# Virtual Bench Warmers
# Piano Learning Aid

*Nathaniel Geerdes, Ian Green, Mert Karakas, Eddie Russell*

12/16/2019

**Capstone Design ECE4440/ECE4991**

## Signatures

Nathaniel Geerdes (12/16/2019)

Ian Greene (12/16/2019)

Mert M. Karakas (12/16/2019)

Eddie Russell (12/16/2019)

## Statement of Work

Nate Geerdes: I served as the main software architect for the project, designing most of the flow of the LabVIEW virtual instruments. I developed sub virtual instruments for mapping frequencies to notes on the piano to an array of bytes to send through SPI to the LED strips. I created a method for making songs in an excel spreadsheet to be read through a USB by the MyRIO. I implemented an algorithm for filtering out harmonic frequencies coming in. I lead the team in making code adjustments and improving the speed and quality of our product.

Ian Greene: I spent most of the first few weeks trying out several different methods of pitch detection. This included algorithms developed in the FPGA and in the main processor. Eventually we settled on one and I built a sub-VI around it. After this I helped develop a sub-VI to map a pitch to the nearest key on a keyboard in order to avoid tuning issues. I helped Nate develop whenever I could and assisted testing efforts with an in-browser piano and an electronic keyboard.

Mert Karakas: I was the designer and engineer responsible of our hardware components. I started the project by constructing requirements on how to power our hardware components. After calculating all of the power requirements, I spent the first couple of weeks picking out potential parts and breadboard designs. This allowed me to verify my design with a Virtual Bench without commiting to a PCB order. Then, I started designing a first iteration PCB board. The first iteration board had a lot of test pins and unprofessional connections. For the next iterations, I started to add professional connections like the MyRIO connector header, DC power jack, the LED Strip connector and the MyRIO power connector. The rest of my work included testing and integration with the rest of the system.

Eddie Russell: In the first couple weeks, I helped research possible techniques for frequency analysis, specifically cepstral analysis. For the rest of the semester, I mainly focused on the output of the system; the LEDs and how to interface them with the microprocessor we would end up choosing. I researched various datasheets for LED strips and helped Mert determine the requirements for providing power to the strip. I also mapped the individual LEDs to each piano key and the keys to frequency ranges. I also helped Nate with creating and testing song files as we got near to the demo.

# Table of Contents

## Table of Figures

## Abstract

The Piano Learning Aid gives users a more interactive learning experience when practicing the piano. The device is primarily constructed from a power source, LED strips, a microphone, and a MyRIO. It guides users on the next notes to play with LEDs corresponding to each key on the piano and receives input from the user by listening to the individual frequencies coming from the piano. The device is portable and can be added onto any standard sized piano. Files containing different songs can be uploaded to the device, allowing users to practice a variety of songs.

## Background

We chose this project due to our interests in Digital Signal Processing and in music-related devices. We wanted to create something that would be both fun to develop and fun to test out. Additionally, we wanted our resulting project to have a fun and interactive demo. This project checked all of these boxes and allowed us to work with an instrument we already had access to.

In the past, pianos have been built with light-up keys that teach students to play songs [1]. Portable versions of our proposed device have also been made that can be swapped between different pianos, using sensors to determine when a key has been pressed [2].

Our spin on this was to use a microphone to determine what notes are being played by the player instead of lasers. We believe that, while this increased the cost of our prototype, it opens up the possibility of interesting use cases. Because of this, the device is compatible with any piano and does not need to be built into an electronic instrument. The base pitch detection and feedback methods created in this product can be applied to almost any instrument, from the piano to the human voice. We also believe that, given more development time, this could be cheaper than many products on the market.

This project drew from several previous courses that we have taken. The embedded component builds off of our embedded systems experience in Introduction to Embedded Systems and Advanced Embedded Systems while also integrating the LabVIEW knowledge we picked up in the FUN series. The frequency analysis and digital signal processing aspects include topics touched on during the Fundamentals of Electrical Engineering course series and the Digital Signal Processing elective.

## Constraints

### Design Constraints

We were definitely constrained by the limits of our microcontroller. Pitch detection algorithms are very computationally expensive, and our design had to be accurate while still operating in real time. This meant that our program design had to be efficient and waste no time in its computations.

We were also constrained by the devices we interacted with. Sampling audio using the MyRIO's FPGA and a microphone disabled the MyRIO's SPI library. Because our LED strip took instructions using SPI, we had to create our own SPI functionality.

One of our project goals was to include a PCB that we designed and built ourselves. to do this we had to develop a board to provide power to the MyRIO and LED strip.

Finally, we were constrained by LabVIEW's design paradigms. Certain algorithms are difficult to develop in flow-based languages such as LabVIEW, and the difficulty of passing data from the FPGA to the processor also led to a tradeoff between difficulty of implementation and speed.

### Economic and Cost Constraints

Many of the parts we used did not end up being so expensive that they ate into our budget in any significant way. Our LED strips and microphone were both reasonably priced as were the voltage regulators and fuses that went into our PCB. We developed this project with the belief that creating a working product was the most important goal, and that after this was done we could look into ways to reduce the cost.

The most expensive component by far was the MyRIO, if we had paid for this using our budget we would have been significantly more limited in our part choices.

## External Standards

In our project project, there are two main components that must adhere to safety and professional standards: powering our project and our PCB. We used a wall adapter and placed a DC power jack to our PCB. The wall adapter followed the standards of "NEMA 5-15" and IEC [3]. When it came to our PCB design, the IPC (Association Connecting Electronics Industries) provides a checklist for each stage of the PCB design: Project Start Level, CAD Level, Printing Board Ordering Level, Printed Board Assembly Ordering Level, and for Cleaning and Conformal Coating [4]. When we started our project, the components we chose through Multisim were standardized according to IPC-2221. This standard defines rules for properties like "minimum isolation distances between holes and tracks", "printed board thickness" and other component related specifications.

While, unfortunately, there is no defined standard for the dimensions of a piano keyboard, wikipedia lists a set of average dimensions for a piano octave [5]. The piano also has a standard set of 88 keys, each with their own standard frequency [6]. By combining this information, we were able to design a device that would work the same on any standard keyboard.

Our device makes use of CSV files to store songs in a machine-readable format. The CSV file format is standardized, so songs can be created on any computer [7]. Additionally, these files are stored on a USB flash drive which makes use of the USB standard [8]. Because of this, files can be easily transferred to the device from a variety of machines.

