ECE 445

SENIOR DESIGN LABORATORY
DESIGN DOCUMENT

Hand Gesture Audio Effects System

Team No. 39

Sarthak Singh
(singh94@illinois.edu)
Zachary Baum
(zbaum2@illinois.edu)
Sergio Bernal
(sergiob2@illinois.edu)

TA: Zicheng Ma

February 22, 2024

1. Introduction

1.1 Problem and Solution

Problem: Both in video production and live music, individuals often want the ability to apply effect to audio in real time. There are solutions for this such as DJ controllers or hardware effects processors. For all of these systems, you usually need to click buttons, turn knobs, etc. Unfortunately, not all people have the ability to perform these functions due to disabilities. Furthermore, those within live settings, operating physical equipment may be challenging to manage with all the other aspects of production.

Solution: This project aims to develop a gesture-controlled audio effects processor. This device will allow users to manipulate audio effects through hand gestures, providing a more dynamic and expressive means of audio control. The device will use a camera to detect gestures, which will then adjust various audio effect parameters in real-time. This will allow for the same features of something like a DJ mixer with less equipment & the ability to with your hands free.

1.2 Visual Aid

On a high level, the final product would contain a camera, a speaker, and the electronics in between. The camera would be monitoring the hands of someone who is in view of the camera. In this example let's say an effect “x” is triggered by a thumbs up. In a standby state, we will be playing some audio files out of the speaker with no effects.

Once the camera detects a thumbs up it will send the proper signal to our PCB which will then pass all the audio signals through a system which will apply the “x”. That effect will remain on until another gesture is shown in the camera & there will be a gesture for “normal”.

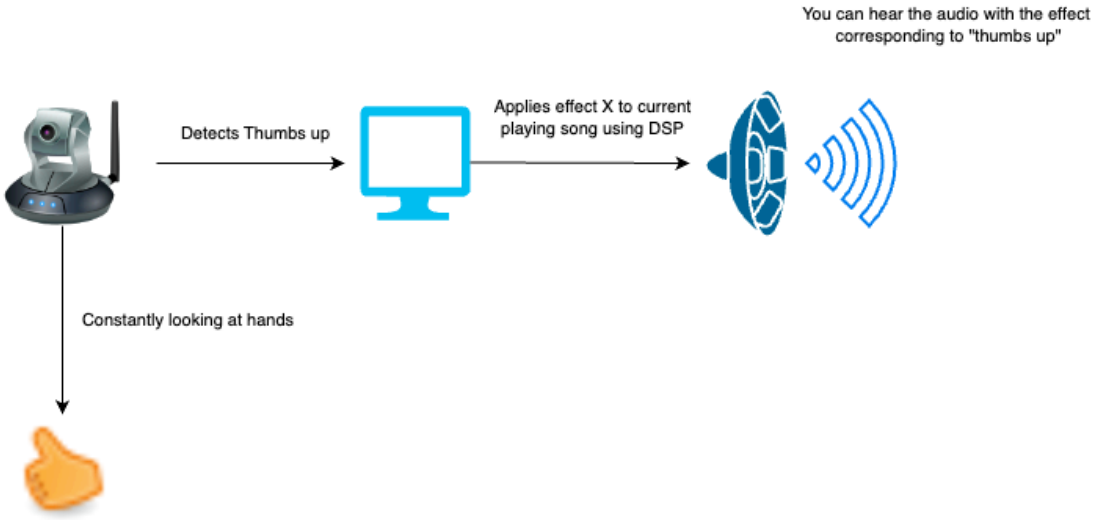

Figure 1a High-Level Overview of the Hands Gesture Audio Effects System

In terms of how the physical design will look we suspect there to be a camera viewing the subject and a speaker/display facing in the same direction all inside of one box. In section 2.2 we will expand on this idea.

