# Pitch Controlled Pong

A Technical Report submitted to the 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

**Charles Hess**

Fall, 2022

Technical Project Team Members

Isaac Duke

Edward Oline

John Phillips

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

Harry Powell, Department of Electrical and Computer Engineering

# The Lizard People - Pitch Controlled Pong

*Teddy Oline, Charlie Hess, Isaac Duke, John Phillips*

December 13, 2022

**Capstone Design ECE 4440 / ECE4991**

**Signatures**

Teddy Oline

Charles Hess

Issac Duke

John Phillips

## Statement of work:

*Isaac Duke:*

Isaac was the software lead and wrote the vast majority of the code for the whole project with a little help from Teddy and John. This included all game logic and functionality including a finite state machine to track what stage of the game the player is in. This also included the code required for the various operations in the digital signal processing such as cepstral analysis. Isaac also was able to implement multithreading allowing the audio processing and game code to run on separate CPU cores. Additionally, he did the setup of the Raspberry Pi which included a custom version of Linux and installed all dependencies and libraries needed for handling the audio signals.

*Charlie Hess:*

Charlie's primary responsibilities included the analytical development of the analog signal processing system. This includes component selection, filter design, and the development of the surrounding circuitry to support the integrated circuit chips selected for ADC and amplification. Charlie completed the analysis for the selection of our current amplifier, designed preliminary support circuitry for the amplifier and ACD, and worked alongside Teddy and John for our current filter design and ADC selection. Charlie's secondary responsibilities include schematic design and PCB layout, implementing the analytical design in KiCad for eventual production on a printed board. Charlie also completed the PCB layout in KiCad, and handled its submission and assembly with assistance from John and Teddy.

*Teddy Oline:*

As part of the hardware team with Charlie and John, Teddy aided in the development of the PCB and its design. This involved both component selection and schematic design. He was primarily responsible for the ADC and the surrounding circuitry required for operation. Additionally, Teddy was the secondary programmer for the team behind Isaac and wrote some auxiliary code for the game including debouncing the button on the machine, adjusting the display output, and helping integrate the digital signal processing of the voice input.

*John Phillips:*

Along with Teddy and Charlie, John initially assisted with PCB development. Tasks included selection of professional grade connections between the case and the PCB as well as sourcing footprints and symbols for the KiCad schematic and layout. Additionally, John took the lead role in the manufacturing of the aluminum console that housed the game. This required use of CAD to communicate machining needs with experienced machinists. Machining the console also required that John get certified in the Mechanical Engineering Machine Shop through a 3-class safety training.

## Table of Contents

## Table of Figures

## Abstract

This project involved constructing a physical single player video game module with vocal controls. The module runs a version of the classic game Pong, but with a twist. The user moves their paddle up and down based on the pitch of their voice. Based on an initial calibration, the user sings a relatively high pitched note to move their paddle up, and a relatively low pitched note to move their paddle down. The opponent is a computer controlled paddle on the opposite side of the screen. The module consists of a small monitor in a display console with an external microphone for the user to vocalize their input into.

## Background

The project was chosen to incorporate a wide variety of electrical engineering, computer engineering, and design skills, while remaining creative, engaging, and fun. As a group, implementing an interactive game that most users could understand and play without background knowledge was a more engaging task than a project in a non-game, or noninteractive format.

Similar projects in the past include a mobile wrist watch video game module [1]. Two iterations of this project featured two games: pong and asteroids. The game was not interactive, however. It displayed gameplay from these two games which corresponded to the current time, so that the device functioned as a wrist watch. It did not accept user input.

Another related past technology is the Atari, on which the video game pong originated [2]. The module utilized joystick input for player movement, and ran the game on a microprocessor in an apparatus which the user would attach to a monitor.

A third related technology is speech recognition. One project implemented voice recognition technology to control classic video games on a microcontroller [3]. The project distilled human speech into features, and used these features for processing and eventual game control.

Our project is separate from these projects primarily through user input. The first similar project, the wrist watch, accepted no user input. Our project will be controlled by a user through recognition of their voice's pitch. The second and third similar projects did accept input, but in a different form. The difference from a joystick input is clear, but the difference between the speech recognition technology arises in how the audio input is used. Our project will not be concerned with the words the user is speaking, but instead it will isolate the primary pitch of their voice. This pitch recognition and utilization as input differentiates our project from these previous works.

This design will require new technical capabilities while drawing on previous coursework. For processing the microphone's input, filtering and amplification will be required. These tasks will draw on filter and amplifier design skills learned in the electrical and computer engineering fundamentals courses: ECE 3750 and ECE 2660 (FUN II and III). Next, a printable circuit board (PCB) will be required to implement the filtering and amplification of the microphone input, drawing on circuit design and PCB layout skills from ECE 2630, 2660, and 3750. Since a microcontroller will be used as the primary processor for implementation of Pong, microcontroller experience and embedded system design skills learned in ECE 3430 will be used. Through programming tasks to develop the functional Pong game on a microcontroller, best programming practices and object oriented programming learned in CS 2110 and 2150 will be used. Finally, project troubleshooting and circuit debugging skills developed during ECE 2630, 2660, and 3750 will be used while testing and troubleshooting each component and the system as it is assembled.

## Physical Constraints

### Design Constraints

Our video game module will be relatively easy to manufacture on a large scale. The PCB is the only thing that needs to be fabricated. The Raspberry Pi, temporarily disregarding procurement concerns detailed in the next section, solely needs to be flashed with the OS and program images. This makes the microcontroller fast and cheap to manufacture, beyond the wholesale cost of the unit and constraints of the manufacturer supply chain. The other components are “off the shelf”, with again the only external constraint here is component availability. Given the limited number of connections in the design plan, this product could be reproduced at a relatively low cost using standard assembly and fabrication equipment.

The main materials needed are the electrical components, printed circuit board, cables and connectors, casing for the module and the external microphone.  In the process of identifying necessary componentry, the state of the global supply chain was at the forefront of thought.  The availability of our selected parts is dependent on the stock of electronics supplier DigiKey Electronics [7] - our primary supplier.  We have identified several acceptable replacements for parts that are low in stock and potentially will be out of stock in the future.  The primary headache revolves around sourcing our selected microcontroller, the Raspberry Pi, through their authorized dealers. This microcontroller is currently in high demand and has limited availability. Both the external microphone casing are available through multiple manufactures and have several acceptable substitutes.

The constraints imposed by the physical device include the processing speed and graphical processing capabilities of the microcontroller. In a single thread it proved impossible for us to create a working version that ran at 60 FPS so multithreading was implemented to account for this. The audio processing was run on one thread while the game logic and graphics were run on another. The microphone also proved to be a constraint, as well as the physical laws pertaining to audio sampling especially when it comes to frequency analysis. We spent significant time tuning the sampling frequency from the microphone, the sampling frequency of the pitch detection algorithm, and the method of calculating pitch from the raw input audio signal.