### Tools Employed

All of the software for the MyRIO was developed in LabVIEW [9]. We had very limited experience with LabVIEW going into this project so we had to quickly learn how to implement the processes we envisioned using LabVIEW paradigms. We divided the implementation of the software into multiple sub virtual instruments in LabVIEW. Within LabVIEW we used the High Throughput Mode for the MyRIO in order to quickly analyze the audio signal coming in through the microphone.

Hardware Design was done through Multisim [10] and Ultiboard [11]. These tools worked together seamlessly since for each component we created in Multisim, we could easily attach it to a footprint that we designed using the datasheets of the components. Fortunately we had experience with Multisim and Ultiboard from our ECE classes. However, we learned how to write simulation scripts for our new components with Multisim and create professional-grade components that referenced datasheets of our components with Ultiboard. Although scripting component simulations in Multisim was difficult, we were able to work with existing components and modifying them to work with our new components. With Ultiboard, we were able to use pre-defined templates that ensured professional-grade connections with the MyRIO and automate a BOM of our design.

## Ethical, Social, and Economic Concerns

### Environmental Impact
No part of our project seems to us to be more environmentally damaging than the average piece of hardware. While the plastics and chips used in the device are probably made using a variety of materials that are not environmentally friendly, we do not believe that any are much worse than normal. The power consumption of our device is minimal but is certainly larger than the power taken to play and electric piano on its own or the complete lack of power taken to play and upright or grand piano.

### Sustainability
By using LED strips in our project, we increased the sustainability of the system since LEDs require less power consumption than other forms of feedback lighting and are not made with any environmentally damaging chemicals [12]. However, since our LED strips were not custom made, not all the LEDs that were produced for the project were being used.

### Health and Safety
Our project was designed to be easy on the eyes for users. We did not want to hurt users' eyes with bright LEDs or have the lights flash quickly enough to raise epilepsy concerns. We set the brightness of the LEDs to a comfortable level and used methods that ensured as few LEDs as possible would be turned on at any one time. We also had to modify the PCB once we realized that our voltage regulator was quickly overheating and creating a safety hazard.

### Manufacturability
Much of our project was created with multipurpose devices that, while great for prototyping, would not be cost effective to use in manufacturing. Additionally, we never developed a plastic housing for our device which would be essential for a marketable product.

To make this device more manufacturable, we would have to move from a hobby LED strip to a custom made LED strip that only had LEDs placed at the location of keys on a keyboard. This would also allow us to move from two narrow LED strips to one wide LED strip. We would also have to replace the MyRIO with customized hardware such as an audio sampler and a DSP chip that could perform dominant frequency analysis. After doing this, we could use a simple microcontroller to interface with external memory containing a song and with the LED strip. Finally, we could save costs by buying a much cheaper microphone that is just good enough to discern the frequencies it picks up.

### Ethical Issues
While this device is intended to supplement the instruction given by a human piano teacher, there is a chance that this device will be used as a substitute for a music teacher and hurt music

teachers' business. There is also a chance that this product may cause students to become too dependent on the device itself and practice following along with the device, not practice playing what they see in front of them. By learning from a computational system instead of another human, a user may pick up bad piano practices that may inhibit their ability to advance that a human teacher would have corrected them on.

### Intellectual Property Issues

The first claim we investigated is very similar to ours in that is uses LED lights to provide guidance to the user regarding notes to play [13]. A key difference is that their primary claim relies on the use of MIDI signals to recognize which notes the user is playing. Our device instead uses data obtained with a microphone.

The second patent is very similar in operation to our device. It picks up musical data from an instrument using a microphone, compares that to the expected sound, and adjusts a feedback system accordingly [14]. Interestingly though, this patent is specifically created for a guitar or a similar instrument with strings, a fingerboard, and a body. The main claim explicitly mentions this and thus does not cover our device.

The third patent utilizes light-up keys within the piano itself to teach the user to play [15]. Not only does this not use a microphone, it is combined with the piano itself while our device is seperate from the piano.

In light of these patents, we believe that our device is patentable. It is obvious that the use of LEDs to indicate key presses is used in many different patents and is thus not novel. What we believe is novel, however, is the use of a microphone to analyze a user's play and indicate whether they are playing a pre-recorded song correctly.

## Detailed Technical Description of Project

Our project is a device that teaches users how to play songs on the piano. It is constructed using a MyRIO, LED strips, a microphone, and a PCB. Songs to be played are uploaded to the device by plugging a USB storage device into the MyRIO's USB port. The MyRIO lights up LEDs on the strips which are layed on top of a piano's keyboard. These LEDs show the user which notes that should play next in a song. Green LEDs show them the current notes they should be playing while Blue LEDs show them the next notes that they will eventually play. A microphone placed near the piano picks up the sound of notes being played and analyses the frequency content to determine whether the correct notes are being played. If they are, the LEDs that were previously blue turn green and a new set of blue LEDs light up. In this way, students are guided through the piece and through repetitive practice, can learn the song.

The device begins by reading in a CSV file from the flash drive plugged into the MyRIO. The CSV file stores all of the notes that are contained in the song. The file begins with an initialization row to indicate where in the file to begin collecting for notes. Each row after contains a set of numbers corresponding to the keys on the piano that should be pressed at that stage. For example: if a row contains the numbers 33 and 54, the 33rd and 54th keys on the piano should be pressed which correspond to F3 and D5. When the CSV has been entirely read in and there are no more rows to read from, the song is over.

The process begins with the microphone, which is constantly taking in audio. This audio is sampled by the FPGA and both channels are added together to create a 1D array of doubles. We then use the Extract Multiple Tone Information to obtain an array of all frequency components that exceed an adjustable input threshold [16]. We configured the Audio Input N Samples VI to have a sampling rate of about 44khz and to output 5000 samples at a time. This meant we could sample all frequencies in the human hearing range and, with 5000 samples, frequency domain analysis was accurate enough that two adjacent notes on the piano could be individually identified.

Once we had an array containing the dominant frequencies found in the incoming audio, we remove any tones that are not between 26Hz and 4300Hz (Which are just above and below the highest and lowest frequencies on a piano). Once this is done we pass the list of frequencies into a VI that find the keys on a piano with frequencies closest to the input frequencies. We do this in order to avoid issues with piano tuning; there are a whole range of frequencies that an out-of-tune piano could make even if the player is striking the correct key. Further, the acceptable range changes for each note: a low C may only have a few Hertz of wiggle room but a high E could be out of tune by almost 50Hz.