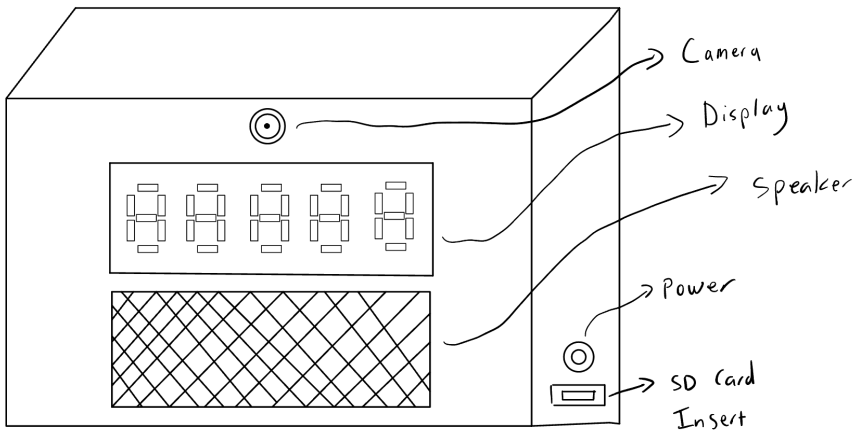

Figure 1b Physical design initial vision

1.3 High-level requirements list

- The gesture-controlled audio effects processor is able to detect hand gestures using a camera with 95% accuracy +-5% within 5 seconds of making the gesture.
- The gesture-controlled audio effects processor is able to apply 5 digital effects to an audio signal and apply that effect to the external speaker within a 1 second of receiving a signal from the gesture detection subsystem with a tolerance of +-½ of a second.
- The external display should accurately show the current playing sound effect with 99% accuracy with a tolerance of +-1%

2. Design

2.1 Block Diagram

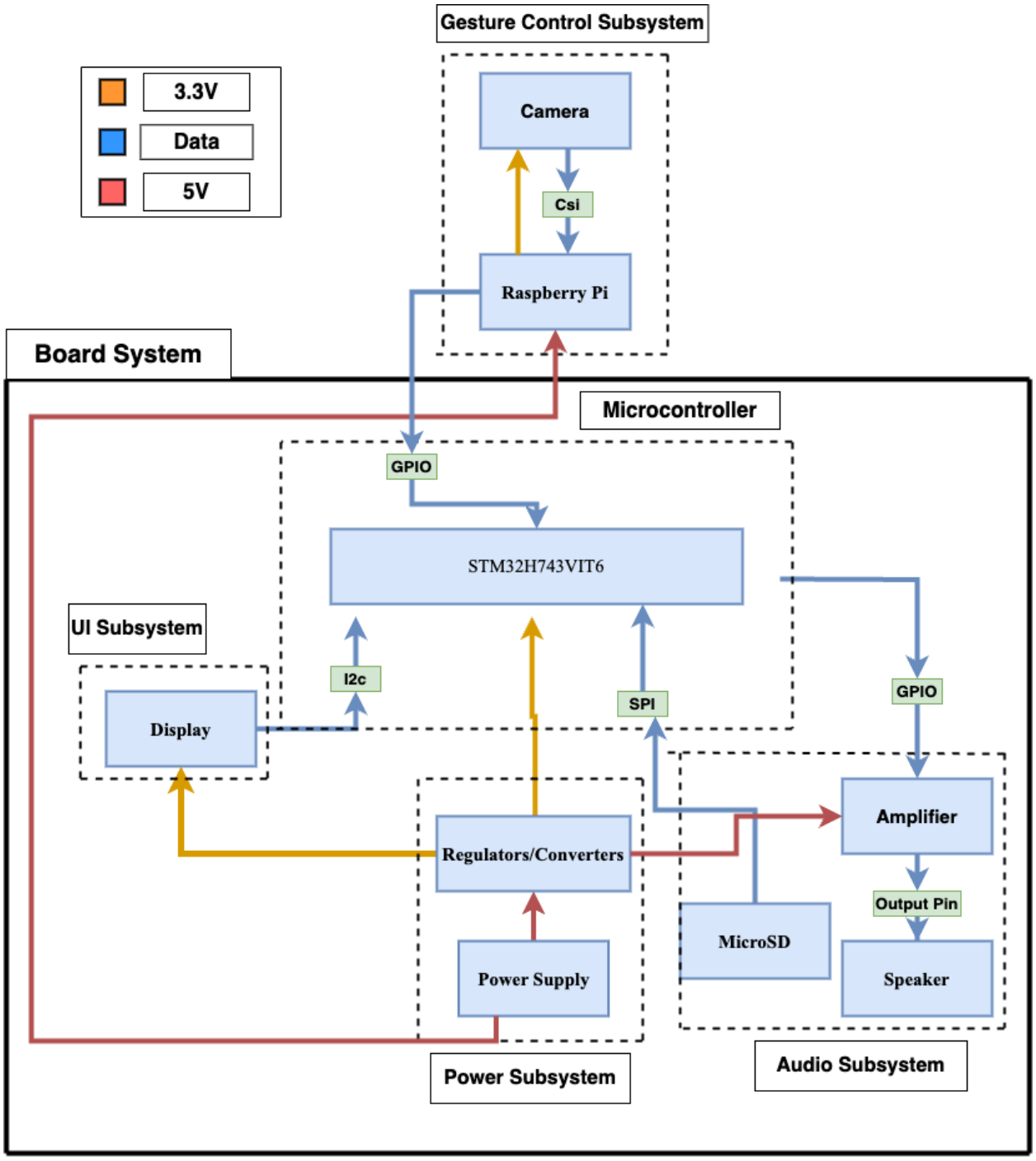

Figure 2. Block Diagram of the Hands Gesture Audio Effects System Design

2.2 Physical Design

The illustration of Figure 3 shows our physical enclosure appearance. Inside of this box, there will be a Raspberry Pi, PCB, and other components necessary for the function of the system. At the top, there will be a camera followed by a display unit, and then a speaker. On the right side of the enclosure, there will be a microSD card insert that will hold the audio file being modified by an audio effect. In addition, there is a DC power jack which will be connected to an outlet using a 5V 4.5A wall power adapter.