### Cost Constraints

The primary economic constraints of this project are the $500 budget specified in the project description and current component pricings affected by recent supply chain limitations.  The largest items on our budget include the Raspberry Pi microprocessor as well as the LCD screen, which totaled $167.94 and $75.89 respectively..  Since we only produced a single unit for our project, we did not have significant difficulty keeping within our budget. If this device were to go into large-scale production, economies of scale would decrease the individual unit price down well below the $500 maximum afforded in this project through bulk part pricing and consolidated labor.  The device also requires external power through a Universal Serial Bus (USB) Type-C port, so long term use would incur energy costs on the part of the user.

**Tools Employed**

Raspberry Pi 4 (4 GB RAM): The Raspberry Pi 4 is a single board computer which functions as a microcontroller and the primary processing power for game functionality. It also served to interface with the audio input from our PCB, and produce an audio output. As a tool, it was used in testing software, and generating power from its available 3.3V and 5V pins.  While Isaac took the lead role on the embedded development necessary, both John and Teddy had experience with microcontroller interfacing.

On the software front we used a custom build of Visual Studio Code [8] to write and test code natively on the Raspberry Pi. However before doing so we had to use BuildRoot [9], a tool used for baking custom Linux images, to create a custom operating system image for our Raspberry Pi. This image was then flashed to a micro SD card and loaded onto the Raspberry Pi. The native UART (universal asynchronous receiver-transmitter) port was useful during this stage to manually check data and ensure the installation process was successful.

Many useful software tools and techniques were used to create the finished product. First of which, the use of an FSM, was a technique learned in our embedded programming courses, but the specific application for our project (especially using Python) required some significant additional skill development. Additionally, the creation of the Linux image that we ran our game on required learning BuildRoot as well as what an RTOS (real time operating system) is and how to effectively use one to create a functional product. Several synchronization primitives including multithreading, locking, and data synchronization were used to improve performance and ensure no data loss between threads that were used to handle different tasks.

The circuit schematic and the final PCB layout of the components were designed using the KiCad software [18]. This was the team’s first time using this software as previous projects in the electrical and computer engineering curriculum had used other applications for the PCB design process. In addition, the analytical design work for the PCB was done in part with assistance from the Analog Devices Analog Filter Wizard [19]. Once filter specifications were determined, the tool was used to acquire component values for the Sallen-Key [20] filter that was implemented on the PCB.

The case itself was bought from Hammond Manufacturing [17].  The aluminum case required several holes that were made with a milling machine.  Before getting certified in the Mechanical Engineering Machine Shop, a CAD file was downloaded from Hammond Manufacturing and modified with the extrusions required.  This modification of a CAD file required the use of Autodesk Inventor, a computer-aided design application for 3D mechanical design, simulation, visualization, and documentation.  John had previous experience in CAD and was able to complete this task effectively. After the design was completed, a 3-axis vertical mill was employed to make the cuts.  This was done under the supervision and guidance of the Lead

Engineering Technician responsible for running the Machine Shop. John had experience with rapid prototyping and fabrication. These skills were helpful in the novel domain of machining metal required by our project.

## Societal Impact Constraints

### Environmental Impact

The main environmental impacts this product will have are in relation to its manufacture and disposal. During manufacture, both our custom PCB connector board and the Raspberry Pi will produce the most waste of their lifetime from the subtractive manufacturing methods employed [10]. Recycling the device will be challenging since it will be composed of multiple parts that are not all recyclable. As long as the device is not physically destroyed, the Raspberry Pi is eligible for factory restoration and recycling [11]. Additionally, we have sourced electronics disposal companies that provide free disposal of LCD screens [12]. Otherwise, during its operational lifetime, the game can be replayed with minimal environmental impact during its useful lifetime.

### Sustainability

While this video game console uses electricity, we do not believe this presents an issue from a sustainability perspective. Given the lack of mechanical componentry and battery this product will have limited deterioration and thus an extended lifespan. Raspberry Pi's and PCBs can last 10-30 years if adequately designed and enclosed [11]. The electrical parts, including the screen, microprocessor, PCB, and connectors, could be replaced if there is electrical or physical damage. If this our product was being produced at the volume where sustainability would be pertinent, the product would be manufactured such that skilled labor could effectively replace components. This requires design choices that would allow access to the components, such as using screws to seal the enclosure instead of permanent methods.

### Health and Safety

In order to ensure consumer safety, materials and manufacturing techniques will be employed to protect the electronic devices from possible water contamination. Additionally, a strict following of standards outlined in the proceeding section will ensure the product is not harmful to consumers. Measures including adequate handling of ground faults, shorts, and rated equipment will protect the user in the event of an unexpected electrical malfunction[13]

### Ethical, Social, and Economic Concerns

One concern about this device relates to the disposal of the LCD this module uses. LCD disposal is dangerous and recycling them is expensive and difficult [12]. This leads to a massive export of end of life LCDs to developing countries where environmental and workplace regulations are more relaxed. This is an unfortunate truth that our device could be subject to if not given to an ethical recycler.

This device will be focused on playing a video game with a human opponent versus a computer. This process will not require or benefit from the use of data collection or profiling, therefore, there should be no ethical issues regarding race, gender, ideology and so forth. The only issue may regard the premium nature of the specialized video game console which would put a barrier between the product and poorer individuals who would struggle to justify the cost.

## External Considerations

### External Standards

With the use of electrical equipment and a PCB designed for handheld operation, one of the standards will be to have an enclosure following the NEMA/IEC type 1 standard. This protects the user from an electrical shock and ensures the electrical equipment is free from physical debris, including dust [14].

Both the OS modifications and the video game program will be written in C. Therefore, the Barr C coding standards will be utilized to help minimize bugs and allow for formal standards to facilitate ease of coding and communications [15].

Another standard will be required because of the power supply utilized by the circuit. We will be designing a Class 1 circuit which is defined as having an output power that does not exceed 30V or 1000 volt-amps, and can use a 120V power source from an outlet as preferred by OSHA standards [16]

### Intellectual Property Issues

Patent No. 2611782: PONG [21]
This patent encompasses the usage of the word “pong.” If we were to bring this product to market our current project title “Voice Controlled Pong" would be an infringement of this copyright. However, as game mechanics are unable to be patented, the gameplay experience could remain untouched in a commercial version.

Patent No. D926749: Raspberry Pi [22]
The only patent owned by Raspberry Pi Ltd. is regarding a computer case. We will have no issues with the usage of a Raspberry Pi in our commercial product. However, we cannot use branded trademarks and logos of Raspberry Pi such as the name “Raspberry Pi” and their logo in certain manners. Their website [27] has a detailed description of the allowed usage of their trademark.