When we identified the piano keys we are picking up, we compared those keys to the song stored as a CSV. In the simple version of our program, as long as the microphone is picking up at least the frequencies we expect the song will move on. The overall virtual instrument including program flow for our simple mode can be seen in the figure below. In complex mode, the microphone must be picking up only the notes we expect and no more. If there are extra notes played, those notes are output as an array of incorrect notes and light up as red on the keyboard, indicating to the player that they must correct their playing. From this VI we receive an array of keys that should be lit up green and keys that should be lit up blue. The complex mode also goes through a software algorithm that filters out unexpected harmonics that register as another note on the piano. This is done to ensure that only the correct notes are not being played despite also possibly picking up additional harmonic frequencies. The complex mode virtual instrument control flow can be found in the appendix.

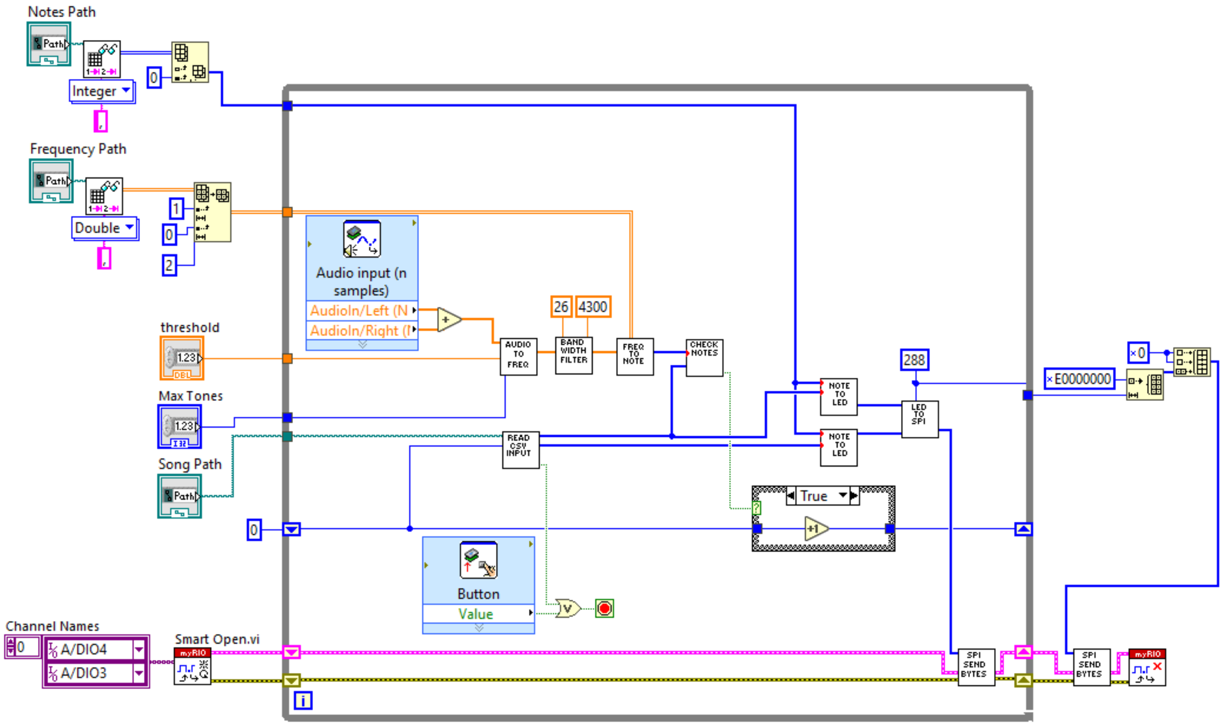


**Figure 1 Simple Mode Program Flow**

The LEDs are lit up by our own implementation of SPI. This implementation bit bangs values to a configured clock pin and data pin on the MyRIO. One VI uses a CSV containing the mapping we created between piano keys and LEDs on our strip to take in the keys we want to light up and output an array of bytes that contains the data for color and brightness of each LED in the strip. We then send this information with the designed SPI module through the pins.

Button0 on the MyRIO is used as a reset for our program. Pressing the button in the middle of play will exit the current song and turn off the LED strip. During this time, the user may swap out USB drives for one containing a different song if they wish to do so. Pressing the button again will bring up the start of current song on the USB. This allows a user to swap songs without turning off the device or to restart a song if they wish to.

The national instruments MyRIO embedded device forms the backbone of our project [17]. We selected it due to its computing speed, its support for SPI and audio input, and its built-in FPGA. on top of this, it allowed us to use LabVIEW and the wide array of VIs available for use in it.

For the LEDs, we used two LED strips ordered from adafruit [18]. We used these strips because of the ease with which we could light up only the LEDs we wanted to use. We also used them because we needed each LED address to produce many different colors. Finally, we used this strip because it had a high LED density. With the fixed spacing of keys on a keyboard, we needed at least one LED to be positioned above each key.

For the microphone, we used a Lavalier lapel microphone ordered from Amazon. The microphone works very well and plugs into the auxiliary input jack on the MyRIO. on top of this, it has an adapter that reduces its three channel plug to two so it can interface properly.

Velcro strips were used to keep the device on our piano. While they were intended to stick well to our device's components but to be easily removed, the plastic that housed to LED strip did not stick well to the velcro and in the future a better solution should be found.

A Yamaha electric piano was used for testing our project. While not necessarily part of our project, it was essential for development. We chose this piano because it was readily available in the NI lab and because it was short enough that we could use one pair of LED strips to cover the entire keyboard's length.

A custom PCB that was designed to power both the MyRIO and the LED strips. The input to the system was a 12 Volt AC/DC wall adapter. The input was converted to 5 Volts output for the LED strip and 6 Volts for the MyRIO.

One key tradeoff involved the use of MyRIO. We were all uncomfortable with our LabVIEW experience going into this project and wanted to avoid using the MyRIO. It quickly became apparent, however, that the MSP430 would be far too slow to run our program and that developing pitch detection algorithms in C would be more trouble than it was worth. In light of this, we chose to learn as much LabVIEW as possible in order to use its impressive array of signal processing tools.

An overarching design philosophy for this project was that "an expensive working prototype is better than a cheap dud." While our LED strips, microphone, and processing unit had more functionality than necessary, it meant that we could confidently say that this product was feasible and that all we would need to do is cut costs.

Another key trade off involved the speed of our program. If the program ran too slowly on the device, users would become bored while playing with it and would abandon it too soon. If we cut too many features in order to create a fast program, the end result might also be too rudimentary or buggy to be used. For this reason, we cut harmonic filtering and incorrect note feedback from the final product that we demoed at the capstone presentation. While it did work and would definitely be a benefit to users, the associated slowdown compromised the usability of the product.

