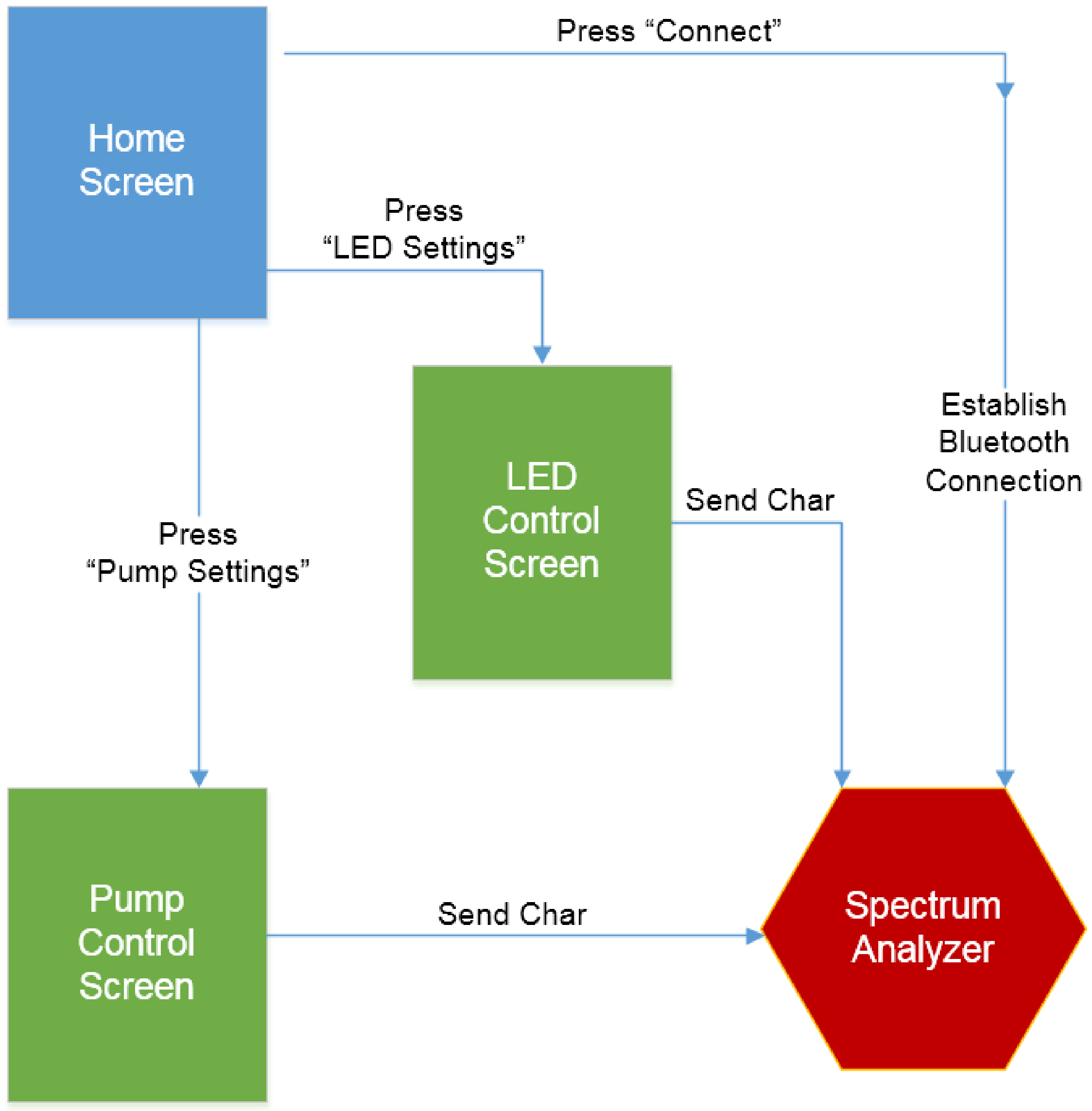


Figure 6.3: An overview of the application and its functionality

spectrum analyzer's microcontroller. The user can then enter the LED and pump control screens, where they can easily send signals to modify the behavior of the display. Section 6.3.2 goes into more detail about the inner workings of the application.

6.3.1 Chosen Platform

After much consideration, the team decided that Android was the platform that was best suited to host the application (see section 3.5.1 for an overview of the factors involved in making this decision). Choosing Android meant that work could be begin immediately, as the developers already had access to an Android device.

To assist with development, the developers decided to use the Android Studio Integrated Development Environment. Based on the IntelliJ IDEA IDE by JetBrains, Android Studio was developed by Google specifically for the development of Android applications and contains many attractive features, including: built-in lint tools to help identify potential problems pertaining to performance, usability, and version compatibility, as well as offering solutions to those problems; several templates for common Android design schemes to help speed up the design process; and refactoring tools to help clean up and optimize code. Android Studio is especially useful for creating user interfaces; it allows users to drag-and-drop different UI components while all of the required code is automatically generated.

