Self-Playing Xylophone with Real-Time Note Detection

A Technical Report submitted to the Charles L. Brown Department of Electrical and Computer Engineering

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Kidus Fasil

Spring 2024

Technical Project Team Members

Charles Monsell

Kaden Hoversten

Brendan Grimes

Mohamed Mahmoud

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Adam Barnes, Department of Electrical and Computer Engineering

Capstone Final Report

Simophone

"Simon Says" “Play the Xylophone “

“The Perfect Fifth”

Charles Monsell

Kidus Fasil

Kaden Hoversten

Brendan Grimes

Mohamed Mahmoud

ECE 4440 Fall 2023

Prof. Adam Barnes

Statement of Work:

Mohamed

I was tasked with the assembly and circuitry side of the project. I started with the assembly of solenoids and LEDs on a xylophone, utilizing custom 3D printed mounts and tools to cut/resize/install the main frame. This process, repeated 32 times, involved several failed approaches to finally have a final assembly that required an addition of small PCBs on each solenoid's surface. These PCBs were populated with diodes, LEDs, and resistors, requiring precision and attention to detail. Additionally, I managed the installation and organization of approximately 120 wires and connections, ensuring efficient and accurate electrical integration. My responsibilities extended to helping with the design phase, where Charlie and I utilized KiCad software to create the PCB layout. Following the arrival of the PCBs, I undertook the task of populating and testing them, ensuring each subsection functioned correctly. My role also encompassed various testing activities, including verifying the functionality of the microphone, fine-tuning key sounds, and adjusting the buck converter.

Charlie

I was responsible for designing the hardware and circuitry for the project. This consisted of the solenoid drivers, microphone selection, relay circuitry, and the buck converter. I calculated the power requirements for all our peripheral components and chose a buck convertor I.C. Next, I designed and routed our main PCB. I also assisted in assembly by soldering the main PCB, routing the necessary wires from each solenoid key to the main PCB, and by adjusting/ fine tuning each of the solenoid's distance to each respective key. Beyond hardware and assembly, I also helped Kidus develop the method and implementation of the note detection algorithm for "Simon Says". First, Kidus and I started with finding a DSP library with optimized FFT functions for our sampled data from the microphone. Then, we implemented an algorithm to calculate which note was played based on the frequency response. Lastly, I fine-tuned the note detection algorithm to ensure that notes get picked up appropriately.

Kaden

I was tasked with hardware and assembly of the project. To start off the project, I contributed to the process of designing the overall system, specifically the circuitry for the solenoids and LEDs. As Charlie and I settled on a final driver circuit design, I began to model and 3D print the solenoid/LED mounts. This took several iterations due to printer failures and tweaks being made to the model. Eventually, after all 32 of the mounts were complete, I assisted Mohamed with assembling the mounts and populating the small PCBs. Sometime during this process, we realized that our original power delivery system would not function, due to a combination of some of our solenoids being faulty, and an unexpectedly large voltage drop across the power transistors. We decided to upgrade our power supply to a higher voltage, and Charlie and I redesigned our power supply to use relays, with coil current being supplied by the power source itself and being controlled by a BJT driven by the microcontroller. As the assembly process continued, I designed an enclosure for our PCB, microcontrollers, and power supply.

This was then 3D printed and assembled by me. Mo did most of the wiring after all the parts were in place.

Kidus

I was tasked with the software for the project. I primarily wrote the control software on the MSP432 in C. This involved the initial SPI communication from the Raspberry Pi to the MSP432, receiving and parsing MIDI information from the Raspberry Pi. Furthermore, Charlie and I wrote the audio processing code, including the note detection algorithm and some of the ADC sampling code. I also wrote the code for "Simon Says" on the MSP432 side which includes listening to the user until the song is over and the comparison algorithm between the user input and the actual song. From the Raspberry Pi's perspective, I wrote the MIDI parsing algorithm that reads a midi file, deconstructs the file, and sends the relevant information for the MSP including the beats per minute (BPM), pulses per quarter note (PPQ), total number of notes, and the start time, note value, and duration of each note in C++.

Brendan

I was tasked with the user interface software on the Raspberry Pi and (along with Kidus) the communication and control flow software for the MSP. The user interface was written in Qt 6.6 with a C++ backend that incorporated Kidus's code for MIDI file processing and SPI interactions. This required that I cross-compile (on my laptop) the Qt framework from its source code using the libraries from the Raspberry Pi, resulting in a set of binaries that could be used to run my application on the environment present on the Raspberry Pi. I also was responsible for the code on the MSP that drives the GPIO pins on the MSP to fire the solenoids and the algorithm for shifting musical notes to ensure they fit within the range of the xylophone. I was also largely responsible for parts ordering and some other administrative tasks.

Abstract

The "Simophone" project represents a cutting-edge fusion of talent, entertainment, music, and technology, offering an immersive and educational musical gaming experience. The Simophone features a self-playing xylophone capable of dynamically generating random tunes sourced from a vast library of MIDI files. The user's objective is to hone their listening skills and musical memory by attentively absorbing the melody produced by the Simophone and then translating it into precise tune by striking the correct notes on the xylophone. The Simophone captures the user's performance in real-time through a microphone, meticulously analyzing each note played. Subsequently, the system compares this live recording to the original MIDI file, delivering instantaneous feedback to the player. Success hinges on the player's ability to accurately replicate the tune, fostering an engaging challenge that encourages memory enhancement and musicianship skills. In the Simophone, we have seamlessly merged the worlds of music, technology, and cognitive stimulation, offering a combination of entertainment, learning, and musical expression.

Background

There has been a myriad of studies done that have shown the benefits of utilizing music in the education system. Evidence has associated musicianship with advantages in general cognitive functions, often loosely related to musical skills, such as intelligence, visuospatial abilities, processing speed, executive control, attention, and vigilance, and episodic and working memory [1]. Our project aims to make learning music easy, fun, and accessible so everyone can reap the cognitive benefits.

The inception of the Simophone project was driven by our team's collective interests and strengths. Our ideas gravitated toward music-related concepts, eventually culminating in the proposal for the Simophone. Several factors underpinned our decision to pursue this idea, including its potential to benefit a diverse user base, the project's feasibility, and its scalability with minimal resources. Furthermore, there will be a user interface, whether that would be on an LCD screen or through an application, that the user will be able to interact with to select different modes that the Simophone has.

When performing background research, we encountered a noteworthy precursor, the "Automated Xylophone" project [2]. This prior endeavor shared some fundamental characteristics with the Simophone, notably the use of MIDI files for input, solenoids for striking the xylophone bars, and transistors for solenoid control. However, the Simophone distinguishes itself by incorporating a unique "Simon Says" feature. This distinctive addition empowers users to engage in a musical sequence playback game, wherein the xylophone generates a sequence of notes for the user to replicate.

Our coursework background has played a pivotal role in assisting us to tackle the challenges in this project. Notably, the *ECE Fundamentals* Series, *Digital Signal Processing*, *Electromagnetic Energy Conversion*, *Introduction to Embedded Systems*, and *Embedded Computing and Robotics 1* courses have provided us with a solid foundation. The *ECE Fundamentals* Series instilled in us the ability to design circuits, simulate electronic systems, and analyze signals and systems. Furthermore, it acquainted us with key electrical components such as BJTs, diodes, resistors, and LEDs, which are integral to our project's hardware design. *Digital Signal Processing* was especially instrumental as it addresses the substantial audio processing requirements of the “Simon Says” feature. Crucially, the primary algorithm for comparing user and xylophone audio signals, the Fast Fourier Transform, was covered in this course. Three of our team members, namely Charlie, Kidus, and Mo, have successfully completed *Digital Signal Processing*. The embedded systems courses have been instrumental in honing our skills for the project, particularly in utilizing the MSP432 microcontroller shared between these courses and our project. All team members have completed *Introduction to Embedded Systems*, and Brendan, Charlie, and Kidus have gotten more experience with the MSP432 by completing *Embedded Computing and Robotics 1*. Lastly, *Electromagnetic Energy Conversion*, a course completed by Charlie and Mo which covered the theory behind the workings of solenoids which are one of the most important mechanical aspects of the project.

Societal Impact Constraints

User Stakeholder Considerations

One of the largest appeals of the Simophone is that it can provide mentally engaging entertainment for a wide variety of audiences. While any user should be able to successfully play a game of Simon-Says using easy mode, the medium and hard modes should appeal to those who wish to hone their auditory recognition and xylophone skills. The main stakeholders in this project would therefore be anyone with a desire to learn how to play the xylophone, or a desire to learn to recognize notes by ear, or simply a desire to have some musical fun. However, this is an overly broad category, so we will break it down into subcategories and explore the concerns of each of these groups individually. While individual children are the largest and most substantial user group, classrooms and adult hobbyists also can potentially draw value from the Simophone. To address the concerns of all these user groups in our original prototyping, we have sought to emphasize the concerns of child stakeholders and their parents by making the device sturdy, educational, easy to use, and fun, while still taking into consideration the desire for more advanced features that would be desirable for a teenager or adult hobbyist stakeholder subset. The classroom setting would require a slightly distinctive design philosophy and a separate variant if commercially marketed, but many of the concerns present for other stakeholder groups are present for this group too.

Children

Because the Simophone is primarily designed to be a game, its primary users would be children. However, due to the game's challenging nature, it is likely that these would mostly be older children, perhaps seven or eight years old at the youngest. While it would be theoretically possible, depending on their manual dexterity and comprehension levels, for younger children to play "Simon Says" with the Simophone, interacting with the user interface would be too complex for most users below this age. Thus, while durability will remain a design factor, we need not build it to withstand the full destructive force of a toddler. Ideally, the device would be able to withstand a small drop without breaking, but we need not make it waterproof nor sturdy enough to be used as a stool.

An additional consideration for this stakeholder group is that their parents would purchase the device. Thus, if this device were to be produced, we would need to focus our marketing attention on the aspects of the device that would appeal to parents. Particularly, we would focus on the ability to provide ear training, build manual dexterity, and reflex speed, and evolve musical percussion skills within an entertaining and gamified framework that requires no parental oversight. We would also need to take measures to reduce the cost, as $500 is a quite a sizeable outlay for a single Christmas or Birthday present and could be entirely prohibitive for lower income families. As previously discussed, one aspect of this would be shifting the user interface to a companion mobile application, which would reduce costs and ease maintainability, potentially even allowing new features to be added without needing to repurchase the device. We could also potentially scale the device down, allowing for the use of smaller (and hence cheaper) solenoids.