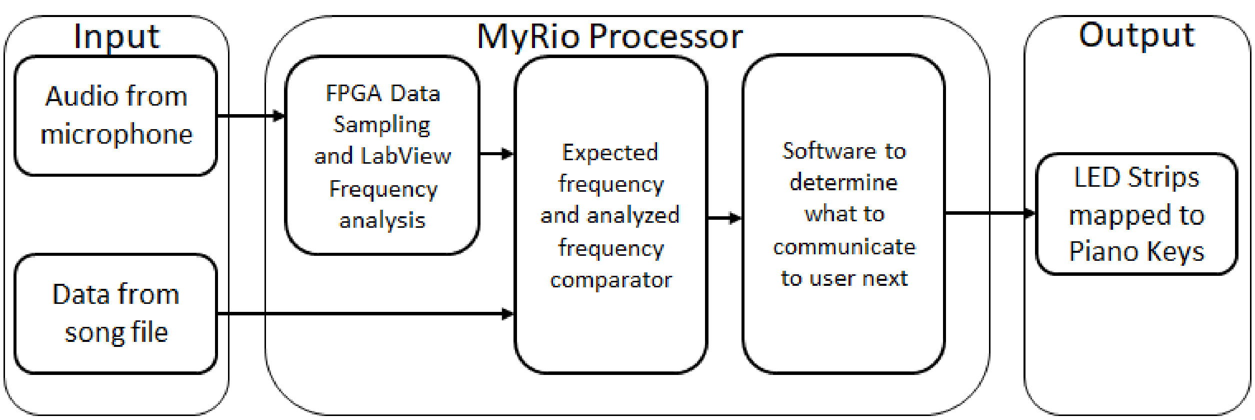


**Figure 2 Control Flow Block Diagram**

Figure 3 shows the system schematic of our PCB design. The flow of the schematic is from left to right. Component U4 is a 2.1mm DC Power Jack that connects to our wall adapter. Then, the input is separated into two nets: the top is to power the MyRIO and the bottom is to power the LED strip. Both nets are first protected by a fuse calculated according to different power requirements for each device (U2: MyRIO fuse holder, U6: LED strip fuse holder). The maximum current the MyRIO can draw is around 1 A and the LED strip can draw upto 4 A. Therefore, we used a 1.25 A fuse for U2 and a 5 A fuse for U6 (fuses are rated lower than their actual values).

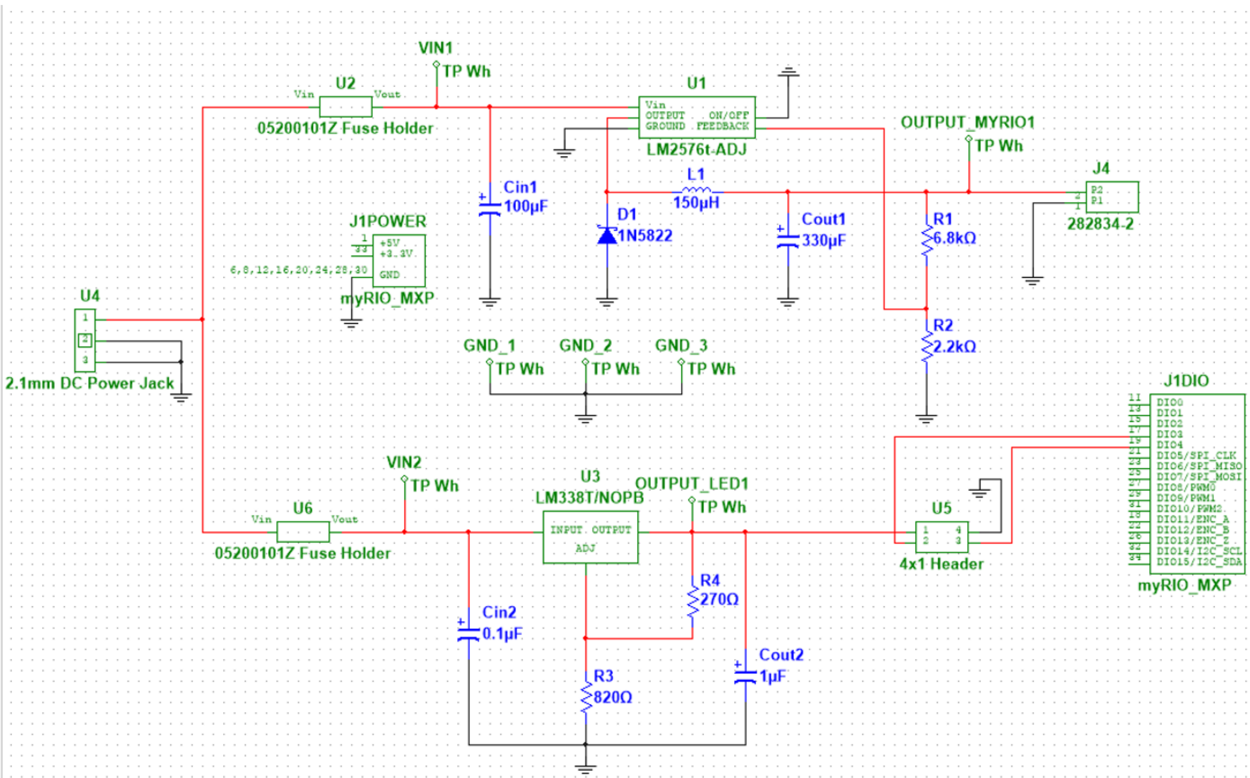


**Figure 3 System Schematic Design**

Then, the next components in both nets are regulator circuits, designed specifically for 6 Volts and 5 Volts, respectively. U1 is a LM2576 switching regulator and U3 is an adjustable regulator. The circuitry for both regulator system was implemented with the suggestions of their datasheets. The last stage of the schematics is the output stage for both devices. The power output is designed to connect a 2.1mm male power jack adapter. The LED strip output connector is a 4x1 header that connects to a JST SM plug. Two of the pins are for VDC and GND (which is grounded with the MyRIO to achieve a common ground) and the other two pins are for the Clock Input and Data Input, respectively. These two pins are connected to the MyRIO header to ensure a safe, reliable connection. For all stages, we placed test pins to test and debug our design with the Virtual Bench.

Figure 4 is our final PCB Design. Generally for regulator circuits, it is very important to have each component as close as possible to the regulator itself. In the figure below, U1 and U3 are the regulators of the board. We chose to place them in a central location, which naturally placed the related components near the regulators.