6.3.2 Application Code Details

There were a few different goals that the development team wanted to achieve with the application (as outlined in section 3.4), but its most important function is the ability to communicate wirelessly with the spectrum analyzer. A Bluetooth adapter is installed on the analyzer PCB, so all the application has to do is utilize the mobile device's built-in Bluetooth module to establish a connection and send control signals, which are represented by single characters.

Figure 6.3.2 shows a class diagram of the Android application. The Main, Lights, and Pumps Activities provide the user interface for the application, while the Singleton class handles all of the Bluetooth connectivity. When the user launches the application, the Main Activity starts, which automatically calls the Init method of the Singleton class. Init checks to see if the spectrum analyzer's Bluetooth adapter has been paired with, and if so, it prepares to connect. The user can then click the "Connect" button, which calls the Connect method on the Singleton, to open an insecure RFCOMM socket connection. If successful, the app will display a green "Connected!" message, and is ready to send commands.

The controls are organized by function; commands that change the LED settings are in the Lights Activity while commands that adjust the water pumps are in the Pumps Activity. There are buttons on the Main Activity screen to enter each of the other activities (see Figure 6.3.3-1). Once inside, the relevant settings can be adjusted. For persistent effects, radio buttons were used to emphasize that only one mode can be active at a time. To initiate a new mode, simply click the radio button next to the desired mode, and then click the "Apply" button. Selecting a radio button changes a char variable to hold the character that corresponds to that option. When "Apply" is clicked, the current activity calls the Send method of the Singleton class, and the character is sent through the Bluetooth adapters to the microcontroller, where it is processed. There are also calibration options that increment or decrement values; these are accessed by regular buttons.

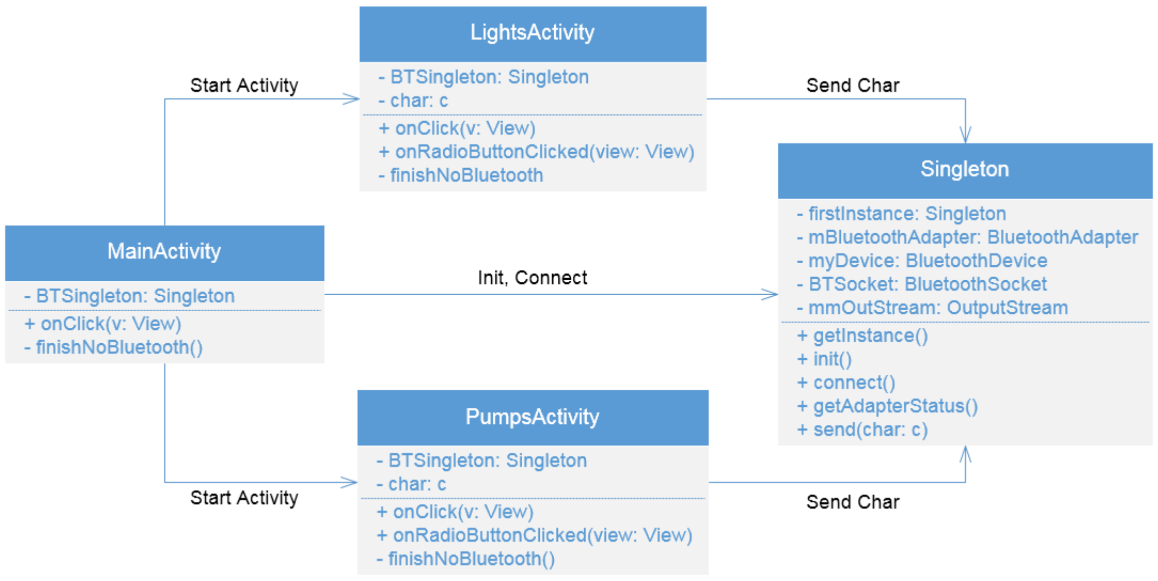


Figure 6.3.2: Class Diagram of the Android Application

The development team decided to make the Bluetooth handler class a singleton class because the application utilizes multiple activities. The singleton model only allows one object of the singleton class to be instantiated in the entire application, so each time an activity requests a reference to it, it's guaranteed to return the same reference ID. This means that all three activities can share one connection without having to disconnect and reconnect every time there's an activity change.

6.3.3 User Interface

In designing the Android application, perhaps the most important feature with which the developers were concerned was how easy it would be to use, and one of the most important approaches to making an application easy to use is by designing a user-friendly interface. The developers did not want the users to have to read a manual to learn how to use the application; they wanted it to be intuitive enough for a user to be able to jump right in, without any prior knowledge, and be able to take full control of all of the application features.