1.5

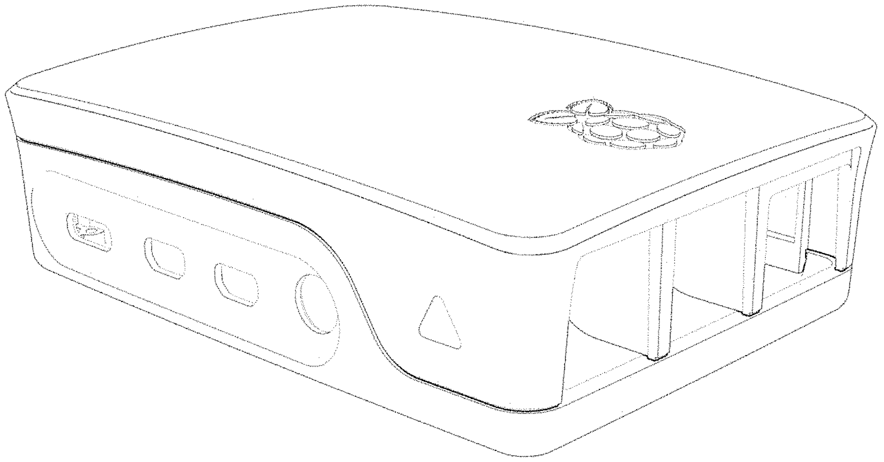


**Figure 1. Raspberry Pi Ltd. Computer Case Patent**

Patent No. US 2004/0138888A1 : Method and Apparatus for Speech Recognition [23]
This intellectual property in this patent discusses cepstral tuning techniques, and leveraging the cepstrum for speech recognition and conversion. The cepstral technique they use is similar to that used within our project, but the object of intellectual property is their method for using the produced cepstral coefficients to recognize speech, which does not pose a problem for the intellectual property of our project.

In summary, we are limited by the copyright rules and claims to the Pong and Raspberry Pi names. This project is likely difficult to patent, as it does not use any of its components in a completely novel way. The Raspberry Pi, monitor, and ICs utilized are fulfilling their intended manufacturing purpose, and thus this integration of the various components may not be considered nonobvious for the purposes of patenting.

## Detailed Technical Description of Project

### PCB Schematic Design

*Summary of Function*

The total function of our PCB was to accept the microphone signal provided by our implemented lapel microphone, and convert it to a decipherable signal for the raspberry pi to use in processing. For this to be done, the function was split into three stages: amplification, filtering, and analog to digital conversion. The small signal from the microphone needed to be amplified, then any frequencies outside of the range of the human voice were to be filtered out, then the signal must be converted from the analog to the digital domain, as the raspberry pi could accept only a digital signal. Finally, the PCB also housed the connection for the pushbutton implemented on the project. The circuitry for each of these functions are detailed in the sections to follow.

*Amplifier*

Amplifier selection was somewhat inhibited by available information for the selected microphone. A lavalier (lapel) microphone was selected for user friendly implementation and flexibility in movement, but a well documented lavalier microphone was not available. However, the lavalier microphone chosen is an electret microphone, thus output impedance was approximated to be 2.2 kOhms, with output voltage in the 20 mV range, as is typical for electret microphones used in this context. An amplifier was then selected on the basis of automatic gain control (AGC) inclusion, input impedance, gain available, and price. The MAX 9814 [24] was then selected, as it includes AGC, has an appropriate input impedance of 100 kOhms, offers three gain selections of 40, 50, and 60 dB, and boasts a price of under $2.20. Standard audio amplification from stipulates a minimum amplifier input impedance of 5-10 times output impedance [28], thus the 45 times ratio the MAX provides should be more than adequate.

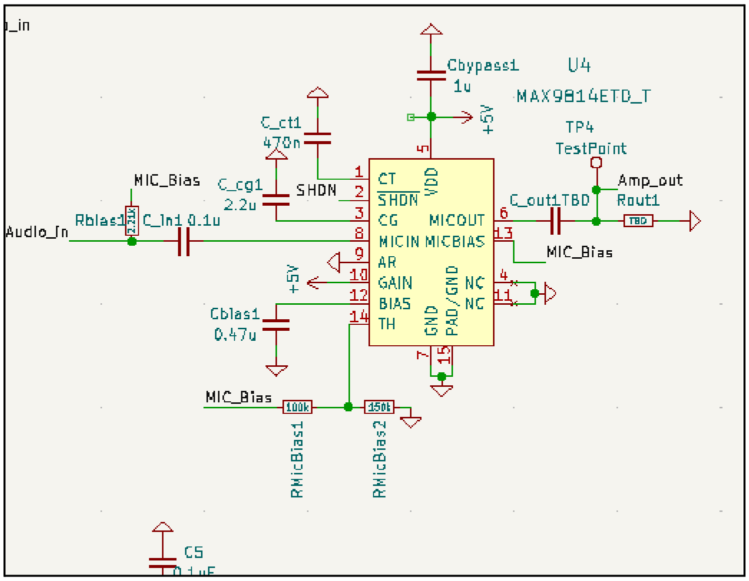


**Figure 2. Amplifier Circuitry**

*Sallen-Key Filter*

Before filter design could begin, the performance required of the hardware filter needed to be defined. For audio, industry standard sampling rate is 44.1 kHz [29], so this rate was chosen for our implementation. With this sampling rate in mind, Nyquist theory dictates that any frequency above 22.05 kHz in our sampled signal would contribute to aliasing. Thus, the primary function of this filter would be to reduce the effect of aliasing. To reduce aliasing, frequencies above 22.05 kHz would need to be attenuated by at least 30 dB. In addition, it was determined via spectral analysis of a vocal sample that the majority of signal power in a human voice will be between 20 Hz and 3.4 kHz, thus 3.4 kHz must be contained within the filter passband.

Filter design utilized the Analog Filter Wizard from Analog Devices [19]. Upon initial analysis, our specifications determined above resulted in a fifth order filter. However, given that the Sallen-Key architecture can produce a second order filter from a single Op-Amp, and that dual Op-Amp chips would be utilized, it was determined that the entirety of the Op-Amp capability of our two dual Op-Amp chips would be used through the construction of an eighth order filter. This eighth order filter provided us additional attenuation of frequencies that would not contribute to aliasing, but would contribute to the noise of the signal (beyond the frequency range of human speech). The filter need not fully attenuate all noise, as additional filtering can be included via a software implemented finite impulse response (FIR) filter. The resulting frequency response and component values of the filter as given by the Analog Devices tool are displayed below.