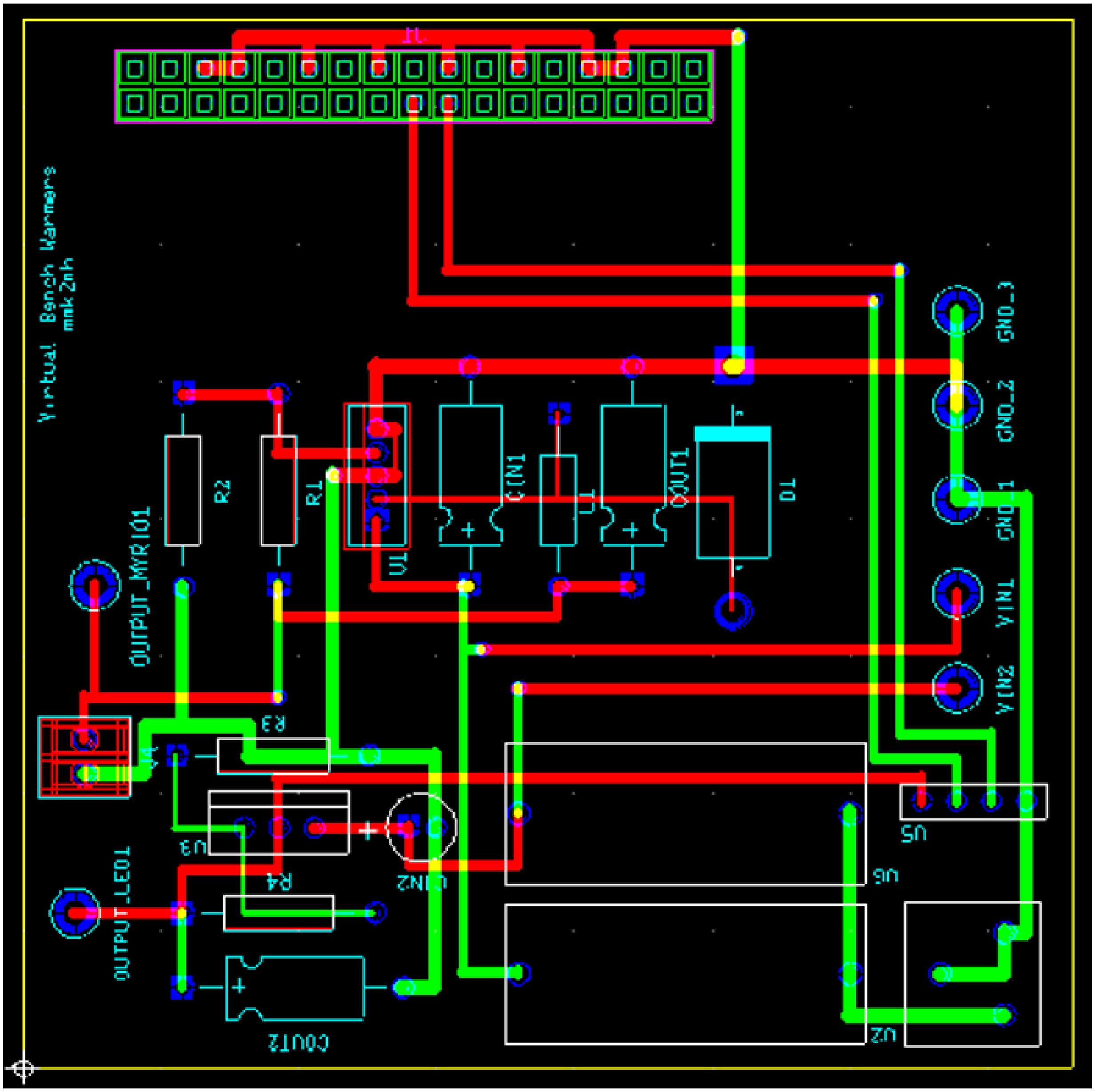


**Figure 4 Final PCB Design**

Another set of components that required special placements were the external connectors such as the MyRIO connector, DC input power jack, RGB LED strip JST SM Plug and the MyRIO input power jack. Due to the fact that we needed to connect the board to our MyRIO, we needed to leave sufficient space near the connector. For the rest of the connectors, we strategically placed them around the perimeter of the board. This allowed us to freely connect/disconnect these I/O connections freely. Lastly, the test pins are placed near the edge of the board for easy access.

One problem arose when we switched the FPGA on the MyRIO to high throughput mode. this mode allowed us to pass several thousand samples at one to our frequency analysis substage and get a good frequency readout, but it also disabled the SPI library we were using to light up the LED strip. We decided that, because our project would not work at all without the frequency analysis, we should keep the high-throughput personality and modify our program to bit bang any SPI output.