One important detail to keep in mind when developing for Android is that it is supported by a wide variety of devices. Many of these devices use screens with different sizes and aspect ratios, so the appearance of an application might differ slightly amongst them. Fortunately, this is not usually a big deal; the vast majority of Android devices use screens with an aspect ratio very close to 16:9, so the application is designed with that in mind. The user interface is also kept simple enough so that if the app has to be scaled for a different screen size or a slightly different aspect ratio, readability and usability will not be severely impacted.

The application consists of three screens: a main screen, an LED control screen, and a pump control screen. When the application is launched, the user is greeted with the main screen, which contains three buttons: two buttons to start the lights and pumps activities, and “Connect”. There is also a string near the bottom of the home screen that, by default, reads “Not Connected” in red font. The “Connect” button is used to establish a Bluetooth connection to the spectrum analyzer and changes the string to say “Connected!” in green font. This gives the user a visual confirmation that the analyzer is ready to accept control signals. Figure 6.3.3-1 shows an emulation rendering of the main screen and Figure 6.3.3-2 shows an emulation rendering of the LED control screen.

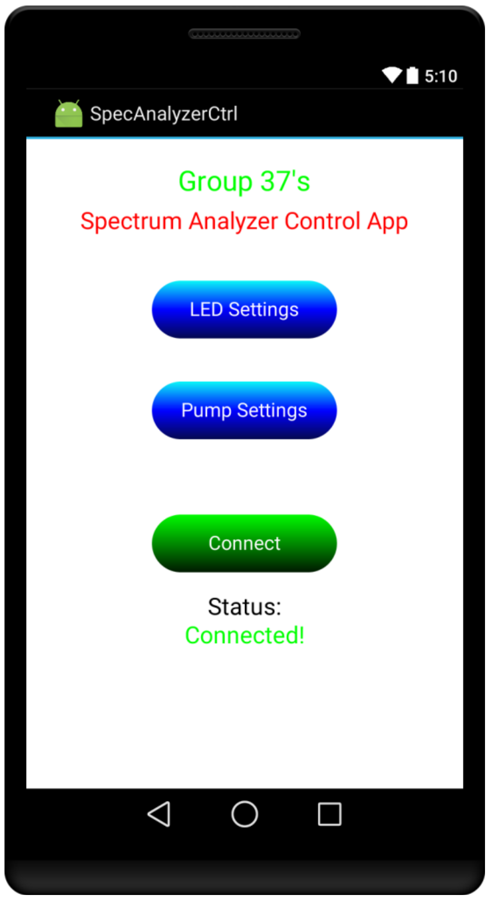


Figure 6.3.3-1: Main Screen

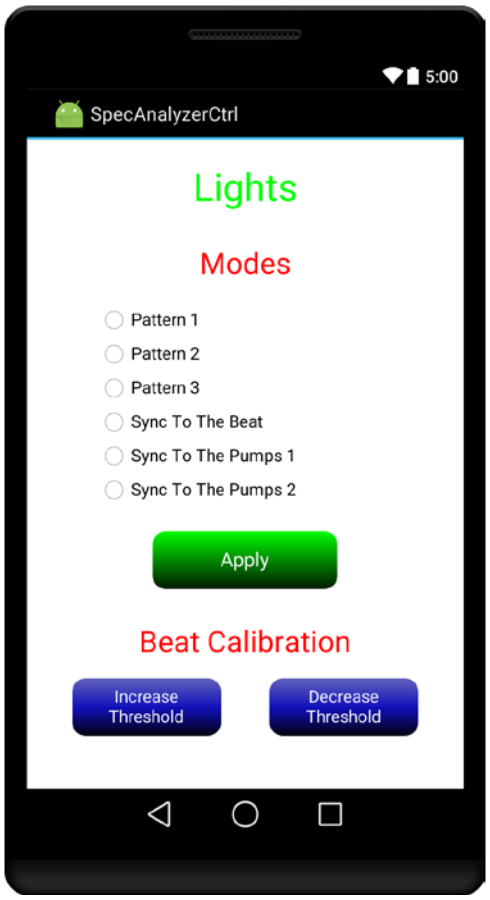


Figure 6.3.3-2: LED Control Screen

The development team decided the application should use bold colors and big, shiny buttons to achieve the perfect mix of lucidity and style.

6.4 Wireless Communication

The most important capability that needs to be implemented wirelessly is the ability to control the spectrum analyzer from the Android application, which means sending control signals from a distance of up to 15 feet away (with a clear line-of-sight). It was decided that this was best accommodated through the use of Bluetooth (see section 3.4.3.5 for the discussion behind this decision). Most modern smartphones include built-in Bluetooth modules, so the application uses that to communicate with the analyzer.