Music Students

While this device is primarily intended as a toy, its highly educational tendencies make it potentially suitable for a classroom setting. Once again, this would only be useful from late elementary school onwards. If we were to market the Simophone for a classroom setting, we would need to focus much more heavily on durability. In this context, there would be no need to replace the screen and Raspberry Pi with a mobile application. Rather, we want to use a more commercial microcontroller here or in place of the Raspberry Pi and the MSP432. In the classroom setting, cost will be less of a concern than durability and the ability for the product to stand alone as a single entity. Of course, we would still like to keep the cost as low as possible for this setting as well, especially to be inclusive of lower income school districts.

Adult Hobbyists and Musicians

Though adult hobbyists, xylophone enthusiasts, and percussionists are the smallest portion of the potential userbase of the Simophone, they can still find some value in it. They primarily stand to benefit in much the same way as do children, but they may find more value in the hard mode of game play and in the ability to playback MIDI files than a child would. The ability to take a MIDI file and listen to how it would be played on the xylophone while seeing which keys have been struck could be helpful, especially for someone wanting to learn the xylophone. And since the Simophone can be played in the same way as any normal xylophone, as well as used in "Simon Says" mode, one can practice or play on the instrument freely when desired by simply letting the program idle without initiating a new game of "Simon Says". Trained percussionists and xylophone enthusiasts, though they may not have need of the educational functionality provided by the Simophone, could find it entertaining as a toy or curiosity.

For this stakeholder group, while we would need to make sure that the device is not fragile, durability is less of a concern than is cost. The key appeal in this stakeholder group is for those looking to learn the xylophone, much as for children, so many of the concerns would remain the same. The main difference, however, is that more advanced features, such as arbitrary MIDI playback, would have much more value for this user group than for children. Consequently, to commercially market to this stakeholder category, we would want to spend more effort working on advanced features that can provide value to this group, such as perhaps the ability to harmonize in real time with what the user plays on the xylophone.

Additional Stakeholder Considerations

While the user base is certainly the largest portion of our stakeholders, should the Simophone become a commercial product, we would also need to consider the manufacturer's concerns, as well as such things as shipping, supply chains, and the like. The primary factor here is manufacturability and cost. The prototype, while useful to build a general idea of how the system could work, would need to be refined as mentioned above to use a single microcontroller, and to support a mobile app for the user interface. This would require substantial design effort, both in terms of software and hardware, but could potentially result in tangible manufacturability improvements, as well as reducing the overall total cost. One change that would also likely reduce costs is to use machine-solderable surface-mount parts instead of through-hole components. Additionally, bulk components are cheaper than buying small quantities for a prototype, which will also reduce total costs. A notable exception to this is solenoids, however, which do not seem to be sold with bulk discount pricing. As mentioned above, we would need to examine ways to trim this cost down by potentially using smaller solenoids.

Public Health and Safety

The Simophone, designed as a musical device, must comply with strict safety standards, including electrical safety and child safety, to ensure its suitability for all users, especially children. This requires securely encasing all components to eliminate risks of electric shock or mechanical injury. A warning label about the electric shock risk is necessary since the device operates on 120V. Additionally, it's vital to regulate sound levels to protect against hearing damage, particularly in children who have more sensitive hearing. Furthermore, the design should be ergonomic to prevent strain or injury, catering to the needs of younger users and those with disabilities, ensuring comfortable and prolonged use.

Figure 1 Warning label

Welfare and Accessibility

In designing the Simophone, it is essential to emphasize inclusivity, ensuring it is accessible to users with disabilities. This can involve integrating features for visually impaired users, such as tactile controls or audio feedback, and a simplified user interface for those with cognitive disabilities, thereby broadening the device's appeal and enhancing its utility. Additionally, the positive impact of engaging with music on mental health should be highlighted. This aspect is particularly beneficial when marketing the Simophone to parents or educators, as it aligns with the growing awareness and importance of mental well-being in educational and home environments.

Global and Cultural Factors

In today's global market, cultural sensitivity is paramount, and for a product like ours, it means incorporating features that reflect diverse cultural backgrounds. This could be achieved by offering a diverse range of music or enabling users to upload their own MIDI files for a more personalized experience that reflects diverse cultural backgrounds. Alongside this, to truly resonate with a global audience, multilingual support in the user interface (UI) and instructional materials is essential, ensuring that users from different linguistic backgrounds can easily navigate and fully engage with the product.

Social Impact

To enhance the appeal of the Simophone to schools and educational institutions, it's essential to emphasize its educational value, such as facilitating the learning of music theory and improving cognitive skills. Alongside these educational aspects, the device should also incorporate features that promote social interaction, like multiplayer modes and options for sharing performances online. While fostering this social engagement, it is crucial to adhere to privacy and online safety standards to ensure a secure and responsible user experience.

Environmental and Economic Impact

For the Simophone, sustainability is a key consideration, emphasizing the use of environmentally friendly materials and manufacturing processes, alongside a focus on energy efficiency to minimize the device's energy consumption. Additionally, the design is oriented towards recyclability, ensuring that at the end of its lifecycle, the Simophone can be easily recycled, thereby contributing to the reduction of electronic waste and promoting environmental responsibility.

To ensure wide accessibility, the product must strike a balance between quality and affordability, making it attainable for lower-income families and schools. Simultaneously, it's crucial to adapt the product for various market segments. This includes creating a more robust version tailored for educational settings, which can withstand frequent use and handling, and a more feature-rich version designed for hobbyists who seek advanced capabilities and functionalities. This dual approach allows the product to meet diverse needs and preferences across different consumer groups.

Legal and Ethical Obligations

To ensure the successful launch and sustained market presence of the product, it is essential to comply with all relevant laws and regulations, including consumer protection laws, in each market. Additionally, offering a clear warranty and support policy is crucial, alongside being prepared for any liability issues, ensuring that the product has undergone all necessary safety tests. Marketing should be conducted with transparency and honesty, openly communicating the product's capabilities and limitations to foster consumer trust. Furthermore, if the device collects any data, robust data protection and privacy policies must be in place, a critical aspect especially if children are among the users, to safeguard their sensitive information.

By addressing these considerations, you not only fulfill a moral and ethical obligation but also position the Simophone as a socially responsible and globally aware product. This comprehensive approach can enhance the product's appeal and mitigate risks, including legal liabilities.

Physical Constraints

Cost Constraints

Managing a project with a tight budget of $500 presents significant challenges, particularly in terms of component selection and contingency planning. The allocation of nearly 50% of this budget to solenoids, a crucial component, leaves only $250 for other necessary parts and PCB manufacturing. This financial limitation significantly restricts our ability to handle unexpected issues or make design modifications, necessitating thorough research and careful planning to minimize the risk of failures or changes.

Despite our best efforts, unforeseen problems can still arise. A recent example involves the solenoids we procured. These components were rated for 12V, and accordingly, we purchased a 12V power supply. However, during testing, we discovered that 30% of these solenoids actually required more than 15V to operate effectively. This discrepancy forced us to acquire a new 16V power supply and subsequently redesign the circuit for the buck converter to accommodate this change.

Such incidents highlight the challenges of working within a strict budget, particularly when unexpected equipment failures or manufacturer errors occur. They underscore the importance of allocating funds for contingencies and the need for flexibility in both design and financial planning. In future projects, it may be beneficial to reserve a portion of the budget for such unforeseen circumstances, even at the expense of other components or features, to ensure smoother project execution.

Design and manufacturing constraints

One of the key design and manufacturing limitations we encountered was the necessity to design trace sizes that are adequately large to handle the current drawn by the solenoids. This requirement must be carefully balanced with the constraint that the overall size of the PCB must not exceed 60 square inches, posing a significant challenge in efficient space utilization and layout optimization.

On the software side, we initially faced limitations while using Ultiboard for our PCB design. A major hurdle was that Ultiboard did not possess the footprints for many of the components we planned to use. This lack of compatibility compelled us to switch to KiCad for designing the PCB. KiCad, with its extensive library and greater flexibility, allowed us to overcome the footprint limitation. This transition, while necessary, also involved a learning curve and adjustments in our design approach to accommodate the different functionalities and interface of KiCad.

There were also memory limitations that were faced during the project. The MSP432 only has 32 kB of ROM and 64 kB of RAM. As a result, the length of songs that we could use for MIDI playback is constrained by the amount of memory that the MSP432 has unless there are available blocks of flash memory.

Tools

The major tools that we utilized throughout the semester were Code Composer Studio (CCS), Visual Studio Code, KiCaD, PuTTY, Git, C, C++, and Qt. CCS was used as the development environment for the MSP432 which is programmable in C. Visual Studio Code was used as the development environment for the Raspberry Pi which used C++ to run the backend code of the user interface and Qt for the graphical user interface. To write code the Raspberry Pi, we established an SSH connection with the Raspberry Pi. To establish this connection, PuTTY was used. Git was used as the version control system for all the software for the project. Lastly, KiCad was used to simulate and design the PCB for our project. For testing, an AD2, oscilloscope, and a function generator were used. Some of these tools were utilized by team members in the past like Git, C++, CCS, and VSCode in which we improved our skills in. However, some of these tools we learned how to use throughout the semester like Qt and KiCad.

External Standards

There are several standards to take into consideration for the Simophone.

PCB Design

As the Simophone can draw on quite a bit of power, extra considerations will have to be made as we design the PCB and our wiring layout to ensure safety. We will be referring to IPC-2152 [12] to determine proper trace sizes on our board, as not to cause overheating.

Wire Gauges

Additionally, National Electrical Code (NFPA-70) [11] standards will be followed as we design the Simophone. Contained within NFPA-70 is a chart detailing ampacity of wire gauges, although it does not extend to thin enough gauges for our intended uses. Instead, we will refer to “AWG Wire Gauges Current Ratings” [10] for smaller gauge wires.

Noise Exposure

Because the functionality of the Simophone is reliant on sound, considerations must be made of the noise level that the xylophone could make when striking a bar. We will be following the OSHA standards 29 CFR 1910.95 [13] to assure the xylophone does not exceed the maximum decibel level specified.

Power Supply

We purchased a 16 V switch-mode power supply that complies with the EMC standard EN61204-3: 2000 [19].

USB Cable

We purchased USB A Cable for the Raspberry Pi that complies with the USB 2.0 standard.

Electrical Tape

To insulate wires, we purchased electrical tape that complies with the UL-510 standard [14].

Intellectual Property Issues

Our implementation of the game "Simon," is substantially different from that of the original patent [7]. The original game consists of an electronic board game, with four buttons, wherein the user repeats patterns that are shown by the game. The game shows these patterns using lights and tones. Our game is much more complicated, as it has not four buttons, but thirty-two xylophone keys which we detect an input from. In addition to this, our implementation uses pre-existing patterns, whereas the original produces it randomly.