## Project Timeline

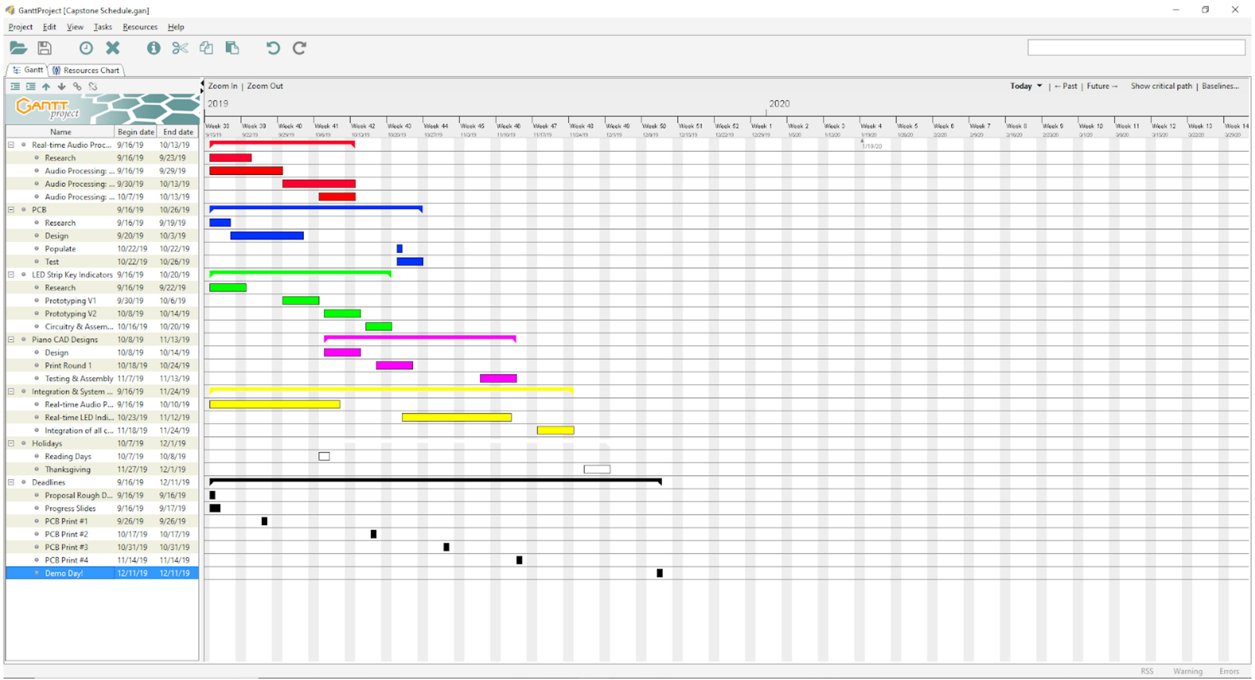

**Figure 5 Initial Gantt Chart**

In our initial Gantt chart, we divided our tasks into 4 categories: Real-Time Audio Processing (Red), Hardware & PCB Design (Blue), Programmable RGB LED Strip Implementation (Green), Piano 3D CAD Designs (Purple) and Integration and System Testing (Yellow). The white items indicate holidays and the black items indicate important deadlines. Our initial thought was the first three categories were high priority tasks and the 3D CAD Design tasks were a lower priority. Ian and Nate were assigned to the Real-Time Audio Processing, Mert was assigned to the Hardware & PCB Design tasks and Eddie was responsible of the LED Strip Implementation and User Experience. All of these tasks were abstracted and isolated from each other categories and were meant to be implemented in parallel. Each divisions were responsible for their own testing. What had to be done in serial was the system integration testing and the 3D CAD designing process, all after each division were ready to combine and test the system.

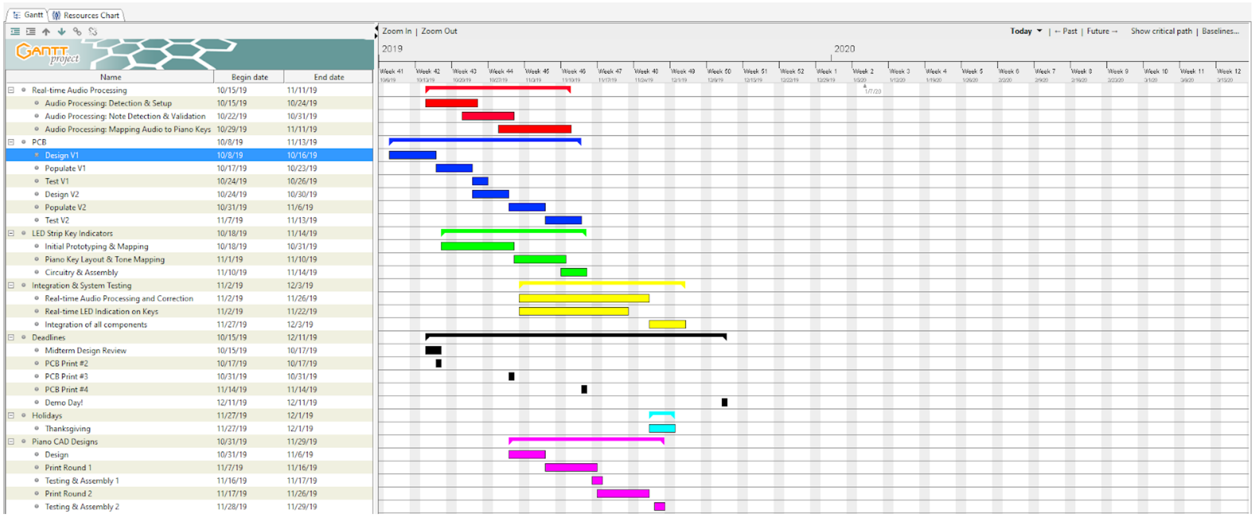


**Figure 6 Midterm Gantt Chart**

As we approached the Midterm Design Review, our schedule and our milestones for the rest of the semester became clearer. Due to ordering and shipping mishaps, we were delayed with our Hardware testing and LED Strip implementation. This pushed us back for approximately two weeks. In contrast, our Real-Time Audio Processing tasks were well underway. In order to compensate the delay on our hardware testing, we needed to allocate extra hours to keep with our previous schedule. We figured out that the LED strip implementation was straightforward. This was the perfect opportunity to assigned Eddie to help with the hardware design to compensate for the delay. This change would help us achieve our set milestones in the beginning of the semester.

## Test Plan

The testing of our project was broken down into testing five different submodules: the PCB board, the microphone audio collection, the LED strips, the song-loaded USB, and the LabVIEW software on the MyRIO.

For testing the PCB boards, we started with a breadboard design using the Virtual Bench to approximately simulate our PCB. We built each regulator circuit independently and then integrated them after verifying expected results. It was important to test the regulators working conditions and heat dissipations for a long period of time. Since our design consisted two, simple regulator circuits implemented with the suggestions of their respective datasheets, we were confident to design our first iteration PCB after verifying the integration of the two circuits. After producing a first iteration PCB, then we would test each component and stage independently to cover each possible flaw in our design. This would be our template testing cycle for our next PCB iterations.

To test audio collection, we made sure that the microphone was had not suffered any damage by plugging it into a laptop and listening to a playback of audio recorded using the microphone. We tested the audio port of the MyRIO by connecting a laptop AUX output to the MyRIO AUX input and check for a detected waveform in LabVIEW. We then attached the microphone to the AUX input of the myrio and made sure that we could read in recorded audio.

The LED strips were tested by running a test LabVIEW virtual instrument that turned on specified LEDs to specific colors and brightness. This ensured that the LED strips were not broken and could still be controlled with SPI.

The song-loaded USB was tested by first double checking the files were correctly added to the USB and that no memory issues had occurred. We then used a test virtual instrument that determined that data was being correctly read into arrays as expected.

Each function within the software was contained within its own sub virtual instrument module. In total 13 different submodules were used in creating the software for our device. Each module was written with expected inputs and outputs in mind. These values formulated black box tests that were used to make sure modules were working as expected. Integrating testing was done with the completion of each module to ensure the total code still worked with the inclusion of each new module. When a software issue was discovered, indicators were placed in the code paths to locate the bug. If an issue was found in a particular module, new tests were made to account for the new issue and code was fixed.

## Final Results

Our final project has two modes: simple and complex. Simple only looks for the user to play the correct notes before moving on, ignoring any incorrect notes played at the same time. Complex runs much more slowly but gives feedback on any incorrect notes it picks up with red LEDs while filtering out any harmonics the microphone is detecting. Both modes display the notes the user should be playing in one color as well as the next set of notes the user should play in another color.