One of the stretch goals of the project involved sending the music to the spectrum analyzer wirelessly, instead of inputting it through the attached auxiliary port. This would provide a few more challenges to the team. First, the music signal would be arriving in a digital format, and would need to be fed through a digital-to-analog converter before being processed. It would also be more prone to corruption and errors, since wireless communication is inherently less reliable than wired communication. It would also put more load on the Bluetooth modules and drain the application user's cell phone battery.

6.5 Wired Communication

During testing, the developers discovered that the Bluetooth module in use on the spectrum analyzer PCB was not as consistent as they had hoped. There were times when the module would be slow to respond or even completely unresponsive. This was concerning as the lack of physical buttons on the display meant that using the RS-232 port on the low-power PCB was the only way a user could communicate with the microcontroller. The development team did not like the idea of having a single method of communication that was so unreliable, so it was decided that an alternative should be developed and tested.

In the event that the Bluetooth module on the microcontroller stops responding, the module can be removed from the RS-232 port and replaced with the male end of a DE-9 connector. The other end of the cable can then be plugged into a personal computer, where a serial terminal program (such as RealTerm) can be used to control the spectrum analyzer. The analyzer is also capable of communicating back to the terminal, to give the user information such as which commands are acceptable, the values of various calibration variables, and debug information.

6.6 List of Commands

Commands to the microcontroller each take the form of a single character. The following sections outline a list of the characters that can be received and interpreted, the command associated with that character, and a brief description of what changes that command will initiate. Alphabetical characters are not case-sensitive.

6.6.1 LED Settings

There are several different LED modes that can be set, including three pre-programmed flashing patterns, one mode that will sync the LEDs with the beat of the music, and two modes that will associate the LEDs with pump activity. The possible LED options are:

- Q – Pattern 1 – Sets the LEDs to flash in a checkerboard pattern.
- W – Pattern 2 – Sets the LEDs to flash in an up and down pattern.
- E – Pattern 3 – Sets the LEDs to flash in a left and right pattern.
- R – Sync To The Beat – Attempts to match the LEDs to the beat of the music. A "beat" occurs when there is a high amplitude signal relative to a moving average amplitude taken across the previous second.
- T – Sync To The Pumps 1 – Associates LED activity with pump activity.
- Y – Sync To The Pumps 2 – Associates LED activity with pump activity.
- F – Increment LED Beat Threshold – Increases the threshold required to register a "beat", making the LEDs flash less often. Only applies when Sync To The Beat mode is active.
- V – Decrement LED Beat Threshold – Decreases the threshold required to register a "beat", making the LEDs flash more often. Only applies when Sync To The Beat mode is active.

There are six LEDs for each pump: two red, two green, and two blue. The "Sync To The Pumps 1" mode contains three different thresholds. The microcontroller compares each pump's current output strength to these thresholds. If it is higher than the highest threshold, all six LEDs corresponding to that pump will light up. If it is only above the second highest threshold, the two green and two blue LEDs will light up. If it is only higher than the first threshold, only the two blue LEDs will be activated. If it is lower than the first threshold, none of the LEDs for that pump will turn on. In contrast, the "Sync To The Pumps 2" mode only contains one threshold, and when a pump's output surpasses that threshold, all six LEDs above that pump are activated.

6.6.2 Pump Settings

The pump drivers can also be controlled by the user. There are four different persistent settings, as well as several calibration tools. The possible pump commands are as follows:

- 3 – Normal Mode – Sets the pump output levels to the results of the FFT.
- 4 – Up Ramp Mode – Causes the pump outputs to scale linearly through the array, creating the appearance of a ramp.
- 5 – Down Ramp Mode – The same as Up Ramp Mode except backwards.
- 6 – Constant Mode – Sets all of the pump outputs to the same constant value.
- D – Increment Overall Gain – Increases the overall gain, which increases the height of the pump outputs.
- C – Decrement Overall Gain – Decreases the overall gain, which decreases the height of the pump outputs.
- A – Increment Pump Constant Value – Increases the constant height. Only applies when “Constant Mode” is active.
- Z – Decrement Pump Constant Value – Decreases the constant height. Only applies when “Constant Mode” is active.
- O – Linear Bin Spacing Mode – Causes the frequency bins to be spaced linearly, from 0 Hz to 4 kHz, with a 250 Hz band to each bin.
- P – Non-Linear Bin Spacing Mode – Causes the frequency bins to be spaced non-linearly, with more of a logarithmic distribution.
- S – Enable Calibration Offset – Prevents the pump output levels from falling below a minimum active value.
- X – Disable Calibration Offset – Allows pumps to turn off completely.
- 7 – Enable Pump Drivers – Enables all pumps.
- 8 – Disable Pump Drivers – Disables all pumps.