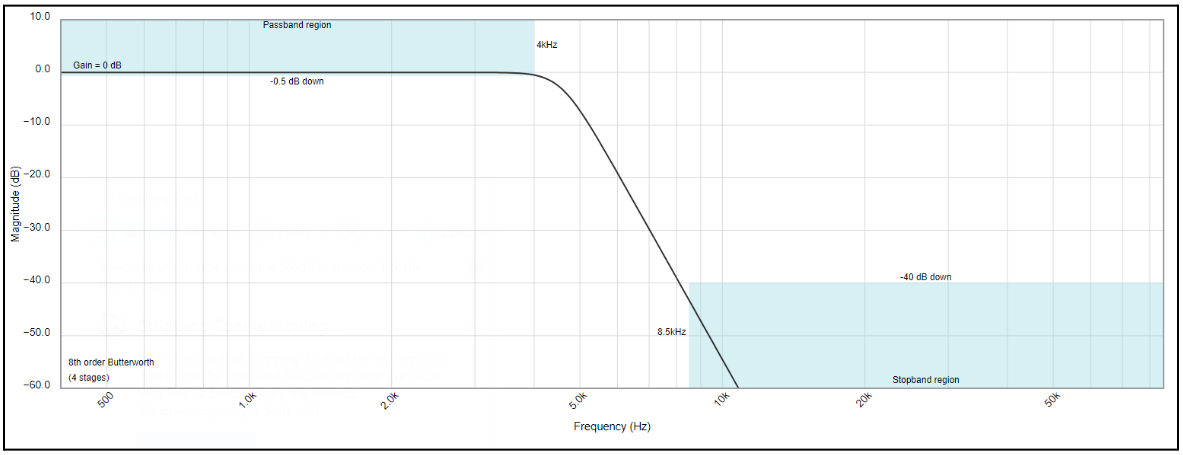


**Figure 3. Sallen Key Filter Frequency Response**

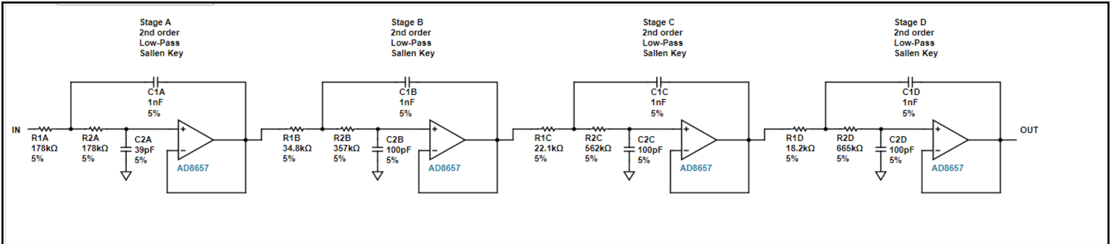


**Figure 4. 8th Order Sallen Key Filter Design Wizard**

The filter was implemented using 2 MCP 6022 dual Operational Amplifiers [26], chosen for their wide bandwidth. The resulting schematic, using values specified by the filter design wizard are shown below.

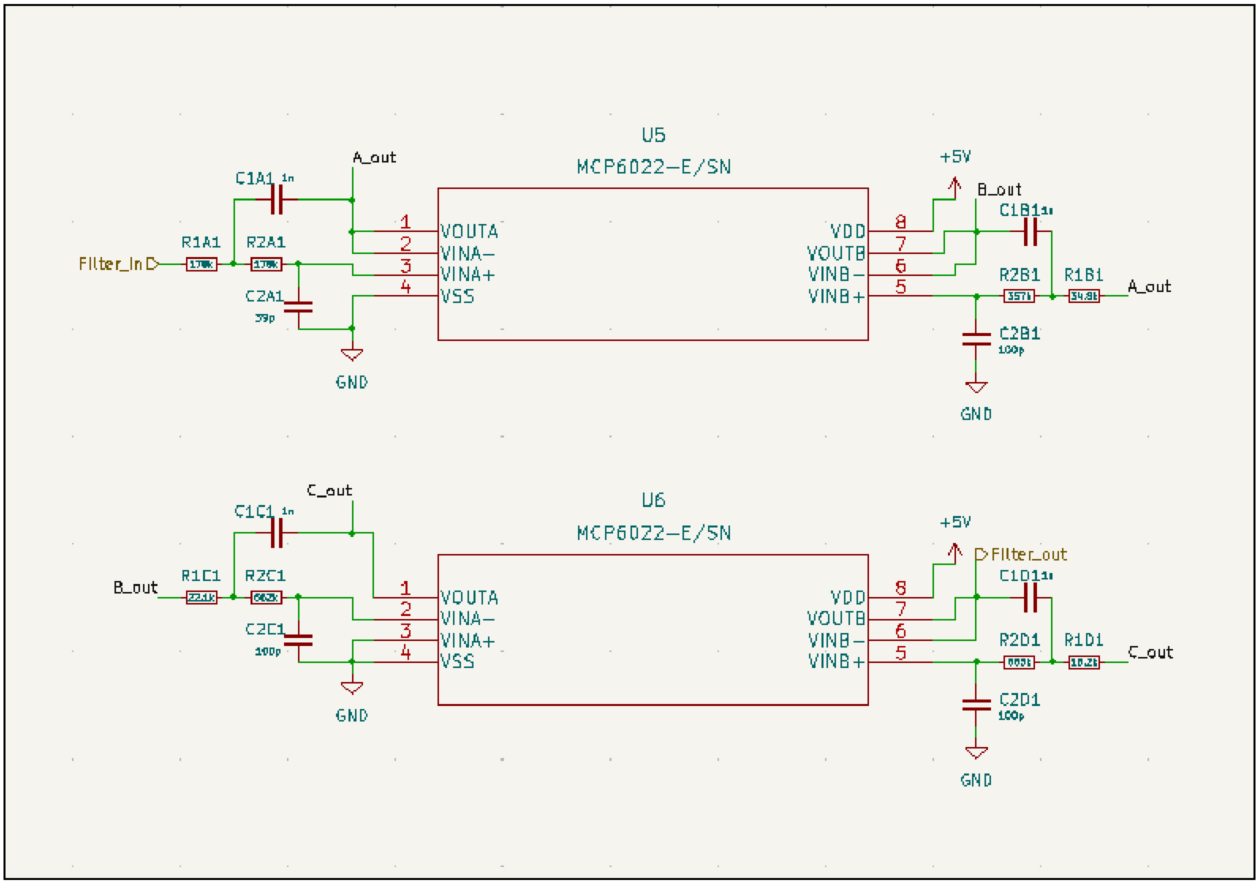


**Figure 5. 8th Order Sallen Key Filter Schematic**

### *Analog to Digital Converter*

As the Raspberry Pi does not feature any analog input pins, analog to digital conversion must be done before the signal is passed to the Pi. At first, we thought we found an easy solution by potentially utilizing the built in ADC on a readily available and cheap Raspberry Pico. It was able to sample 12 bits at a blazing fast 500 kilosamples/second, but after investigating determined the output was subject to extreme levels of noise. Taking into consideration possible large variances in pitch and volume from users, we desired a chip that was more forgiving. Research led to the MAX11163 ADC [25], which is 16 bit instead of 12 for greater signal resolution, but still sampled at 250 kilosamples/second operation. That's more than enough considering the common industry standard for audio sampling is 44.1 kHz [29]. Additionally, this chip was chosen for its SPI compatibility which will help greatly in interfacing with the Pi.