Looking at self-playing instruments, our Simophone is still mostly unique. Most patented self-playing xylophones do not use solenoids to strike the keys, and most also do not use a digital media format for playback of songs. Many devices are more mechanically driven than ours, and many are simply toys [9], and many are only able to play a single tune, similar to a musical box. Many self-playing instruments, especially player pianos, are devices that can be installed upon standard models of the base instrument. In fact, it is a primary claim of patent *US3472110A* [8], along with many others.

We believe that our project is patentable, as it is distinguishably different from every patent that we were able to find. For each feature that exists in a preexisting, patented design, we have greatly improved upon it. We also have many features that do not exist in any other devices, such as the functionality that allows us to teach our users how to play songs.

Project Description

Performance Objectives and Specifications

The proposed project is a self-playing xylophone capable of performing MIDI files that can be directly uploaded into the system's local storage. Beyond just playing MIDI files, the project aims to be an interactive experience by playing "Simon Says" with the user. The idea is that the Simophone will play a string of notes, a part of a song for example, and the user would be expected to play it back. The goal of this project is to help aspiring musicians learn the basic parts of music theory such as rhythm, chord progression, and scales in a way that is engaging and simple to understand.

The next core feature for the project is the implementation of "Simon Says". Since this project is oriented around music, this feature will revolve around teaching the user parts of a song or scales. For this to work, a tempo will be set by the system and will be relayed to the user via flashing an LED. This LED will remain flashing for the entire game to help the user keep tempo. Once the tempo is set, the Simophone will begin to play something for a defined block of time. After the Simophone is finished playing, a different colored LED from the one keeping the tempo will begin to flash coordinated with the tempo LED to count off to the entrance of when the user is supposed to play. The user will then attempt to play back what was played and if correct, the simophone will play something slightly more difficult. If not correct, the game ends and the user will be prompted to start over.

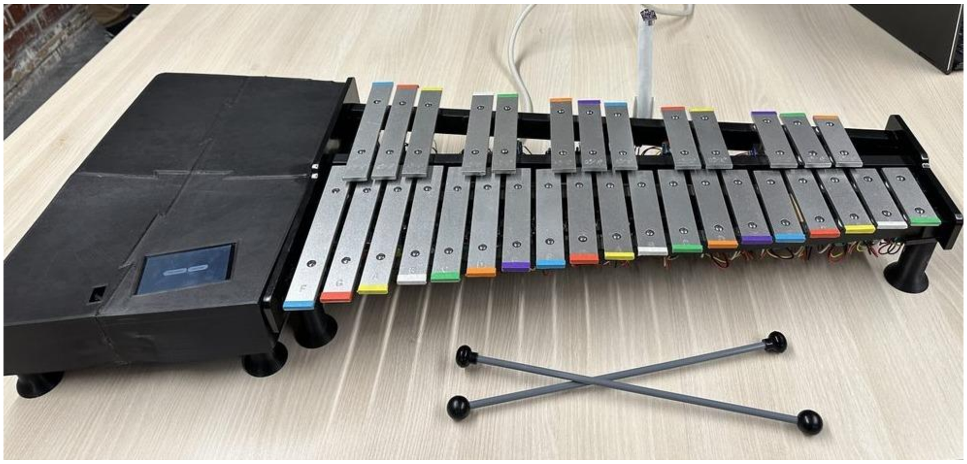

Figure 2 Final version of the Xylophone

Diagrams and Schematics

The overall block diagram of the system is shown in Figure 3 below. The Simophone is composed of the Raspberry Pi 3B+, MSP432, Microphone, Xylophone, LCD/UI, Power Supply, LED Drivers, and Solenoid Drivers. Each arrow represents either an electrical connection or data sent between the devices.

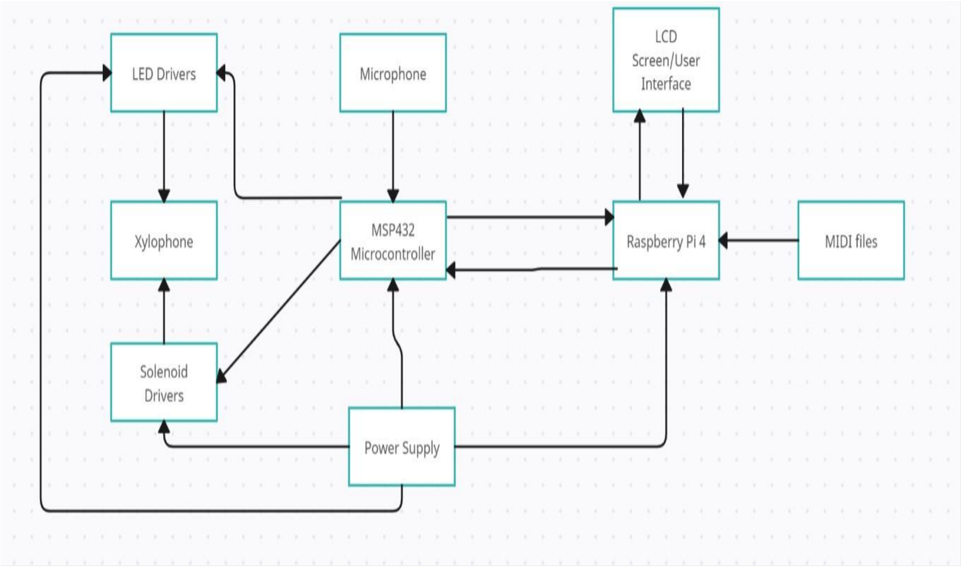

Figure 3 Block Diagram of the System

The overall system is interacted by the user interface which is a touch screen connected to the Raspberry Pi. The MIDI files are stored in the Raspberry Pi and when a user selects a

MIDI file to do "Simon Says" or MIDI playback, the relevant MIDI information is sent to the MSP432. The MSP432 then either actuates the solenoids and/or LEDs through its GPIO pins which are underneath the xylophone, or it uses the microphone to record the users input through the MSP432's analog to digital converter (ADC). The power supply powers the Raspberry Pi, MSP432, and supplies the LED and Solenoid rails.

The most important hardware component of our project is the solenoid drivers. Figure 4 below shows the repeated circuit pattern for each key of the xylophone (32). In this case, the inductors represent the solenoids, the diode (1N4004) in parallel with the solenoid is the freewheeling diode to suppress switching transients. The next diode is to prevent current from flowing back through all of the LED whenever the solenoid rail is turned off. Lastly, the resistor is just for current limiting for the LED. The transistor is used to drive the necessary current through the solenoid and the LED and to be controllable by a GPIO pin from the MSP. The chosen transistor was a Darlington pair BJT as seen in [5].

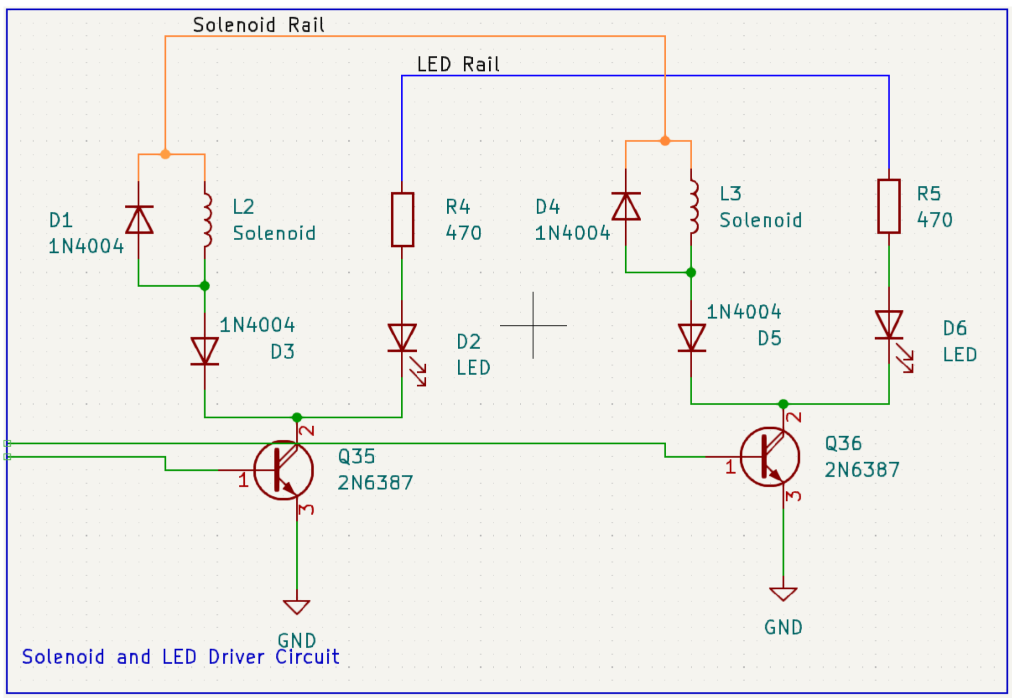

Figure 4 Example Solenoid Driving Circuit

The schematic in Figure 5 below is the buck converter circuit used to step down our 16V power supply down to 5V to drive the MSP, microphone, RPi, and the LCD screen. The power consumption for all the components adds up to around 900mA (600mA for the RPi, 100mA for the MSP, 150-200 for the LCD). The current consumption of the microphone is negligible. To handle power requirements, a constant 5v output buck converter rated for three amps was chosen. In this case it was the LM2596SX as seen in [18].

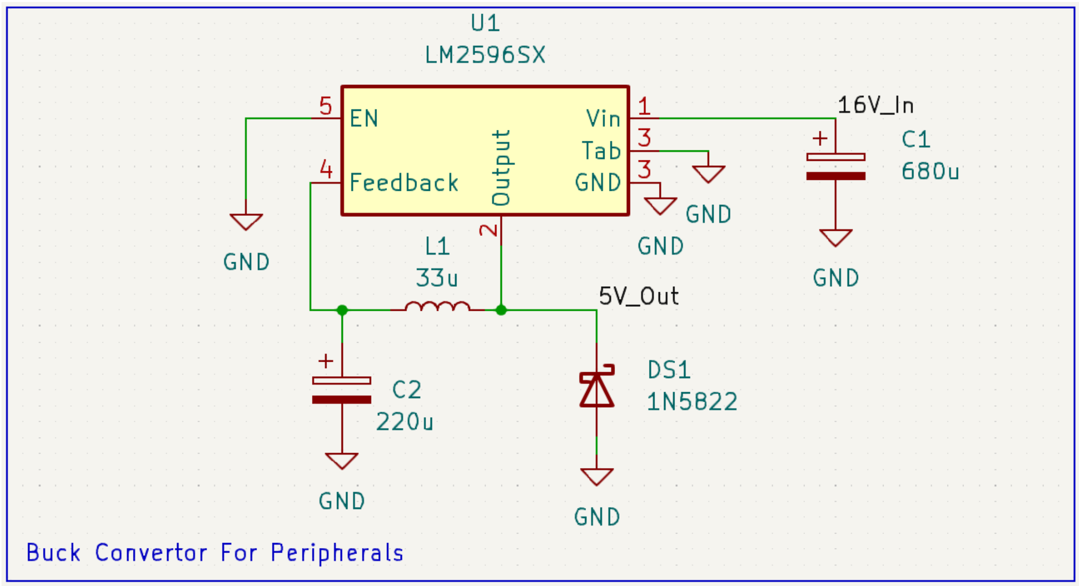

Figure 5 Buck Convertor Schematic

Lastly, Figure 6 below shows the circuitry to control the relays to switch the two power rails; one for the LEDs and one for the solenoids. The different power rails correspond to difficulty. Higher difficulties disable the LEDs to help train the user to remember notes solely based on sounds. The solenoids can also be disabled if the user chooses to. To control the relays, transistors are used to drive the coils to actuate the switch. The BJTs are controlled by GPIO pins on the MSP.