The overall gain is a multiplier attached to the output of the water pumps so adjusting it will cause all of the pumps to scale their levels in all modes. The developers recommend using non-linear bin spacing mode when playing music on the analyzer, and only using linear bin spacing mode for testing and calibration.

6.6.3 Terminal Settings

The use of a terminal proved to be extremely useful for debugging, so several terminal options were added to help with that endeavor. They are:

- 1 – Print ADC/FFT Results – Sends all of the data gathered from the ADC and the results of the FFT to the terminal.
- 2 – Print Internal Debug Info – Sends some debug information to the terminal, such as the longest time spent in an interrupt.
- 9 – Print Debug Menu – Sends some possible commands to the terminal that might be useful for debugging.
- J – Continuously Print Uncalibrated Water Pump Levels – Constantly streams the uncalibrated pump output values to the terminal.
- K – Continuously Print Calibrated Water Pump Levels – Constantly streams the calibrated pump output values to the terminal.
- H – Continuously Print Beat Data – Constantly streams the beat calculations to the terminal.
- G – Continuously Print ADC Data – Constantly streams the incoming ADC data to the terminal.
- L – Disable Continuous Printing – Stops any continuous streaming.
- = – Enable Terminal Outputs – Enables normal terminal outputs.
- – – Disable Terminal Outputs – Disables normal terminal outputs.

It should be noted that the terminal settings commands cannot be sent from the accompanying application as the application does not have a terminal interface on which to display information being fed to it from the microcontroller.

7.0 Project Prototype Coding

Before serious development began on a final product, the team considered it worthwhile to construct a series of software prototypes. There are several advantages to developing prototypes, but the main reason was to ensure that a functional communication link could be built between the application and the microcontroller. The team wanted to be sure that all of the components in use were compatible with each other before investing too much time traveling down a road that would turn out to be a dead end.

7.1 Microprocessor Coding Plan

To program the PIC32 microcontroller, the development team decided to use an IDE called MPLAB. MPLAB was developed by Microchip, the same company that designs and manufactures the PIC processor line, so using their IDE seemed like it would lead to a more efficient programming experience. The team also installed the Harmony plugin for MPLAB, which contained a suite of libraries, drivers, and system services designed to assist in firmware development.

Due primarily to the team's lack of experience with the PIC32, developing the embedded software proved to be quite a challenge. The embedded software did not go through a few major revisions like the application software; instead, it was developed a little at a time, with a lot of testing and debugging built into each step of the process. Sometimes, especially near the beginning, the process felt more like trial-and-error than anything else. And whenever the team was working on the code, a copy of the PIC32 User Manual was always within reach, and it was referenced often.

The team started by drawing a program flowchart that roughly resembled the one shown in Figure 6.2.1. While learning the PIC32, bits and pieces of the flowchart were converted to pseudocode and eventually to C code. The team started small, by programming an LED to flash on the microcontroller. This led to using an interrupt to time the LED flashes. Slowly, more functionality was added as the team learned to interface with the various components. First they enabled UART communication, and then added a couple of output DACs, then an input ADC. Then the FFT was implemented and tested. The spectrum analyzer was beginning to take shape.

The team then acquired a few test LEDs. They were wired up and a few basic patterns were applied to see what looked good. Three of those patterns made it into the final options list. Work continued with the addition of frequency binning and amplitude scaling. The water pumps that the team had ordered finally arrived and basic patterns were coded in to test them, including the constant mode and the ramp modes. Over the course of several months, the team continued adding more functionality, fixing bugs, and tweaking calibration values. Because of inconsistencies in pump performance, each water pump in the array had to be individually calibrated with its own internal gain multiplier.

7.2 Android Application Coding Plan

Programming the Android application was a process that began with installing an IDE that was developed by Google specifically for Android development called Android Studio. Android Studio comes with an SDK Manager, which can be used to download various SDK tools. Several of the SDKs were installed for this project. The Android 5.1.1 Platform was installed, along with Google APIs for an x86 system image. Android 5.1.1 Lollipop was chosen as the target platform because that is the version of Android that is installed on the developer's phone, although the application should be backwards compatible with Android platforms as old as Android 2.3.3.

Also installed was a tool called HAXM (Hardware Accelerated Execution Manager), which would help speed up the emulator that would be used for debugging. The emulator provides a fast and efficient method of testing and debugging most aspects of most applications. Unfortunately, it does not support Bluetooth, so a physical Android device was still required to test most of the functionality associated with this project. For this, the developers used a Motorola Moto G3 running Android 5.1.1.

Early versions of the application were quite unimpressive. The first version attempted to cram all of the available options onto one screen, making the app feel messy and crowded. The development team decided to use three activities so that the controls could be arranged by function: commands that changed the LED settings went in the Lights Activity while commands that adjusted the water pumps went in the Pumps Activity. This second version of the app felt much more organized.