The ADC was implemented using the typical application circuit specified in its datasheet, with a unity gain buffer implemented using an MCP6022 Operational Amplifier [26]. The resulting schematic is shown below.

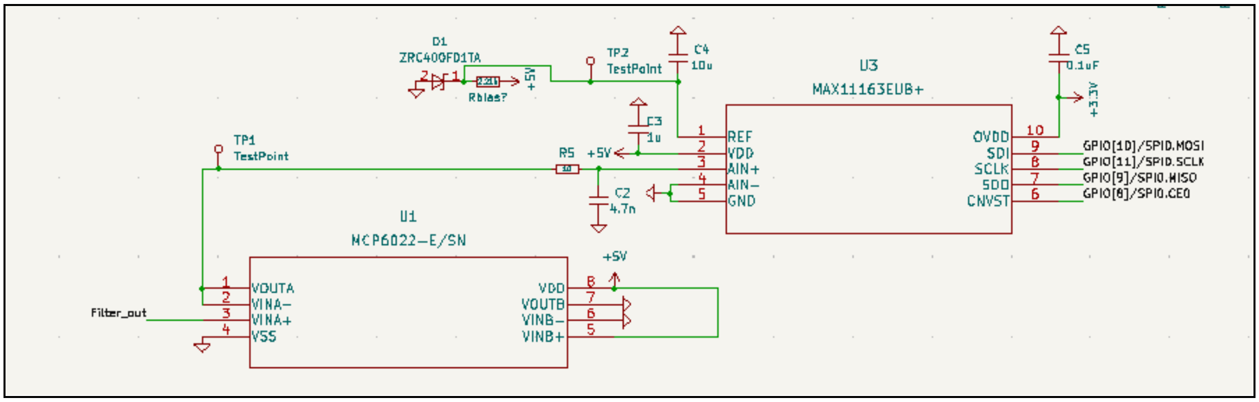


**Figure 6. ADC and Unity Gain Buffer Circuitry**

*Button and Pi Hat Circuitry*

Finally, circuitry was required to integrate a button and interface with the raspberry pi. Because the pi featured both pull up and pull down resistors, minimal circuitry was required for the button to be integrated. Therefore, a simple 1000 kOhm resistor was implemented between the GPIO pin on the Pi and ground. The circuitry surrounding the pi hat and button is shown below.

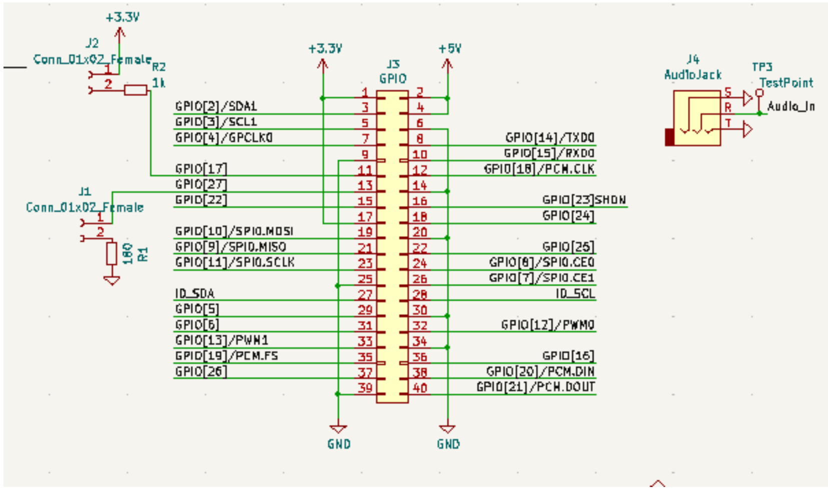


**Figure 7. Pi Hat and Button Circuitry**

## PCB Layout and Assembly

For PCB layout, a Pi Hat shape was chosen to interface easily with the raspberry pi on which the game is run. With that shape constraint in mind, traces were drawn the the convention of copper top traces running vertically, and copper bottom traces running horizontally to minimize unnecessary crossover. One challenge encountered was the small trace size required for many of the integrated circuit chips used. Power traces typically require a higher thickness, but thickness was limited when drawing traces to the integrated circuits. This was addressed by thickening the power traces, but simply drawing thinner traces at the interface with the board. The resulting PCB layout in KiCad is shown below.

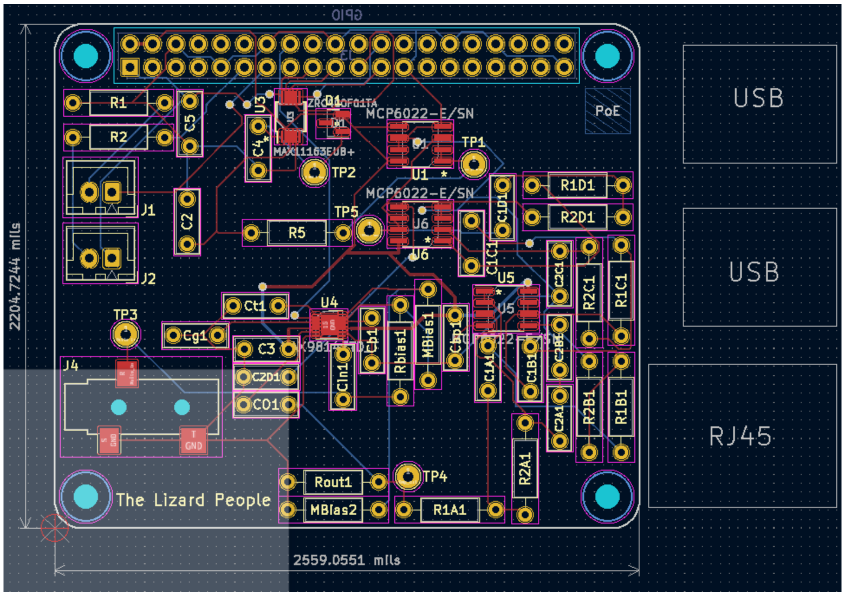


**Figure 8. PCB layout**

## Project Timeline

The plan for our team this semester to complete the project was to split up into a hardware team and a software team. The latter of which was only Isaac who wrote the vast majority of the code for the project with some help from Teddy. Then, the hardware team was made up of Charlie, John, and Teddy and was led by Charlie. Additionally, John was responsible for the housing of our project. These three key components could then be worked on simultaneously as the weeks went on. Figure X below shows the Gantt chart for our projected timeline at the very beginning of this process, which is clear was a very rough outline. The next figure shows the much more detailed and updated timeline which was created after we had been working for a considerable amount of time. The different colors group together the main parts of the project which makes it easier to follow.