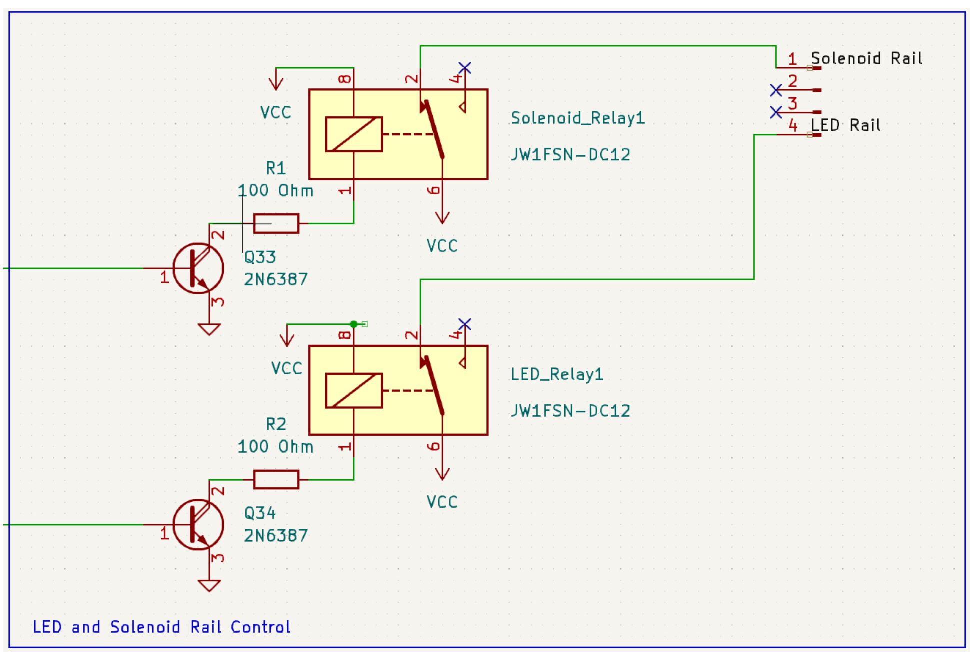

Figure 6 LED and Solenoid Rail Control Schematic

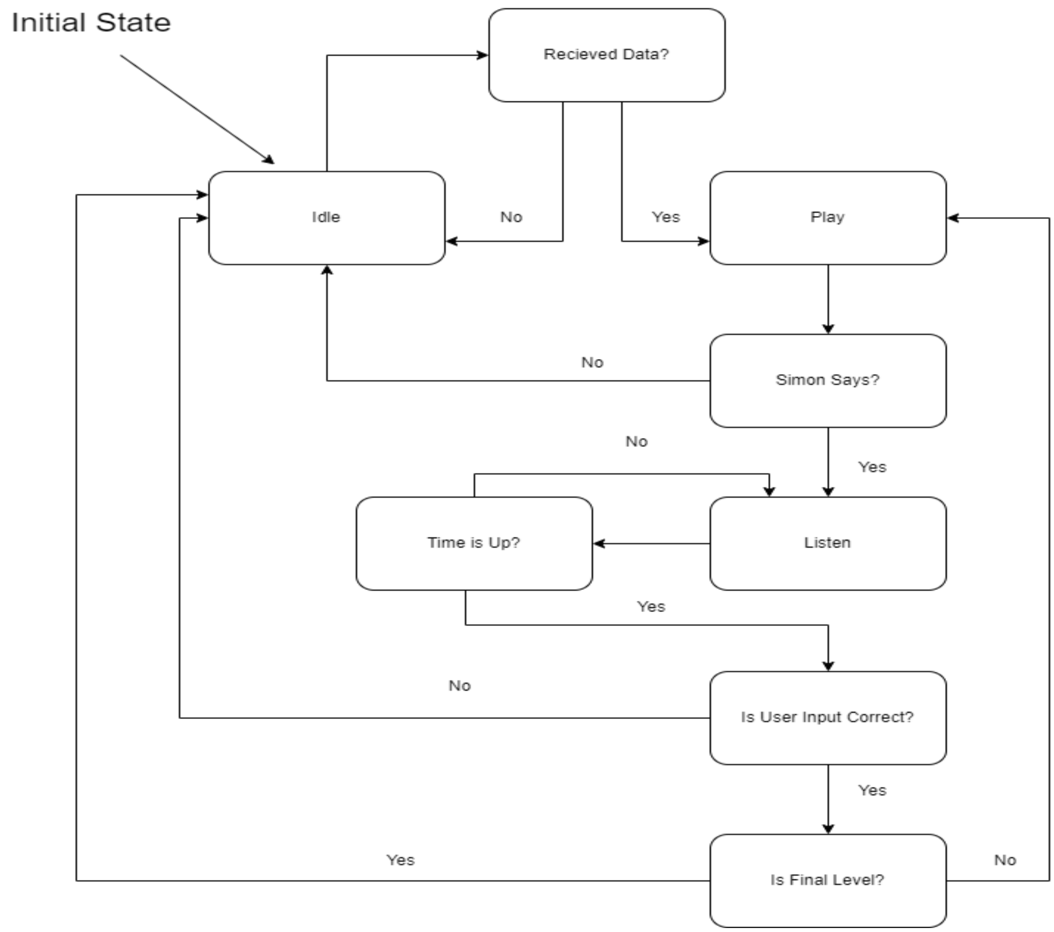

Figure 7 Software Flowchart of the MSP432

Figure 7 represents the software flowchart of the MSP432. The MSP432 initially starts out in an idle state waiting to receive data from the Raspberry Pi. Once it receives data from the Raspberry Pi, it will determine from the information sent if the mode is "Simon Says" or MIDI playback. Regardless, the xylophone will play the notes sent from the Raspberry Pi, but if the mode is MIDI playback, the MSP432 will return to the idle state. If the mode is "Simon Says", the MSP432 will listen for user input until the time is up. If the user input matches the song played by the xylophone, it will mark it as correct and move on to the next level if the user requests so. If the user input does not match the song played by the xylophone, the MSP432 will return to its idle state. If there are no more levels to play, the MSP432 will return to the idle state.

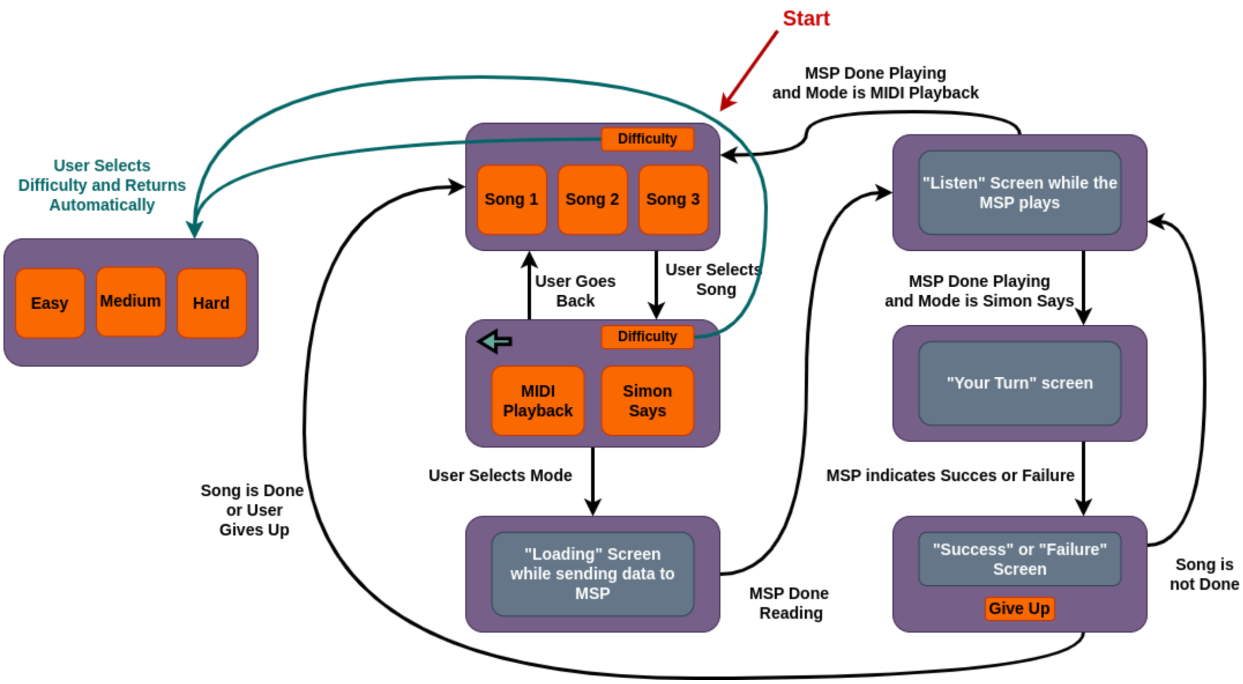

Figure 8 UI Flow for the Raspberry Pi 3B+

Figure 8 represents the UI flow for the Raspberry Pi 3B+. An interaction starts with the user selecting a mode ("Simon Says" or MIDI playback) and then a song. The MIDI files are stored on the RPi (although for a MIDI playback a user can also insert a flash drive with MIDI files on it). When the user selects a song, the MIDI data for the song is parsed by the C++ backend and a stripped-down version of the data is sent via SPI to the MSP432 while the UI shows a “loading” screen. Then, the UI polls the MSP432 for its status while showing a “playing” screen until the MSP432 indicates that it has finished playing. Then, if the user is playing "Simon Says" the UI will poll the MSP432 status again while showing a “playing” screen. Otherwise, it will return to the Mode selection screen. Once the user finishes playing in "Simon Says" mode, the MSP432 will indicate a status of either Success or Failure and the UI will show a matching screen. The user can also use the UI to select a difficulty mode for "Simon Says", which will change which songs the user can choose to play and will also affect the way that the device behaves.