We believe that we have provided all of the functionality that we said we would provide: the device provides full feedback with pitch detection and allows users to upload their own songs. With that being said, much of the functionality is still not smooth. Complex mode will not progress if a user is playing an incorrect note but sometimes picks up notes a student is not playing due to external noise sources. Additionally, if the song requires the same note to be played back-to-back, the program must detect silence before progressing which slows the program down. The filtering out of harmonics in complex mode also removes the ability to detect a user accidentally playing a harmonic of a required note. While these do not affect the core functionality, they are all problems that would have to be addressed before really using this device to learn a song.

The speed of our project is difficult to analyze. While it is slower than an experienced player would be able to play, our project was not designed for that user. The user our project was designed for, a novice who has never played the song on the device before, we think that the pause in between note changes really gives the user time to match the notes they are playing with the notes the see on their sheet music.

## Costs

Based on our parts orders, the cost of our first prototype comes out to about $580. Including the cost of the MyRIO we used would add another $454 to the cost of the project. While over $1000 seems like a lot for this project, especially considering that a quality electric piano could be sold for nearly half of that, this cost could be dramatically reduced in a manufactured product.

If this prototype were to be professionally manufactured and sold in bulk, the production prices would drop quite a bit; many of the parts we used would only be a fraction of the price when ordered in bulk (see the appendix for the ratio of the bulk price to the original price for individual parts), and several of them were not even used. Removing the extra LED strips would discount the price by a substantial $99.90.

Taking into account all the unused parts, our manufactured prototype should only cost $190.72 + $454 for the MyRIO.

If we were to produce 10000 units, we could probably streamline our microprocessor by using only the parts of the MyRIO that we need, and cutting the price down to a roughly estimated $300. Factoring in the bulk prices, each unit would then only cost about $370.5 to manufacture and $3.7 million to produce 10000 units.

## Future Work

We chose to center our project around frequency analysis partially because of how extensible it makes the device. While our device was built for the piano, future development could adapt it for violin, clarinet, or even human voice. Developers would need to change the feedback system to suit the instrument. A piano is relatively easy to guide a user to play but a violin or vocalist would need a much different feedback system. Additionally, while it is easy to divide the frequency spectrum into the 88 piano keys, instruments that could be slightly out of tune would need some sort of frequency tolerance to allow for small errors. At its core, however, this proposed project could re-use much of the code and many of the concepts developed in our project.

Our project could also be improved by speeding up its bottlenecks. We chose a frequency analysis technique that works well, but that does not mean that it’s the best for the job. We found a plethora of published research on the topic of pitch detection while doing research for this project. Implementing these techniques and evaluating their performance could be a valuable addition to our project. We would like to warn future groups that while LabVIEW is a very useful tool, some of the proposed algorithms would be difficult to implement in it.

## References


[1] D. Cooper, "One's Smart Piano helped me play, but not understand", Engadget.com, 2018. [Online]. Available: https://www.engadget.com/2018/11/27/one-smart-piano-learn-to-play/. [Accessed: 24- Sep- 2019].

[2] "The One Piano Hi-Lite", Coolthings.com, 2019. [Online]. Available: https://www.coolthings.com/the-one-piano-hi-lite-led-guide/. [Accessed: 24- Sep- 2019].

[3] “International Standards Reference Chart.” [Online]. Available: https://www.stayonline.com/product-resources/reference-international-plugs.asp. [Accessed: 15-Dec-2019].

[4] “IPC Checklist for Producing Rigid Printed Board Assemblies,” p. 28.

[5] “Musical keyboard: Size & Historical Variation.” Wikipedia.

[6] “Piano key frequencies,” Wikipedia, 05-Dec-2019. [Online]. Available: https://en.wikipedia.org/wiki/Piano_key_frequencies. [Accessed: 16-Dec-2019].

[7] “ Sustainability of Digital Formats: Planning for Library of Congress Collections,” CSV, Comma Separated Values (RFC 4180), 27-Nov-2012. [Online]. Available: https://www.loc.gov/preservation/digital/formats/fdd/fdd000323.shtml. [Accessed: 16-Dec-2019].

[8] Universal Serial Bus 3.2 Specification, 1st ed. USB 3.0 Promoter Group, 2017.

[9] “What Is LabVIEW?,” What is LabVIEW? - National Instruments. [Online]. Available: https://www.ni.com/en-us/shop/labview.html. [Accessed: 16-Dec-2019].

[10] “What is Multisim?,” What is Multisim™? - National Instruments. [Online]. Available: https://www.ni.com/en-us/shop/electronic-test-instrumentation/application-software-for-electronic-test-and-instrumentation-category/what-is-multisim.html. [Accessed: 16-Dec-2019].

[11] “Ultiboard,” Ultiboard - National Instruments. [Online]. Available: http://www.ni.com/en-us/shop/select/ultiboard. [Accessed: 16-Dec-2019].

[12] “What Makes LED Lighting Sustainable?,” Energy Focus. [Online]. Available: https://energyfocus.com/blog/led-lighting-sustainable/. [Accessed: 16-Dec-2019].

[13] Electronic apparatus to assist teaching the playing of a musical instrument, by F. S. Pierce, J. L. Clemons, W. V. Davis, R. H. Dupont, R. D. Sartore, and L. Stricklan. (1995, Jul 2).US5394784A

[14] Musical instrument training system having displays to identify fingering, playing and instructional information by Frederick W. Breitweiser, Jr and Paul F. Weeks, Jr. (1995, Apr 25). US5408914A

[15] Electronic piano training device by Kevin M. McCarthy. (2014, Dec 2).US8901405B1

[16] “Extract Multiple Tone Information VI,” Extract Multiple Tone Information VI - LabVIEW 2018 Help - National Instruments. [Online]. Available: https://zone.ni.com/reference/en-XX/help/371361R-01/lvwave/extract_multiple_tone_information/. [Accessed: 16-Dec-2019].

[17] “What is myRIO?,” What Is myRIO? - National Instruments. [Online]. Available: https://www.ni.com/en-us/shop/engineering-education/portable-student-devices/myrio-student-embedded-device/what-is-myrio.html. [Accessed: 16-Dec-2019].

[18] Adafruit Industries, “Adafruit DotStar Digital LED Strip - Black 144 LED/m - One Meter,” adafruit industries blog RSS. [Online]. Available: https://www.adafruit.com/product/2241. [Accessed: 16-Dec-2019].