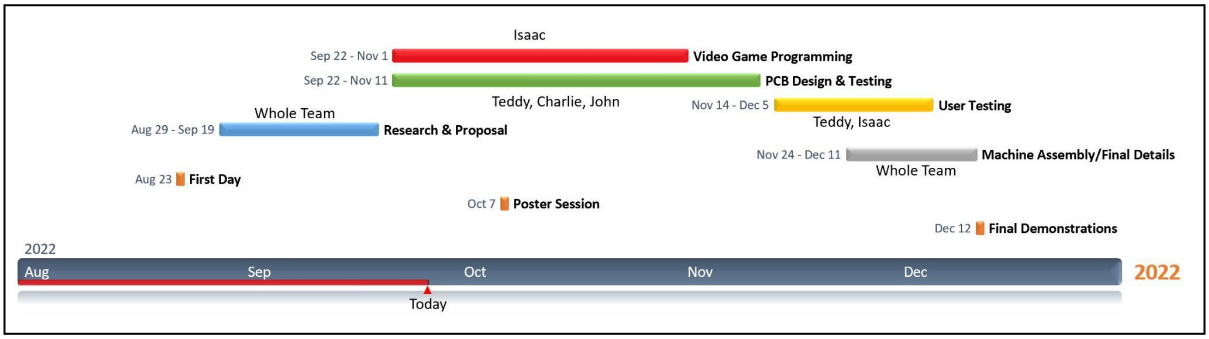
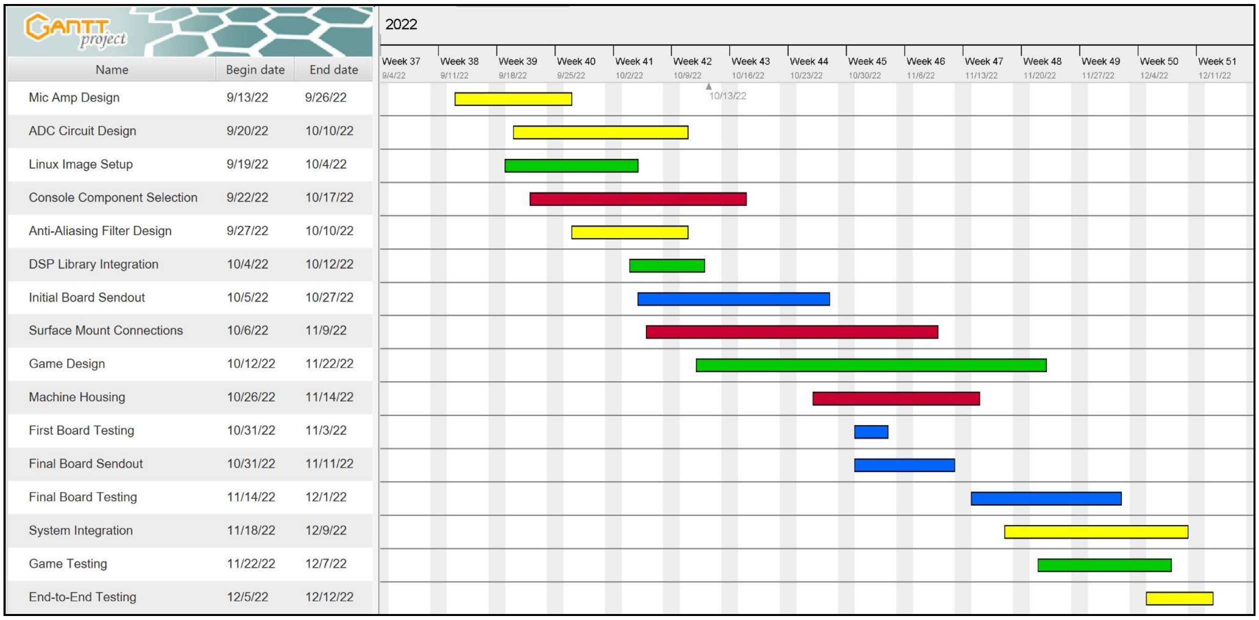


**Figure 9. Proposal Timeline**



| Name | Begin date | End date |
|---|---|---|
| Mic Amp Design | 9/13/22 | 9/26/22 |
| ADC Circuit Design | 9/20/22 | 10/10/22 |
| Linux Image Setup | 9/19/22 | 10/4/22 |
| Console Component Selection | 9/22/22 | 10/17/22 |
| Anti-Aliasing Filter Design | 9/27/22 | 10/10/22 |
| DSP Library Integration | 10/4/22 | 10/12/22 |
| Initial Board Sendout | 10/5/22 | 10/27/22 |
| Surface Mount Connections | 10/6/22 | 11/9/22 |
| Game Design | 10/12/22 | 11/22/22 |
| Machine Housing | 10/26/22 | 11/14/22 |
| First Board Testing | 10/31/22 | 11/3/22 |
| Final Board Sendout | 10/31/22 | 11/11/22 |
| Final Board Testing | 11/14/22 | 12/1/22 |
| System Integration | 11/18/22 | 12/9/22 |
| Game Testing | 11/22/22 | 12/7/22 |
| End-to-End Testing | 12/5/22 | 12/12/22 |

**Figure 10. Midterm Timeline**

By the end of the semester, our workflow followed pretty closely along with the above chart. The only major things that changed were the construction of the final board and the digital signal processing. The board was only finalized in the last week before the capstone fair. Furthermore, the software that handles the digital signal processing was completed in the final few days available for us to work on it. This was due to the fact that final game testing had revealed too many errors in actually measuring the pitch of the user, so we switched to an entirely different Python package and completely reworked the code. It was in those final few days where instead of working on the different parts in parallel we came together as a team and worked together.

## Test Plan

The majority of testing required for the unit as a whole came in the form of software debugging and detailed software testing. To ensure success, the following flowchart was developed to dictate the logical flow of software testing given various outcomes of initial tests.