Technical details

The first core feature of the Simophone is being able to store and play MIDI files. To do this, 32 individual solenoids will be placed under each bar of the xylophone and will be actuated to strike whatever note needs to be played from the MIDI file. The solenoids that will be used for this project can be seen in [3]. From the datasheet, the selected solenoid has a DC resistance of around 40 Ohms and needs around 300mA of startup current to produce the rated 5N of force. This means that the power supply to drive the solenoids will be 12 volts. To hold the solenoids in place underneath the xylophone, 80/20 aluminum extrusion will be used. In this case, the extrusion will be mounted under the xylophone and 32 custom 3D printed parts will be produced to mount each solenoid to its respective key. An example of how the solenoids will be mounted can be seen in Figure 9 below.

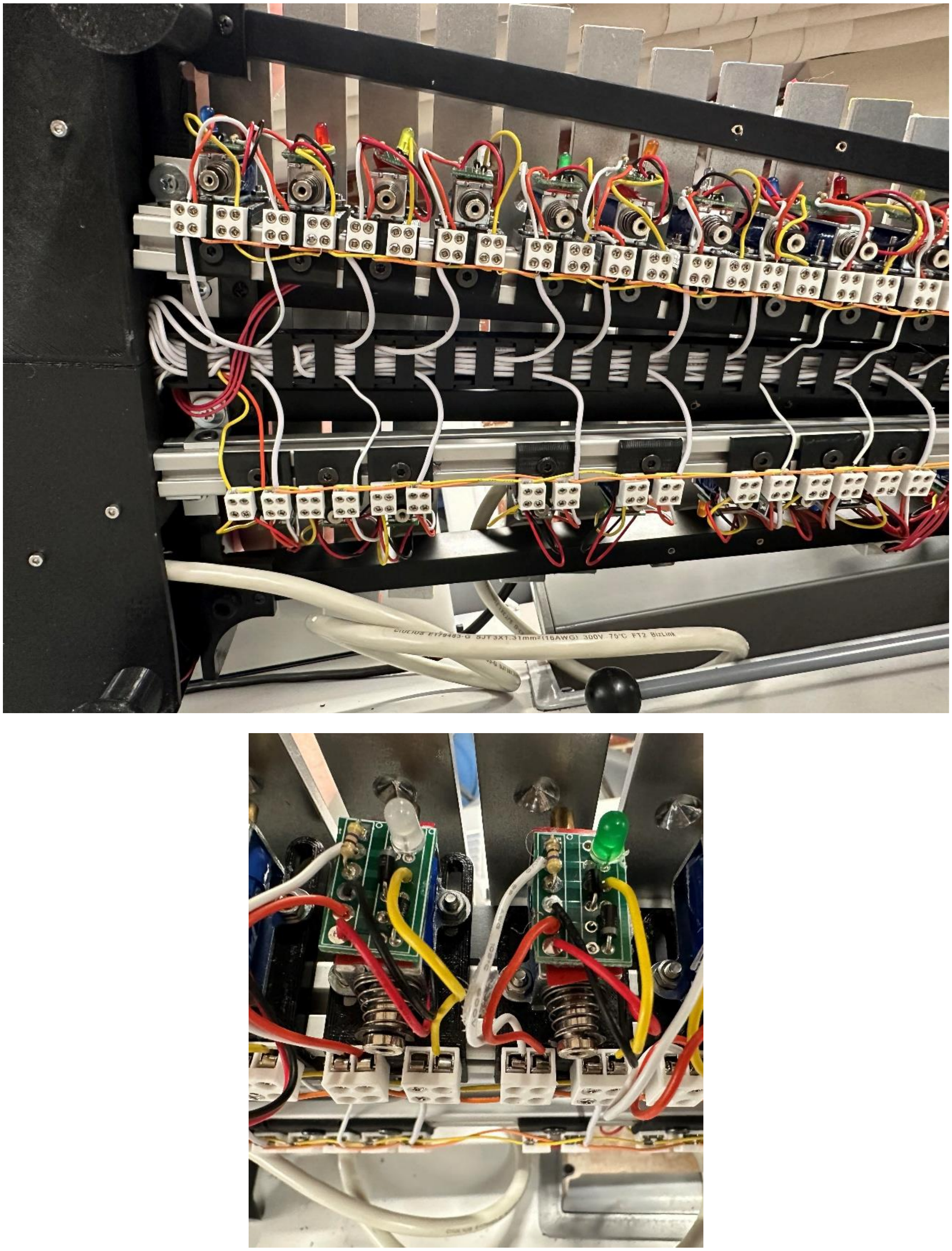

Figure 9 Example Mount for the Solenoids

To actuate the solenoids at the correct time to play a MIDI file, an MSP432 will be used to control each solenoid individually via the GPIO pins. That is, the MSP will activate the pin controlling the solenoid corresponding to the correct note at the correct time according to the MIDI file. This will require at least 32 GPIO pins which is definitely covered by the MSP as it has 48 GPIO pins with wake-up and interrupt capability [4] However, since a GPIO is not capable of driving a solenoid by itself, an NPN transistor will be used to control the current through the solenoid with the GPIO pin controlling the voltage on the base of the transistor. The figure below shows the entire system.

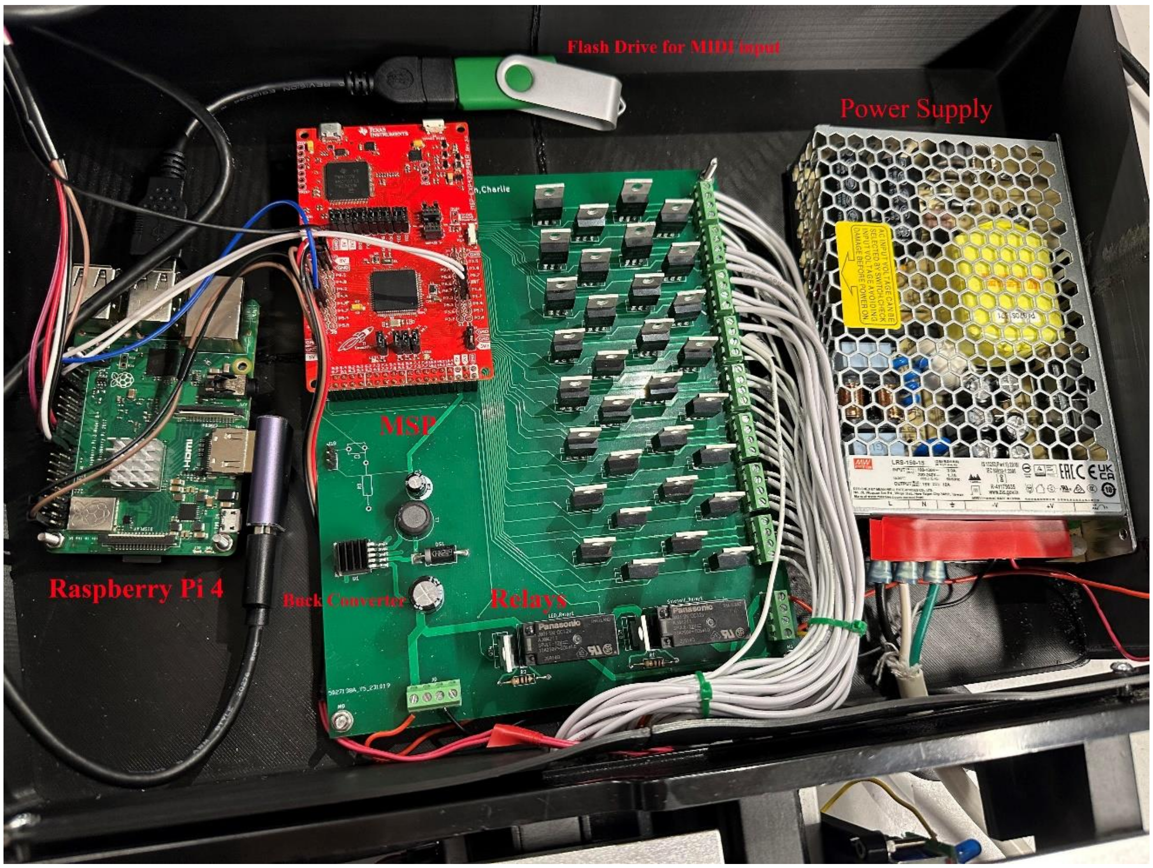

Figure 10   Full System components

A Darlington NPN transistor was chosen because of its high current gain and low cost. High current gain allows great control of the load current through the solenoid with minimal loading to the GPIO pins. From the datasheet as seen in [5], a collector current of 300mA corresponds to a DC current gain of around 2000. This means that the required base current from the MSP IO pin would be around 150 micro amps which is well within the ratings of the GPIO pin as seen in section 5.25.6.2 of the datasheet for the MSP [4].

To further enhance the user experience, different colored LEDs will be placed underneath each bar and will light up when the corresponding solenoid strikes its bar. This is done to aid the user in remembering which bars were played to help them associate each sound to each bar. Also, to provide additional customizability, the note flashing LEDs can be turned off by the user's choice if they feel that they do not need them. To control whether the note flashing LEDs are on or off, a current controlled switch or a relay will be used to control the power to the LEDs.

The relays need to be able to handle up to 8 solenoids worth of current. At 16 volts and that the solenoids have a resistance of 40 ohms, as seen in [3], the current due to one solenoid firing is around 400mA. This means that 8 solenoids firing at once yields around 3.2A. Given this, a 10A relay was chosen as seen in [20].

The overall power consumption is calculated for the extremes of 8 solenoids playing at the same time. From the circuit in Figure 4, given that the rail voltage is 16 volts, the voltage drop across the BJT is .7 volts, and the voltage drop across the diode for the series diode and LED is .7 volts as well, the power can be found as follows:

$$P = 8 \cdot 16 \left(\frac{14.6v}{40\Omega} + \frac{14.6v}{470\Omega} \right) = 51.2W$$

Given this, the power supply was chosen as seen in [19] because of its low cost and large amount of available power.

Next, to determine whether the user played the correct set of notes, MIDI data will be sent over from the Raspberry Pi and will be the expected output for "Simon Says". This MIDI data will be played first then the user is expected to respond with the same set of notes when the xylophone has finished playing. When the user is playing, the microphone will record the user's input until the song is expected to end. The microphone used has a configurable preamp and is shown in the figure below.