Utilizing multiple activities introduced a new problem, however. Since a BluetoothSocket object is not serializable or parcelable, passing a socket reference between activities was proving difficult. The development team realized that a string could be passed between activities relatively easily, so some genius came up with the idea of passing the address of the remote Bluetooth device as a string. This meant that each time a new activity was started, it would have to use the address to establish a new connection to the microcontroller's Bluetooth adapter. Disconnection functions were added to activity closing procedures to smooth the transition. This third version of the app was much slower, but it worked.

After a few days of fiddling with this frustratingly slow version, the development team decided that there simply must be a better way. After doing some research, the team discovered the concept of a singleton class. A singleton class only allows the instantiation of a single object of that class across the entire application; any activity can request a reference to the object and they are all guaranteed to receive the same reference ID as the other activities. So the team created a new singleton class to handle all of the Bluetooth interactions. The Main Activity automatically instantiates the singleton object when the app is opened, and pressing the "Connect" button will open a socket within that object. When a different activity is opened, it requests a reference to the singleton which it can use to call the send method to transmit characters. The singleton model allows all three activities to share a single connection, which makes a lot more sense than disconnecting and reconnecting each time the screen changes. This was the fourth version of the app and it worked much more quickly and reliably than the third version. This is basically the same version that the team is using today; a few cosmetic changes were made but the underlying architecture remains the same.

8.0 Project Prototype Testing

Assembly began in the spring semester when Senior Design 2 started. Since the final project includes a variety of apparatuses, it's important to see if each component operates as anticipated in the drawn out plans, and also whether they function as a single system. It is also important to document each testing phase along the assembly of the display so that if there are any errors or faulty components, it would be easy to go back and repair any complications that might slow down the group's progress.

8.1 Hardware Tests

Table 8.1 shows the hardware testing development plan.

Hardware	
2/27/16	Power Supply
2/27/16	Microcontroller
2/27/16	Kill Switch
3/5/16	Water Pump
3/12/16	LEDs
3/26/16	Speakers
4/16/16	Casing and Structures

Table 8.1: Testing schedule

The hardware testing will consist of the power supply, microcontroller development board, water pumps, LEDs, kill switch, speakers, electrical components and casing, along with a laptop computer and a smartphone. With the basic equipment, designs for prototyping can be done as well as small scale testing to ensure that the most essential piece of the display, which are the water pumps, works as intended and reach the desired height. Once this is in order, the group can continue to put together the rest of the project and add finishing touches to the final project.

The testing environment will be conducted within an optimum humidity and temperature setting and a well-lit area, most of which will be indoors. The structure should, however, be able to withstand a little bit more than just room temperature, which should be between 60 to 90 degrees, with a humidity of roughly 50%.

Power supply testing can be done at the senior design laboratory at UCF. The equipment needed are a multimeter, function generator, and oscilloscope. The multimeter is necessary to check the voltage and current ratings, which the function generator and oscilloscope will be used to create and analyze signals.

The Magicfly DC30A-1230 pump was tested out and is able to shoot water at a maximum height of 13 inches with a 12 volt power supply and scaled linearly. An inch of tubing was added to make the pump above the water, however it reduced the height of the water drastically. When a straw was added to narrow the flow, the height increased to 18 inches. The team decided on setting the height of the display to about 16 inches so that the water can reach the roof. The line of the pumps must be aligned flawlessly so that the stream of water is vertical and does not throw off any balance. The final water pump tests, before the assembly of the structure, should be done either indoors in a bathtub or outdoors since it can get messy with splashes.

The interaction between the PIC32 microprocessor, STP08CP05 device, and water pumps will need to be tested. This involves establishing control pins and observing the communication that occurs among the devices. For a small-scale prototype test, the microcontroller can be connected to the shift registers which can be attached to the LEDs. An oscilloscope can be used to note the speed at which the data is transferring.

The next testing phase will confirm that the LEDs work as planned. This can take place by first making sure that the testing takes place in a well-lit area. Then, the drivers need to be verified to ensure current control, functionality, color, and brightness of each of the LEDs, confirm that they pulse to the audio signals that are received, and that they are capable of being turned on and off. Another test that can be done for the LED array is to see the variation in flashing speed and determining whether or not the flashing rate would need to be sped up or slowed down. Once this is done, the group needs to check for any defective or burned out LEDs so that the need to replace them when it is too late can be avoided.

The easiest assessment during the hardware tests contains the kill switch and speakers. All that is needed to be checked is the ability of the switch to shut down the display when necessary and the sound output of the speakers, which can be done by connecting a music-playing device to the audio jack.

The last thing to test is the casing and structure. The final display must be stable and light enough to be portable. To guarantee stability, the group needs to notice any unwarranted vibrations and tremors due to the audio output and jets of water that may affect the assembly. To test this, the display must be placed on a perfectly leveled surface and turned on to ensure firmness.