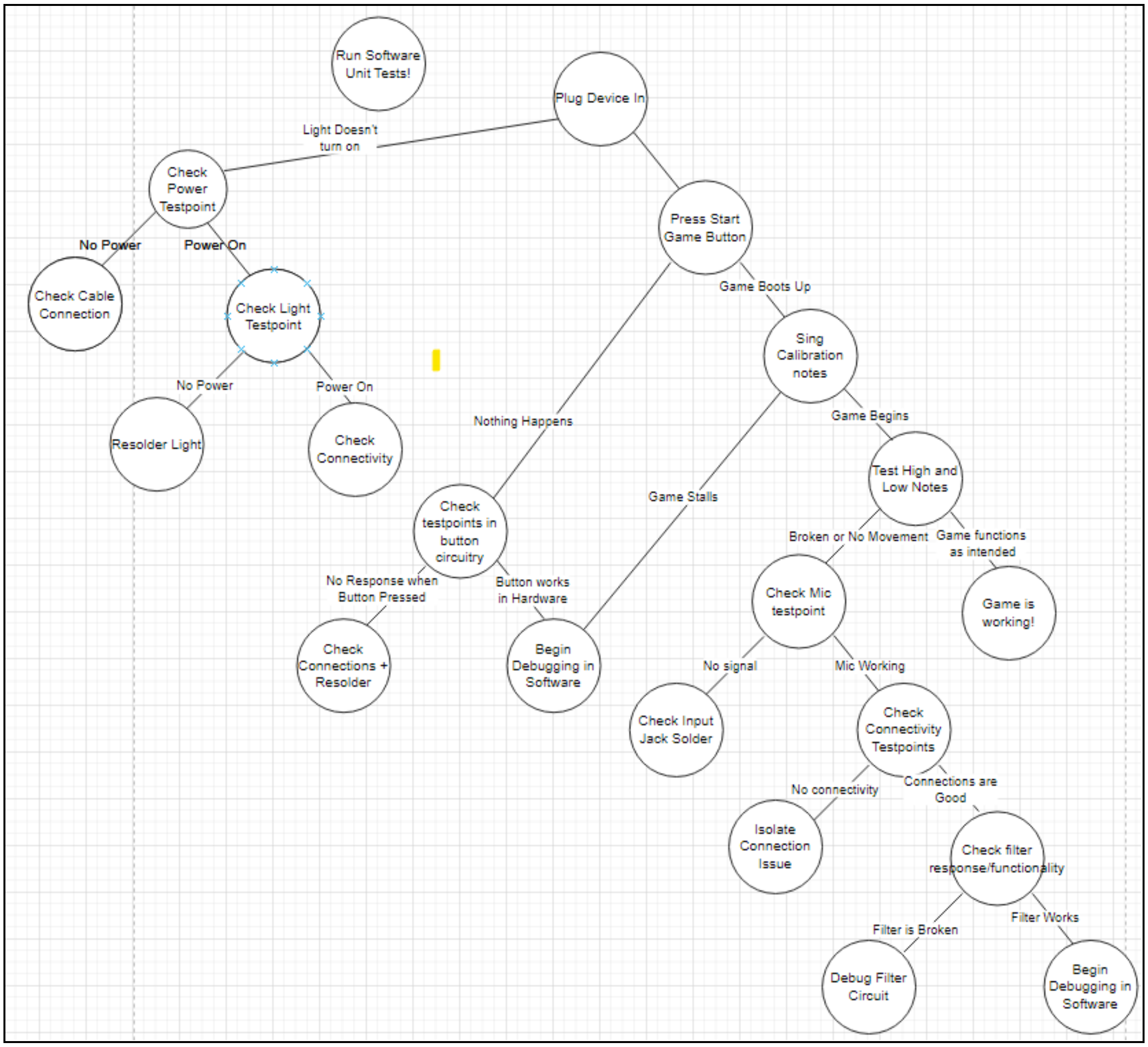


**Figure 11. Software Testing Flowchart**

In addition to the testing described in the flowchart above, unit tests were added to the game system, pitch recognition software, and state machine implemented in software. A CI/CD pipeline was also added in GitHub to ensure frequent software changes would be supported without danger to game function.

## Final Results

We set out to build a game that is an all-in-one experience for a single player. The user needed to be able to plug in the machine to an external USB-C power supply and play Pong against the computer, controlling their paddle using solely their voice. The screen, processor, and PCB needed to be housed in a single portable enclosure that allows anyone to play the game wherever they have access to a power supply. The device should allow for calibration of input pitch in order for players of any voice type to successfully control the input to the game.

More concretely, our success criteria are enumerated below:

- Device can accept and utilize calibration input
- Device can operate a pong game with smooth expected visual output
- Device can provide audio output to complement gameplay
- Device can accept audio input to control user's pong paddle
- Pitch functionality for paddle direction is successfully implemented
- Device is enclosed in an appropriately professional apparatus

Our final project fully realized the goals we set out with at the beginning of the semester. We have successfully met all of the success criteria listed above. Our game of pong is housed in a professional, aluminum enclosure with panel mounted connectors. The user is able to play a pong game with smooth expected visual output with each game started by calibrating a personalized reference pitch. Lastly, a collision sound is played upon contact of the ball with the paddle. In summary, all the criteria we set for ourselves were met with our final product.

## Costs

Our development of Voice Controlled Pong used $467.81 of our budget. The largest cost items consisted of the Raspberry Pi, Screen, Console and SD Card. The Raspberry Pi microcontroller is in high demand and is unlikely to be heavily discounted when purchased in volume. The screen purchased for our project was bought on Amazon, at scale these would be sourced from a wholesaler for a significant price deduction. The PCB and Components (Raspberry Pi Hat) for our project were manufactured and assembled separately. For an order of 10,000 boards, the unit price would drop to $0.43 as per an online quote [30]. Combined with the components discounted unit price an order of 10,000 boards would cost $0.72 per unit. Hammond Manufacturing [17], the supplier of the case, is able to custom machine the consoles. This would allow us to easily order large quantities of the console that would come ready to assemble.

| **Item** | **Total Cost** |
|---|---|
| Raspberry Pi 4 | $167.94 |
| 10" IPS Touchscreen Display (Screen) | $75.89 |
| PCB and Components (Raspberry Pi Hat) | $62.04 |
| Hammond Manufacturing Console | $40.72 |
| SD Card (update w/ capacity) | $27.37 |

**Figure 12. Abbreviated Cost Table**

## Future Work

If we had another semester to develop this project, something we considered was adding a “hard mode” to the game. Currently the game works by determining if the inputted pitch is higher or lower than the user’s reference pitch and moves the paddle accordingly. In hard mode, the high and low calibration notes would be used to set upper and lower frequency bounds instead of being averaged to create a reference note. These high and low bound frequencies would correspond to the upper and lower bounds of the screen accordingly, so that a linear scale is made with a certain frequency corresponding to a specific position of the paddle. Therefore, when a user inputs a specific vocal frequency, the paddle moves towards the frequency’s corresponding position on the screen. We did not implement this feature because it requires significantly more processing power, microphone performance and user skill.

## References

[1] *Maker Faire 2008: Pong and Asteroids Watch*, (Jan. 01, 2010). [Online Video]. Available: https://ieeetv.ieee.org/channels/spectrum/maker-faire-2008-pong-and-asteroids-watch

[2] T. Perry and P. Wallich, “Inventing the Atari 2600,” *IEEE Spectrum*, Mar. 1983, [Online]. Available: https://spectrum.ieee.org/atari-2600

[3] R. Rabenja, S. Amine, and J. Davidson, “Speech recognition for game control,” in *The 2nd Annual IEEE Northeast Workshop on Circuits and Systems, 2004. NEWCAS 2004.*, 2004, pp. 97–99. doi: 10.1109/NEWCAS.2004.1359028.