# Appendix

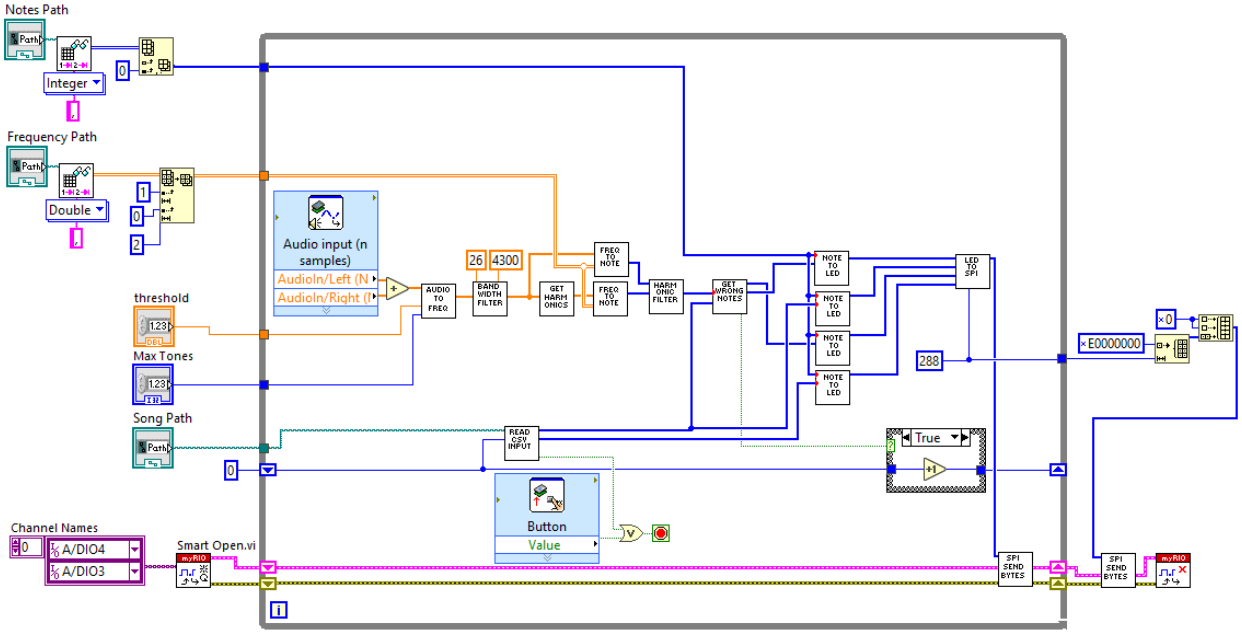


**Figure 7 Complex Mode Program Flow**

## Costs Spreadsheet

This is a spreadsheet containing all the parts we ordered. Due to changes in design and the natural progression of our prototype, a handful of the parts were not used. Parts will be highlighted yellow if Used (if quantity not highlighted, assume 1 used), and teal if Not used.

| # of Item | Manufacturer | Description | Ind. Price | Total Price | Link | Mass Produc. ratio |
|---|---|---|---|---|---|---|
| 15 | Littelfuse | 12.5 Amp fuse | 4.26 | 63.90 | 1 | 0.53 |
| 5 | J.W. Miller | 330uH Inductor | 1.77 | 8.85 | 2 | 0.5 |
| 2 | Adafruit Industries LLC | RGB LED strip | 49.95 | 99.90 | 3 | 1 |
| 3 | Adafruit Industries LLC | Female DC Power adapter - 2.1mm jack to screw terminal block | 2.00 | 6.00 | 4 | 1 |
| 4 | Adafruit Industries LLC | 4-pin JST SM Plug + Receptacle Cable Set | 1.5 | 6 | 5 | 1 |
| 10 | Schurter Inc. | 4 amp fuse | 2.89 | 28.9 | 6 | 0.42 |
| 5 | Schurter Inc. | 10 amp fuse | 2.89 | 14.45 | 7 | 0.42 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 10 | Littelfuse Inc. | fuse holder (x2) | 1.27 | 12.7 | 8 | 0.45 |
| 4 | TE Connectivity AMP Connectors | molex connector 2 pin (282834-2) | 0.88 | 3.52 | 9 | 0.4 |
| 2 | Tensility International Corp | power cord | 3.08 | 6.16 | 10 | 0.54 |
| 10 | STMicroelectronics | diode (1N5822) | 0.38 | 3.8 | 11 | 0.58 |
| 10 | Stackpole Electronics Inc | 2.2k resistors | 0.07 | 0.7 | 12 | 0.19 |
| 10 | Stackpole Electronics Inc | 6.8k resistors | 0.07 | 0.7 | 13 | 0.19 |
| 5 | Microchip Technology | lm2576-5.0WT | 1.79 | 8.95 | 14 | 0.83 |
| 5 | Vishay BC Components | 100uF capacitor | 1.9 | 9.5 | 15 | 0.45 |
| 5 | Vishay BC Components | 330uF capacitor | 2.15 | 10.75 | 16 | 0.64 |
| 2 | SparkFun Electronics | Power connector for MyRIO | 2.95 | 5.9 | 17 | 1 |
| 15 | TE Connectivity Aerospace, Defense and Marine | wires | 0.49 | 7.35 | 18 | 0.41 |
| 10 | Stackpole Electronics Inc | 820 ohm resistors | 0.07 | 0.7 | 19 | 0.19 |
| 10 | Stackpole Electronics Inc | 270 ohm resistors | 0.07 | 0.7 | 20 | 0.19 |
| 10 | Vishay BC Components | 1uF capacitors | 1.24 | 12.4 | 21 | 0.56 |
| 10 | Nichicon | 0.1uF capacitors | 0.17 | 1.7 | 22 | 0.27 |
| 1 | Lavalier Lapel | Microphone | 20.99 | 20.99 | 24 | 1 |
| 1 | Adafruit | Adafruit 5V 10A Power Supply | 37.99 | 37.99 | 25 | 1 |
| 5 | Texas Instruments | LED Adjustable Regulator (LM338) | 2.1 | 10.5 | 28 | 0.42 |
| 10 | Bourns Inc. | 150uH inductor | 0.1 | 1 | 29 | 0.9 |
| 3 | Texas Instruments | Switching-Regulator (lm2576-ADJ) | 2.77 | 8.31 | 30 | 0.47 |
| 1 | Adafruit Industries LLC | wall adapter 5V 10A Power Supply | 38.94 | 38.94 | 31 | 0.82 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 5 | Illinois Capacitor | 470uF cap | 0.85 | 4.25 | 32 | 0.32 |
| 5 | Bourns Inc. | 150uH inductor | 0.31 | 1.55 | 33 | 0.39 |
| 3 | Adafruit Industries LLC | DC power jack | 1.95 | 5.85 | 34 | 1 |
| 7 | Sullins Connector Solutions | MyRIO header connector | 2.42 | 16.94 | 35 | 0.53 |