If the tests are successful, the group can begin soldering the electrical components and constructing the larger, final scale of the product.

8.2 Software Tests

Table 8.2 shows the software development plan for the project

Software	
2/27/16	C Code for Processor
4/9/16	Java Code for Mobile Application
4/9/16	Bluetooth Capability

Table 8.2: Testing

The software is one of the key parts of this project and each piece of software should be thoroughly executed on a simulator test bed before they are assimilated to the final product.

The first things to test are the hardware portions such as the drivers, power supply, filters, processor, converters, and amplifiers by simulating them on Multisim. After that test is completed successfully, the designs can be applied to the actual device.

The next testing phase is the C code that was written for the microprocessor. It needs to be made sure that there are no errors within the code and that each of the 16 divisions are accounted for. A Fast Fourier Transform is also to be performed and once it is implemented, the DACs should run the water pumps accordingly.

Another portion of testing will include seeing if the mobile application, written in Java, works on one of the major smartphone brands, preferably Android. The group will use one of the members' phones to test its ability to communicate with the spectrum analyzer wirelessly by sending control signals, turning the device on and off, and even possibly changing the color of the lighting and pump settings. The software should enable interaction between the PIC32 microcontroller and the Bluetooth module. The performance will be judged based on how accurate and quick the response is and how well it works over Bluetooth.

9.0 Administrative Content

To ensure that the group completes the senior design project successfully, it is important to manage planning, budgets, and workload. Being organized in these aspects will help everyone perform quicker and more efficiently.

9.1 Project Management

The team made sure to keep in contact via GroupMe on their cellular devices and by sharing and uploading information and links to the Senior Design folder on Google Drive. They held meetings at least once a week to ensure the progress of the assignment, discuss further plans and goals, and to meet with Dr. Richie for approvals on each checkpoint of the project. It is vital for group members to keep each other up-to-date on individual progress as well and to remind one another of upcoming meetings and weekly objectives.

9.2 Milestone Discussion

One of the challenges was following a schedule and meeting all deadlines so that they do not fall behind. The milestone table below shows due dates and team goals to be followed for completion of the project, as well as descriptions for some of the components, for both the fall and spring semesters. Table 9.2-1 shows the milestones for fall and Table 9.2-2 shows the milestones for spring.

Date	Detail	Notes
9/3/15	Form Senior Design group	
9/3/15-9/15/15	Research project ideas	
9/12/15	Senior Design Boot Camp	Attended event on campus where the team was given helpful tips on planning, brainstorming, managing time, and funding
9/15/15	Initial Project Details	Initial project document describing idea, objectives, requirements, and block diagrams
9/16/15	Project approval meeting	
9/28/15	Funding Proposal	Filled out a detailed project budget form in order to request funding from Boeing/Leidos
9/23/15-10/12/15	Research	Worked on design notes for processor, power, drivers, converters, and water pumps
10/15/15	Table of Contents	
11/5/15	Began ordering parts	Ordered microchip and microprocessor development board
11/11/15	Display Diagram	Worked on schematics and calculated dimensions of display; researched parts
11/12/15	Rough Draft	
12/10/15	Final Draft	

Table 6.6.3-1: Milestones for Fall 2015

Date	Detail	Notes
1/31/16	Main Hardware	PCBs should be manufactured and obtained.
3/21/16	Water Pumps	Water pump array should be assembled and tested for functionality.
3/21/16	LED Array	LEDs should be able to flash in various patterns, as well as to the music beat.
4/2/16	Bluetooth	Bluetooth connection should be working.
4/9/16	Mobile Application	The application should be done by now. It will be tested on a smartphone to see if it can control the display via Bluetooth communication.
4/10/16	Structure Assembly	The physical assembly of the display should be done by this time. Casing and structure should be tested accordingly.
4/11/16	Testing	Final integration and testing
4/29/16	Final Documentation	Final report, website, and evaluations due.

Table 6.6.3-2: Milestones for Spring 2016

9.3 Budget and Finance Discussion

All purchases for the project will be made online and sent to Dr. Richie's office, from where the team will need to retrieve the materials. A document requesting funding was submitted to Boeing/Leidos for approximately $1000. The full amount was not available and the amount received is a generous $899.46, which will be reimbursed to the members after expenses are made. Although it is unlikely, any costs that go over the funding amount will be divided among the group members evenly. Table 9.3 shows the budgets for project materials.