[4] “Raspberry pi documentation,” *Raspberry Pi*. [Online]. Available: https://www.raspberrypi.com/documentation/. [Accessed: 12-Sep-2022].

[5] “Pygame,” *Pygame Front Page - pygame v2.1.1 documentation*. [Online]. Available: https://www.pygame.org/docs/. [Accessed: 12-Sep-2022].

[6] KiCad, “Documentation,” *KiCad*. [Online]. Available: https://docs.kicad.org/. [Accessed: 26-Sep-2022].

[7] “DigiKey Electronics - Electronic Components Distributor.” https://www.digikey.com/ (accessed Sep. 12, 2022).

[8] Microsoft, “Running visual studio code on raspberry pi,” *RSS*, 03-Nov-2021. [Online]. Available: https://code.visualstudio.com/docs/setup/raspberry-pi. [Accessed: 26-Sep-2022].

[9] “Buildroot - making embedded Linux Easy.” [Online]. Available: https://buildroot.org/. [Accessed: 27-Sep-2022].

[10] “Printed Circuit Board Recycling Methods,” Workshop Materials on WEEE Management in Taiwan, 2012. https://www.epa.gov/sites/default/files/2014-05/documents/handout-10-circuitboards.pdf

[11] “Raspberry Pi boards are now being recycled - at the same factory that made them,” *ZDNET*.https://www.zdnet.com/article/raspberry-pi-boards-are-now-being-recycled-at-the-same-factory-that-made-them/ (accessed Sep. 26, 2022).

[12] “Final Report | Recycling of Liquid Crystal Displays for Maximum Resource Recovery | Research Project Database | Grantee Research Project | ORD | US EPA.” https://cfpub.epa.gov/ncer_abstracts/index.cfm/fuseaction/display.abstractDetail/abstract_id/9847/report/F (accessed Sep. 26, 2022).

[13] “NFPA 70®: National Electrical Code®.” https://www.nfpa.org/codes-and-standards/all-codes-and-standards/list-of-codes-and-standards/detail?code=70 (accessed Sep. 12, 2022).

[14] “NEMA 1 Enclosures - NEMA 1 Electrical Enclosure | Nema Enclosures.” https://www.nemaenclosures.com/enclosure-ratings/nema-rated-enclosures/nema-1-enclosures.html (accessed Sep. 12, 2022).

[15] Webmaster, “Embedded C Coding Standard,” May 26, 2016. https://barrgroup.com/embedded-systems/books/embedded-c-coding-standard (accessed Sep. 12, 2022).

[16] “Electrical - Standards | Occupational Safety and Health Administration.” https://www.osha.gov/electrical/standards (accessed Sep. 12, 2022).

[17] H. Mfg, “Electrical & Electronic Enclosures, Cabinets & Racks, Outlet Strips and Electronic Transformers,” *Hammond Mfg.* https://www.hammfg.com/ (accessed Dec. 13, 2022).

[18] “KiCad EDA.” https://www.kicad.org/ (accessed Dec. 13, 2022).

[19] “Analog Filter Wizard.” https://tools.analog.com/en/filterwizard/

[20] R. P. Sallen and E. L. Key, “A practical method of designing RC active filters,” IRE Transactions on Circuit Theory, vol. 2, no. 1, pp. 74–85, 1955, doi: 10.1109/TCT.1955.6500159.

[21] “PONG Trademark of ATARI INTERACTIVE, INC. - Registration Number 2611782 - Serial Number 76148525 :: Justia Trademarks.” http://trademarks.justia.com/761/48/pong-76148525.html (accessed Dec. 13, 2022)

[22] “Patents Assigned to RASPBERRY PI (TRADING) LIMITED - Justia Patents Search.” https://patents.justia.com/assignee/raspberry-pi-trading-limited (accessed Dec. 13, 2022).

[23] T. Ramabadran, “Method and Apparatus for Speech Reconstruction within a Distributed Speech Recognition,” 2004/0138888A1

[24] “Microphone Amplifier with AGC and Low-Noise Microphone Bias.” Analog Devices. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/max9814.pdf

[25] “MAX11163 16-Bit, 250ksps, +5V Unipolar Input, SAR ADC, in Tiny 10-Pin µMAX.” Analog Devices. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/MAX11163.pdf

[26] “MCP6021/1R/2/3/4 Rail-to-Rail Input/Output, 10 MHz Op Amps.” Microchip. [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/20001685E.pdf

[27] “Raspberry Pi trademark rules and brand guidelines,” *Raspberry Pi*. https://www.raspberrypi.com/trademark-rules/ (accessed Dec. 13, 2022).

[28] "Should I match impedances of my microphone to my mixer?". Shure Service & Repair. Shure. May 23, 2022. Retrieved June 27, 2022

[29] R. R. Devi and D. Pugazhenthi, “Ideal Sampling Rate to reduce distortion in Audio Steganography,” in INTERNATIONAL CONFERENCE ON COMPUTATIONAL MODELLING AND SECURITY (CMS 2016), 2016, vol. 85, pp. 418–424. doi: 10.1016/j.procs.2016.05.185.

[30] “PCB Cost Calculator | Online PCB Manufacturing Quote.” https://www.pcbcart.com/quote (accessed Dec. 13, 2022).

## Appendix

| Item | Total Cost |
|---|---|
| Raspberry Pi 4 | $167.94 |
| SD Card (update w/ capacity) | $27.37 |
| 10” IPS Touchscreen Display | $75.89 |
| PCB Manufacture | $33.00 |
| PCB Assembly | $30.00 (estimate) |
| Hammond Manufacturing Console | $40.72 |
| USB C Round Panel Mount Extension Cable | $24.88 |
| Mini LED Arcade Button - 24mm | $5 |
| Shunt Voltage Reference IC | $3.87 |
| 16 Bit Analog to Digital Converter | $11.11 |
| Amplifier IC 1-Channel (Mono) Class AB | $4.36 |
| General Purpose Amplifier | $7.24 |
| Ground Reference (Virtual) Voltage Reference IC | $2.46 |
| Panel Mount Stereo Audio Extension | $3.95 |
| Unidirectional Lapel Microphone | $16.97 |
| Micro USB Right Angle Ribbon Cable | $15.00 |
| Hex Standoff Threaded M2.5x0.45 Steel 0.472" | $2.85 |
| M2.5x0.45 Hex Nut 0.197" (5.00mm) Steel | $0.9 |
| Hex Standoff Threaded M2.5x0.45 Brass 0.197" | $1.45 |
| M2.5x0.45 Flat Head Machine Screw Phillips Drive | $0.75 |
| HDMI Right Angle Ribbon Cable | $8.41 |
| Raspberry Pi Hat Header Pins | $6.03 |
| Silicon Sealant and Adhesive | $7.66 |