Figure 11 Microphone Connected to the Xylophone

To be able to analyze the users input, the Fast Fourier transform (FFT) will be used. We utilized the CMSIS-DSP library for ARM processors to compute the FFT. When it is the user's turn to play the xylophone, the microphone starts sampling at a rate of 40kHz and then takes the FFT of every 51-millisecond window. The equation from converting frequency bins to frequency is shown below.

$$f = \frac{F_s k}{N}$$

The sampling rate of 40kHz was used because to determine some of the higher notes, frequencies of around 19 kHz were used to help detect the note. The 2048 sample window was chosen specifically because it gave us enough frequency resolution (~19.5 Hz) to distinguish different notes on the xylophone but also small enough to ensure we do not run out of memory on the MSP432. As a result, a 2048 buffer would result in 2048 samples/(40000 samples/second) = 51.2 millisecond window. Using the results of the FFT, the window is then analyzed to see if there were any notes on the xylophone were detected. A variation of the Harmonic Product Spectrum (HPS) was used to detect notes [16]. The note-detection algorithm first checks if the magnitude of the fundamental frequency exceeds a certain threshold. If the magnitude does exceed the threshold, it then takes the product of each note's relevant harmonics, which were determined analytically and experimentally, and checks if the result product exceeds a certain threshold. After the note is detected, a timestamp is placed for each note played. This means that the Simophone can determine what frequencies were played at a given time to enable the comparison of the expected MIDI data and the recorded sample.

The system is controlled via a user interface shown on an LCD touchscreen controlled by a Raspberry Pi 3B+ running Raspbian Bookworm. The Raspberry Pi (RPi) communicates with both the MSP432 and the touchscreen using the Serial Peripheral Interface (SPI) protocol, using separate SPI pins for each rather than a multi-secondary setup using a chip-select signal. The user interface was written using the Qt 6.6 framework with a C++ backend as this allowed for a very high-quality production with a relatively simple and readable codebase that could be run both on the RPi and on our laptops for testing. Qt also provides seamless integration with C++ which made it ideal for handling low-level interactions such as SPI. The downside to using Qt is that it required cross-compiling the framework from source using the libraries from the RPi. However, once this was accomplished, this proved to be an effective and efficient tool.

The user interface is executed on the RPi by setting the Raspbian OS to run in console mode, and then starting a standalone X11 server (with the UI binary running as a subprocess). The X11 server is run as a system service, which allows it to be controlled and monitored via the systemctl CLI and to be automatically restarted on failure or power cycling. The udisksd service along with the udiskie wrapper program controls auto-mounting of USB flash drives. A Bash script running once per minute then creates symbolic links to all MIDI files on the flash drive to allow the user to provide their own songs to play. The bash script also cleans up any symbolic links that point to files on flash drives that have been removed.

Lastly, for more hardware requirements, most of the circuit seen in Figure 4 will be built on a PCB. That is, the only things that are not on a PCB will be each LED under the keys and the solenoids themselves. This will allow for a much better physical connection between the bases of the transistors and the GPIO pins of the MSP. Likewise, header pins will be placed on the PCB so the MSP can be placed directly on the board. The PCB layout can be seen in Figure 12 below.

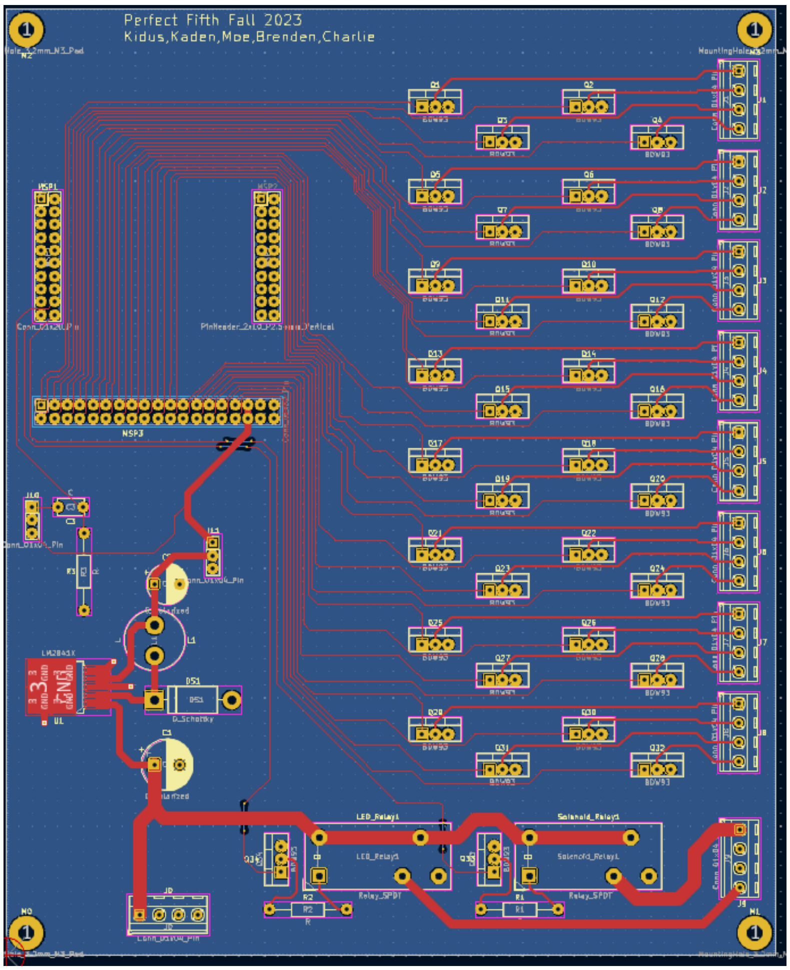

Figure 12 PCB Layout

Testing Plan

To ensure that testing the overall system goes as smoothly as possible, the project should be broken up into defined categories and each category should be tested individually. The project categories will be defined into three parts: hardware, embedded software, and user interface. The hardware consists of the PCB which holds the components for the solenoid and LED drivers and header pins for the MSP and Raspberry Pi. The embedded software consists of the code for the MSP used to actuate the solenoid and LED drivers. The user interface consists of the Raspberry Pi code used to control the LCD and store the MIDI files. The MSP and Pi will communicate via

SPI. The hardware will be tested by breadboarding the entire system and slowly transferring it to the PCB. The PCB itself will contain test points for the collector and base voltages of the BJTs and on the signal lines for SPI between the Pi and the MSP. For the software, a similar approach will be taken where it will be completed in chunks and tested as it is completed.

Resource and Equipment

For most of the electrical hardware design and testing of the project, a multimeter, a DC power supply, and the AD2 will be sufficient. All of which our group already has. For assembly, some power tools will be required to drill holes in things like the aluminum extrusion. Furthermore, a way to cut the extrusion and some woodworking tools may be required. Some of our members already have things like saws, drills, and woodworking tools which can be used. Other than what was listed, no specialized equipment is required.

Challenges

One substantial challenge was the wiring and assembly, because the solenoids and LEDs must be positioned across both rows of keys. There is one wire for each of our 32 keys (ground, solenoid power, and LED power), plus the two power rails, yielding 34 wires feeding back to the control box. We had to find a method to safely run these. Each of these wires will carry less than an amp of current, so we were able to use PVC insulated 24 AWG or larger [10]. We also had to be careful that we avoid overheating our wires by bunching too many of them together, as this could lead to the insulation melting, and then a short circuit. Thankfully, current does not flow through the wires for long enough on a key strike to produce a significant enough amount of heat to not be dissipated immediately.

Another substantial challenge was the audio processing for "Simon Says". Our initial approach was to record both the xylophone and the user playing and compare the spectrograms of both recordings. This initial approach was unfeasible due to both memory and processing power limitations of the MSP432. As a result, we had to change our approach when doing our frequency analysis of the recording. After contemplating several different approaches, we decided to continuously sample whilst the user is playing and take an FFT of every 50-millisecond window. Our first attempted implementation of this approach would be to use the FFT's results and take the two highest magnitude frequencies. It would then map these frequencies to notes, using bins. This method did not end up working, because some of the lower notes on the xylophone produce a high frequency resonance that exceeded the magnitude of the fundamental frequency. These detected pitches would then map to bins other than the fundamental. We then experimentally measured the frequency spectrum for each note. We noticed that some of the resonant frequencies were consistent when a note is played, and some others were unique to specific notes. We used this to our advantage and measured the "relevant" harmonics by checking if they passed a certain threshold. If the relevant harmonics contained the consistent harmonics of a note, it would map it to that note. Otherwise, it would check if it contained the inconsistent harmonics of a note and mapped it to that note. Lastly, if it contained neither the consistent harmonics nor inconsistent harmonics of any note it would not detect any note. The performance of this algorithm was decent, but we had some issues between distinguishing G5, G6, and C8 because they share much too similar of these resonant frequencies. We did more research on note detection algorithms to see if there were any that could pick up two notes at the same time, and eventually came across the harmonic product spectrum (HPS) method which we used in our final product. Verifying the HPS necessitated a lot of testing to make sure the thresholds were correct to avoid not picking up notes or aliasing to a different note.

Timeline

Task	Category	Assigned to	Progress	Start	Due
Initial fill out of Gantt Chart	admin	brendan	100%	9/4/2023	9/6/2023
PCB UltiBoard Design	PCB	mo	0%	9/26/2023	10/3/2023
Component Selection	solenoid system	kaden	95%	8/28/2023	9/26/2023
Component Selection	power supply	brendan	95%	8/28/2023	9/26/2023
Component Selection	signal processing (hard)	mo	100%	8/28/2023	9/26/2023
Design main control architecture	control logic (soft)	brendan	100%	9/7/2023	9/14/2023
Solenoid driver from lookup table	solenoid driver (soft)	kidus	5%	9/4/2023	9/18/2023
SPI Communication with PI	song input (soft)	kidus	5%	9/11/2023	9/24/2023
Play MIDI from program mem	song input (soft)	brendan	0%	9/18/2023	10/9/2023
Read MIDI from PI	song input (soft)	kidus	0%	10/9/2023	10/23/2023
Play MIDI from PI	song input (soft)	kidus	0%	10/23/2023	10/27/2023
Develop STFT implementation	signal processing (soft)	kidus	50%	9/22/2023	10/6/2023
write STFT comparison algorithm	signal processing (soft)	brendan	0%	10/7/2023	10/19/2023
Digital Filtering	signal processing (soft)	charlie	0%	9/14/2023	9/21/2023
Utilize PI Screen (possibly)	screen (soft)	brendan	0%	11/1/2023	11/15/2023
			0%		
Xylophone frame	assembly	mo	25%	10/1/2023	10/15/2023
Solenoid mounts	assembly	kaden	15%	10/16/2023	10/23/2023
PCB and Microcontroller mounts	assembly	charlie	0%	10/24/2023	11/4/2023
Wiring	assembly	charlie	5%	11/4/2023	11/8/2023
Soldering	assembly	mo	0%	10/24/2023	10/31/2023