Item #	Description	Quantity	Total Cost
1	Bluetooth Connection	1	$60
2	Circuit parts A/D, D/A, Microprocessor	-	$454
3	PCB manufacturing	-	$400
4	Water Pumps	16	$176
6	LEDs	96	$30
7	Casing	-	$40
9	Power Supply	1	$65
10	Cables and miscellaneous	-	$85
		Total Estimate	**$1310**

Table 9.3: Estimated Budget

9.4 Labor Distribution

The labor distribution depended on interests and skillset, and the group tried to divide it up as evenly as possible between the four members. Katie and Esha took responsibility for customizing the power supply, while Tim and Katie worked on the microprocessor and schematics of the circuitry, and overall hardware design. Esha and Josh worked on setting up and making sure the LEDs work as planned. Josh and Tim were in charge of the software design and PCBs. Esha and Katie also worked together on designing the display frame, calculating its dimensions, and constructing it. The final report was divided among the team members based on choice of topic and having an equal distribution of page count. Materials for the project have been researched together and all financial decisions were made as a team based on cost, efficiency, and optimization.

9.5 Supported Coursework

Almost all the core courses for the electrical engineering curriculum greatly helped with the implementation of the project. Engineering Analysis and Computation, Computer Organization, and Embedded Systems were required for the knowledge of programming languages such as C and Assembly. Digital Systems taught the basics of logic schematics, Verilog language, and how to use FPGA. Electrical Networks, Networks and Systems, Electronics, are vital for comprehension of circuit and hardware design. Digital Signal Processing, Analog and Digital Communication, and Linear Control Systems were necessary for understanding how to use Fast Fourier Transforms, binning, and sampling.

9.6 Skills Acquired

As state in preliminary course information, the Senior Design courses are intended to offer the opportunity to apply the engineering skills that were accumulated from the courses taken and to learn concepts in engineering practice. In Senior Design I, lecture topics included a history of engineering education, profession and management, design constraints and standards in the real world, ethical responsibility, and engineering economics. After forming a group, students should work together to form teamwork skills, communication with one another, and the ability to recognize and form an approach to solving engineering problems on a global, economic, environment, and societal scale. By implementing this knowledge to the project, students are able to encounter obligations that they have not been subjected to in previous courses. It allows them to rely on each other to perform critical work to complete team goals.

While these dexterities were learned in Senior Design I, there are many hands-on skills that Group 37 had both practiced and attain in Senior Design II. This includes actually applying techniques learned in labs in the design and construction of the Spectrum Analyzer such as manufacturing a circuit board, soldering, wire wrapping, and properly handling power tools. Additionally, the team worked to compile C language and Java codes to improve programming skills, interconnect a microprocessor to water pumps and light-emitting diodes, and to generate an authentic mobile application to interface with the device. The project also helped in developing time management skills, technical writing abilities, organization, and team building.

10.0 Conclusion

The target for Senior Design I was to complete very extensive research and come up with a fully detailed initial design for the project that the team began constructing in Senior Design II. Throughout the first semester, many design and materials decisions were made and changed as the plans and ideas were adjusted. Making the device customizable allows for more freedom with the design process for the team members, so further modifications can still be made while still sticking with the initial proposal.

By the beginning of Senior Design II, all the designs and schematics have been finalized for the most part and the team began to order materials. A detailed testing procedure was also obtained, and based on the established milestones chart, the group has currently been meeting calendar goals.

The most considerable result of this documentation for the team members was the learning experience. The project spanned a large number of topics in electrical and computer engineering and took the group beyond basic classroom understanding. The final phase of Senior Design was the physical developmental stage during which the team came across a number of challenges. Among them were programming the PIC32, creating the smartphone application, and implementing the Bluetooth due to the fact that the team only had one computer engineering student and three electrical engineering students with little to no experience in the software. Other challenges were structure related. A number of the water pumps malfunctioned and had to be switched out with working ones before they were permanently mounted and wired. The wooden water reservoir had leakage so that was solved by placing rubber pond liner inside it. The acrylic tank also had water spilling from underneath and cracked glue, most likely because of handling and the weight of the LED lid on top. The solution to this was to line the edges of the opening with rubber window sealing and duct tape so that the water falls back inside the reservoir. Super glue was also used instead of the solvent to make the tank sturdier. The last challenge was to make the display aesthetically pleasing. This was done by spray painting all wooden pieces black, and giving the acrylic tank a white background so that the wires are not showing and also to make the LEDs more visible.

Although the amount of labor that had been put into the display was not appealing, the end result was worth it as it shows how far the group as come and how close they are to applying the experience and knowledge into the world of engineering.

10.1 Acknowledgements

Group 37 would like to acknowledge Dr. Samuel Richie for being a teacher, mentor, and advisor all semester and giving the group helpful tips on how to improve the documentation and ideas for the final product. They would also like to recognize Dr. Lei Wei for lecturing on real world engineering concepts, problems, and solutions. Lastly, the team would like to thank Boeing/Leidos for generously funding them, without which it would have been much more difficult, if not impossible, to complete the project.