Figure 13 Initial Gantt Chart

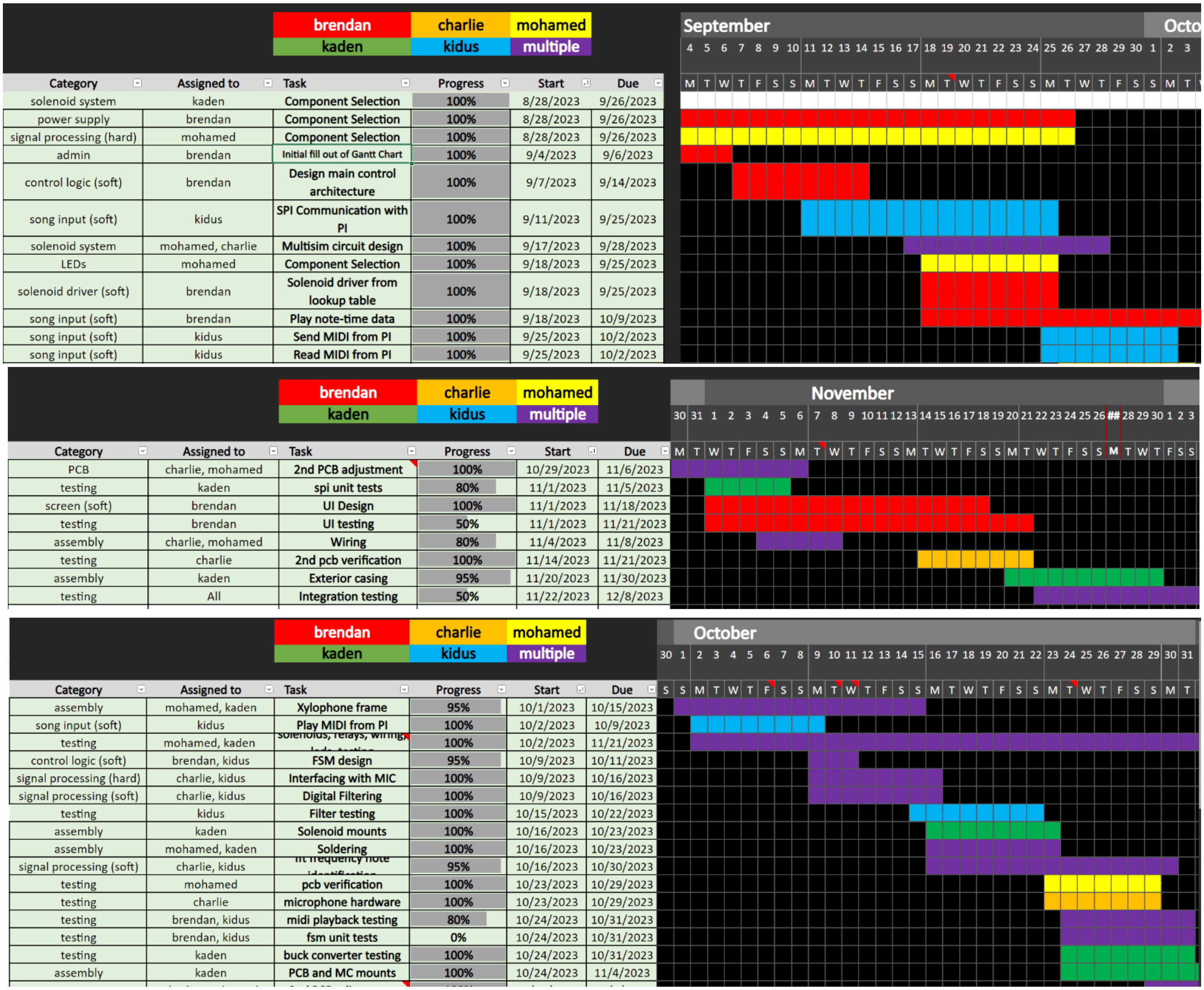

Figure 14 Final Gantt Chart As of 11/27/23

The evolution of project management techniques can be clearly seen through the comparison of the initial Gantt chart created at the beginning of the semester with the revised version drafted towards the end of the school engineering project. Initially, the Gantt chart was aspirational yet lacking in the precision required for effective project management. The initial timeline suffered from an optimistic estimation of task durations, an absence of essential tasks, and a general lack of detailed planning. Furthermore, the task assignments to the five-person team were too vague to offer any actionable guidance or accountability.

As the project unfolded, it became evident that a more sophisticated approach was needed. The revised Gantt chart reflects a significant improvement in project organization and clarity. Assigning colors to each team member’s tasks not only improved visual organization but also heightened individual accountability. By defining tasks sequentially and integrating meticulous details, the team established a more realistic and coherent workflow.

Adjustments to the timeline became necessary due to unforeseen circumstances such as shipping delays, equipment malfunctions, and the integration of new ideas and tasks, which are common challenges in engineering projects. Some tasks proved to be more complex than initially anticipated, prompting a reassessment of the time required to complete them. The team's ability to adapt to these changes was crucial. Flexibility, a key trait of any successful team, was embodied in the collaborative spirit demonstrated when individual tasks evolved into collective efforts. This dynamic approach ensured that obstacles were efficiently managed and that team members could pivot to support others or take on different tasks when faced with a dead end.

The software development aspect, involving the user interface (UI) and game features like "Simon Says", naturally consumed a significant portion of the project time. This was anticipated, given the iterative and unpredictable nature of software development, which often requires numerous revisions and testing cycles. Consequently, a substantial portion of the remaining semester has been allocated to this phase, acknowledging that it is subject to ongoing changes and refinements.

Despite the challenges and the need to extend some deadlines due to external factors, all tasks have been completed on the revised schedule. This is a testament to the team's capacity for effective time management, and their ability to remain productive and progress despite obstacles. The experience illustrates a crucial lesson in project management: the importance of adaptability, continuous reassessment, and the willingness to recalibrate plans in response to the realities of project execution. It highlights the inherent value in starting with a flexible plan and the readiness to iterate and improve upon it as the project evolves. This approach ensures that even when deviations occur, they can be managed without jeopardizing the overall project timeline or success

Costs

Index	Item	Manufacturer Part #	Supplier #	Part number #	Qty in Stock	Qty Req'd	Qty purchased	Per Unit Price	Cost	10000 unit Qty	Bulk per unit price	10000 unit cost	Savings for 10000 units
1	Fuses	0617004.MXP	DigiKey	0617004.MXP-ND	2,056	1	5	0.24	1.2	50000	0.10	5151.0	57%
2	Fuse Clips	01110005MR	DigiKey	F043CT-ND	66,100	1	4	0.36	1.44	40000	0.14	5646.0	61%
3	16 v power supply	LRS-150-15	Mouser	709-LRS150-15	178	1	1	24.09	24.09	10000	23.80	237970.0	1%
4	Diodes	1N4004	Mouser	637-1N4004	112,767	32	50	0.125	6.25	500000	0.07	37000.0	41%
5	16 AWG terminal connectors	52934	DigiKey	A28166-ND	3,419	4	10	0.844	8.44	100000	0.51	50574.0	40%
6	solenoids	412	Mouser	485-412	238	32	32	7.5	240	320000	4.00	1280000.0	47%
7	Orange T1 LEDs	HLMP-Y402-G0000	Mouser	30-HLMP-Y402-G000	24,418	10	10	0.273	2.73	100000	0.08	7984.0	71%
8	Purple T1 LEDs	NTE30120	DigiKey	2368-NTE30120-ND	1200	10	10	0.76	7.6	100000	0.63	63000.0	17%
9	4x1 terminal blocks	EBWA-04-B	DigiKey	2057-EBWA-04-B-ND	1,990	10	25	0.4316	10.79	250000	0.21	53732.5	50%
10	470 ohm 1/2 W resistor	CFM12JT470R	DigiKey	S470HCT-ND	14507	32	50	0.0372	1.86	500000	0.01	3835.0	79%
11	Darlington Pair Transistor	BDW94C	Mouser	512-BDW94C	17512	32	50	0.97	48.5	500000	0.36	180035.0	63%
12	Microphone	CMEJ-0415-42-LP	Mouser	490-CMEJ-0415-42-LP	9626	1	5	0.43	2.15	50000	0.22	11000.0	49%
13	Buck Converter (Diode + Cap)	DFR0883	DigiKey	426-DFR0883	788	1	1	43	43	10000	30.00	300000.0	30%
14	Buck Converter (usb connections)	DFR0884	Mouser	426-DFR0884	22	1	2	8.5	17	20000	8.50	170000.0	0%
15	Control 0.5 A BJT	KSP92BU	Mouser	512-KSP92BU	27325	34	100	0.087	8.7	1000000	0.06	55270.0	36%
16	Relays	255-4059-5-ND	Digikey	255-4059-5-ND	98	2	2	6.33	12.66	20000	5.00	100000.0	21%
17	Wires	NA	Amazon	NA	48	1	1	12	12	10000	12.00	120000.0	0%
18	Connectors	NA	Amazon	NA	76	64	100	0.3	30	1000000	0.30	300000.0	0%
19	Mini-PCB	NA	Amazon	NA	200	1	1	11	11	10000	11.00	110000.0	0%
20	PCB manufacturing	NA	JLCpcb	NA	NA	1	1	37	37	10000	4.00	40000.0	89%
								Total cost	489.41		10000 unit Total Cost	3091197.5	
											Total Saving	37%	

Free Items	Source	Market Cost $
Xylophone	Kaden	130
LEDs	LabKit	7
3Dprint material	Kaden	20
MSP432 X2	Professor Delong	60
Raspberry pi	Charlie	70
	Total	287

Figure 15 cost breakdown

Operating under a tight budget of $500 necessitates creative and strategic cost management for the self-playing xylophone project. Luckily, we were able to manage our budget efficiently and stay under the given $500. The project's total cost is 489.41$ as shown in the table above.

The pursuit of cost-effectiveness began with securing complimentary components wherever feasible, which is a smart initial move to mitigate expenses. We were able to acquire several key items, such as the xylophone, MSP432, and Raspberry Pi, at no cost. As indicated, these items would have constituted a substantial portion of the budget, and their exclusion from the cost calculations offers considerable savings. These items would usually cost us around $287 as listed in the table above.

Despite these savings, the solenoids, essential for the instrument's automation, remain a substantial cost factor, consuming half of the budget. With each solenoid priced at $7.5 and a requirement for 32 to cover all xylophone keys, the cost quickly adds up to $240. The project's nature, demanding a solenoid for each key, inherently escalates the expense, underscoring the challenge in budget allocation when dealing with multiple identical components.

To alleviate financial strain, sourcing less expensive solenoids was considered. However, this approach reached a dead end as the lower-cost alternatives failed to meet the necessary force specifications to strike the xylophone keys effectively. Reliability and performance are paramount, and compromising these for cost could jeopardize the project's success. Consequently, the decision to proceed with the more costly but dependable solenoids was a calculated one.

Exploring bulk purchases presents another viable strategy for cost reduction. As detailed in the provided table above, scaling up production to 10,000 units could yield a budget saving of 37%. Mouser and Digikey offer lower prices for bulk purchases. The savings per item vary from 0% to 80%. items from Amazon do not offer any bulk deals. Some Digikey items have low deals due to limited supply. Another thing we must keep in mind is if the amount required is in stock or not given the shortage in such field.

The mass production is not limited to acquiring the components. Manufacturing and assembly can be costly leading to an expensive product that out of the public’s budget. The end goal is to produce a cost-effective product that aligns with consumer budgets, thus making it accessible to a wider audience. Automation stands out as a promising solution for reducing both the time and cost associated with manufacturing, thereby addressing one of the primary cost drivers of the product.

Innovative manufacturing techniques, such as 3D printing, can be harnessed to create more efficient assembly processes. For instance, the solenoid mounts, typically one of the more time-consuming aspects of assembly due to their quantity and precision required, could be designed to be 3D printed in a connected chain. This would allow for the simultaneous printing of multiple mounts, dramatically reducing the time and labor needed when compared to traditional manufacturing methods.

Furthering this approach, the xylophone's casing itself could be optimized for 3D printing, ensuring that all components fit seamlessly within a single, cohesive design. By creating a design where solenoids can simply slide into a preformed block, the assembly process is significantly streamlined. This not only speeds up production but also minimizes the potential for assembly errors, as the design inherently guides the correct placement of parts.

The automation of wiring is another area ripe for innovation. Traditionally a manual and tedious task, modern automated wiring solutions could pre-cut, strip, and route wires throughout the device with precision and speed, far surpassing what is achievable by hand. Such automation could be integrated into the manufacturing process, cutting down on both time and potential for human error.

However, the transition to automated production methods requires an initial investment in machinery and design software capable of supporting these advanced techniques.

Final Results

To begin, our first success criterion of arbitrary MIDI file playback was fulfilled. The Simophone is able to play any single-channel MIDI file that we load onto a thumb drive. If the song has more than one channel, the Simophone will still function, but only play the first as is intended. There is one slight limitation to MIDI files being played, being that they cannot contain more than a few hundred notes, due to memory restrictions on the MSP432. Typically, this is a non-issue, as we've determined most songs tend to be either short enough, or sparse enough with their notes, to not reach the limit.

As for our "Simon Says" game and its success criteria, we were also successful. A user is able to play the game, with the Simophone successfully determining the notes being played by the user, comparing them to the notes previously played by the device, and determining if what the user played constitutes a win or a loss. We were able to implement the "Easy," "Medium," and "Hard" difficulties; the user can typically win each mode at a rate higher than the lower limits expected. These lower limits were a 5% win rate for "Hard," a 25% win rate for "Medium," and a 50% win rate for "Easy."

In addition to meeting the previously defined success criteria, the Simophone has additional functionality and is a more robust system than originally planned. Our "Simon Says" game includes both a metronome-type rhythm to get the user started, and an additional "Win" or "Loss" jingle, to better alert the user to the status of the game after playing. The Simophone is also fully self-contained, with an enclosure for our microcontrollers, power supply, PCB, and screen. Our device is a single piece which is portable and able to plug into a wall socket and play anywhere. Overall, we feel quite satisfied with ourselves and the quality of device we were able to achieve.

Future Works

Reflecting on the development of the "Simophone" project, it's important to acknowledge the constraints and challenges we faced, particularly the limited timeframe of just one semester. This period, while sufficient to lay the groundwork and bring the initial concept to life, did not allow for the exploration and implementation of all potential enhancements and refinements. There are numerous avenues for enhancements and innovations that future teams can explore, taking the Simophone from a promising prototype to a fully realized educational and musical instrument. Some of those possible improvements are:

Enhanced Display and User Interface

One significant improvement could be the incorporation of a larger screen for displaying key notes and guidance. This would enhance the learning experience by making it easier for users to follow along with the music and understand the fundamentals of playing the xylophone.

Additionally, integrating uploaded instrument courses onto this screen could provide a structured, self-paced learning environment, benefiting beginners and advanced players alike.

Internet Connectivity and Expanded Resources: Equipping the Simophone with internet access and a search engine would be a major leap forward. This feature would allow users to look up new songs and access a wealth of online musical resources, significantly expanding the instrument's repertoire and educational value. It encourages continuous learning and exploration, as users can easily find songs that match their interest or skill level.

Remote Control via Mobile App

Developing a mobile application for remotely controlling the xylophone could add a layer of convenience and flexibility. This app could enable users to play songs for entertainment, choose learning modules, or even control the Simophone during performances. The app could also serve as a platform for updating the instrument's software, accessing additional content, or customizing its settings.

Voice Control Integration

Incorporating voice command capabilities could make the Simophone more accessible and easier to use, especially for individuals with limited mobility or those who prefer hands-free operation. This could involve simple commands for playing songs, adjusting settings, or accessing educational content.

Interactive Gaming Enhancements

Improving the "Simon Says" game to record player scores and names would introduce a competitive element, making the learning process more engaging and fun. The addition of a two-player feature would further enrich the experience, promoting social interaction and collaborative learning. These gaming aspects could make the Simophone particularly appealing in educational settings or as a family entertainment device.

Portability and Design Modifications

Making the Simophone portable by downsizing its dimensions and adding a battery would greatly enhance its usability, especially for children. A portable version could be used in various settings, from classrooms to outdoor activities, making music education more accessible and versatile. Additionally, optimizing the design by relocating circuits underneath the xylophone could help reduce its overall size, making it more practical for everyday use.

Advanced Feedback System

Enhancing the feedback mechanism to provide more specific guidance, such as indicating missed notes, tempo issues, or difficulties in playing multiple keys simultaneously, would significantly improve the educational aspect of the Simophone. This level of detailed feedback would aid users in identifying and working on their weaknesses, leading to a more effective and personalized learning experience.

Hardware Improvements

Shifting the circuits under the xylophone could not only streamline the design but also potentially enhance the instrument's durability and safety. This repositioning might allow for a more robust and user-friendly design, particularly important when the Simophone is used by children or in educational settings.

Sustainable and Eco-Friendly Materials

In the manufacturing of the Simophone, using sustainable materials and environmentally friendly processes can appeal to eco-conscious consumers and institutions. This could include recyclable components, non-toxic finishes, and energy-efficient electronics.

These enhancements focus on making the Simophone more user-friendly, educational, and enjoyable, potentially broadening its appeal to a wider range of users and settings, from individual hobbyists to educational institutions.

Each of these improvements, while beneficial, brings its own set of challenges. Future teams should approach these enhancements with a balanced perspective, considering the trade-offs between functionality, cost, usability, and technical complexity. Careful planning and testing, along with feedback from users, will be crucial in navigating these pitfalls successfully. One of the difficulties that our team was not expecting is the failing hardware. It is easy to assume everything bought new from a certified seller will work and live up to the specs. A good percentage of our purchased equipment and components did not function as expected leading to delays and more costs. Therefore, future engineers building up on similar projects should keep in mind to use high-end components to guarantee a functional project.

References

[1] R. Román-Caballero, M. A. Vadillo, L. J. Trainor, and J. Lupiáñez, “Please don’t stop the music: A meta-analysis of the cognitive and academic benefits of instrumental musical training in childhood and adolescence,” *Educational Research Review*, vol. 35, p. 100436, Feb. 2022, doi: 10.1016/j.edurev.2022.100436.

[2] Tokyo Steve, “Automated MIDI Xylophone,” *Instructables*. https://www.instructables.com/Automated-MIDI-Xylophone/

[3] “Small Push-Pull Solenoid -12VDC.” Accessed: Sep. 20, 2023. [Online]. Available: https://www.mouser.com/datasheet/2/737/Adafruit_05132020_412-1858431.pdf

[4] “MSP432P4011 pdf, MSP432P4011 Description, MSP432P4011 Datasheet, MSP432P4011 view::: ALLDATASHEET:: ”:, *pdf1.alldatasheet.com*. https://pdf1.alldatasheet.com/datasheet-pdf/view/1004561/TI1/MSP432P4011.html

[5] “To learn more about ON Semiconductor, please visit our website at www.onsemi.com.” Available: https://www.onsemi.com/pdf/datasheet/bdw94c-d.pdf

[6] Uctronics, https://www.uctronics.com/download/Amazon/B0106.pdf

[7] Microcomputer controlled game, by H. J. Morrison, R. H. Baer. (1977, Sep. 19). *US4207087A*. [Online]. Available: https://patents.google.com/patent/US4207087A/en

[8] Player piano, by M. A. Slaats, R. T. West. (1964, Dec. 11). *US3472110A*. [Online]. Available: https://patents.google.com/patent/US3472110A/en

[9] Musical Instrument, by R. W. Crawford. (1958, Feb. 17). *US2992511A*. [Online]. Available: https://patents.google.com/patent/US2992511A/en

[10] “AWG Wire Gauges Current Ratings,” *www.engineeringtoolbox.com*. https://www.engineeringtoolbox.com/wire-gauges-d_419.html

[11] National Electrical Code, NFPA-70, 2023

[12] Standard for Determining Current Carrying Capacity in Printed Board Design, IPC-2152, September 16 2009

[13] Temporary worker Initiative - Occupational Safety and Health Administration, https://www.osha.gov/sites/default/files/publications/OSHA3953.pdf (accessed Dec. 5, 2023).

[14] “UL 510,” UL 510: UL Standard for Safety Polyvinyl Chloride, Polyethylene, and Rubber Insulating Tape, https://global.ihs.com/doc_detail.cfm?document_name=UL+510&item_s_key=00097130

[15] CMSIS DSP Software Library, https://www.keil.com/pack/doc/CMSIS/DSP/html/index.html

[16] Pitch estimation using harmonic product spectrum derived ... - IEEE xplore, https://ieeexplore.ieee.org/abstract/document/6718976/

[17] "Raspberry Pi 3 Model B+." Nov. 2023. https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-b-plus-product-brief.pdf

[18] "LM2596 simple Switcher® power converter 150-KHz 3-a step-down voltage," https://www.ti.com/lit/ds/symlink/lm2596.pdf

[19] "150W Single Output Switching Power Supply," LRS-150-15. Mar. 3, 2023. https://www.meanwell.com/Upload/PDF/LRS-150/LRS-150-SPEC.PDF

[20] "JW Relays," JW1SFN. Jul. 2023. https://www.digikey.com/en/products/detail/panasonic-electric-works/JW1FSN-DC12V/251886