For the moment, there are no omitted dimensions on this diagram since there is no information about the dimensions of the custom PCB. Once there is a rough draft of the PCB board dimensions, then a decision can be made on how it will be mounted with the other components to the physical enclosure.

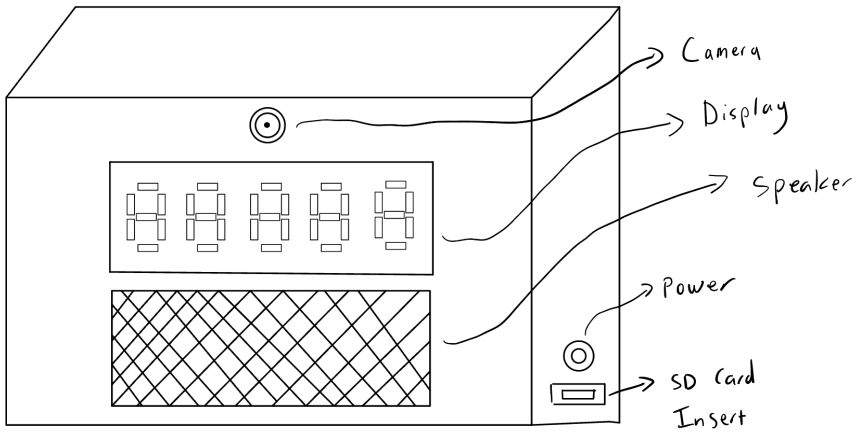

Figure 3. Initial vision for physical design

2.3 User Interface Subsystem

The UI (User Interface) Subsystem facilitates interaction between the user and the Audio Subsystem primarily through a display. This subsystem is essential for providing visual feedback to the user, showcasing current audio effect settings. For example, if the user is making a hand gesture that leads to reverb, then the display signals the user that the effect has been successfully applied to the audio input.The display is a critical component for user feedback, ensuring users can navigate and control the device's functions effectively without physical buttons. The display communicates via I^2C with the microcontroller of the custom PCB. This allows for dynamic updates on the screen based on system changes or user interactions via other input methods, such as the Gesture Control Subsystem. The display will be a Chip-on-Glass Character Liquid Crystal Display. It can display up to 20 characters. By default, it will display that an audio effect hasn’t been applied (no gesture detection) or the microSD (a part of the Audio Subsystem) hasn’t been plugged into the socket input. Lastly, the UI subsystem (LCD) uses a 3.3 V power supply coming from the Power Subsystem. Furthermore, this specific external display was chosen since it fits the 3.3 V design decision mentioned in the Power Subsystem description.

Component List:

1 x NHD-C0220BIZ-FSW-FBW-3V3M (Liquid Crystal Character Display Module)

Table 1. UI Subsystem RV Table

Requirement	Verification
The display must show the user what audio effect has been applied to the input audio signal, 0.1 seconds (at most) after the microcontroller communicates with it via I^2C that a unique hand gesture has been detected by the gesture detection algorithm.	1. Connect the female end of a jumper wire to each STATE signal (3 in total) of the Raspberry Pi's pin headers. 2. Connect the male end of each jumper wire to a respective LED's cathode pin on a breadboard. 3. Connect each LED's anode pin to a respective male end of a jumper wire on the breadboard. 4. Connect the female end of each jumper wire (connected to LED anode pin) to its respective STATE pin connector on the custom PCB. 5. Start a timer for when the three LEDs change to a new STATE value. 6. Stop timer until the display changes to the STATE value's name. If we cannot time how quickly it is, safe enough to say it's quicker than .2 seconds

2.4 Gesture Control Subsystem

The Gesture Control Subsystem utilizes a camera to detect hand gestures which are translated into an electrical signal which is passed into a custom PCB using the GPIO pins on a Raspberry Pi. This subsystem consists of a imaging sensor (camera) module connected to the Raspberry Pi via CSi (Camera Serial Interface) in terms of hardware. In software, each captured image is analyzed to recognize specific gestures in software. The Raspberry Pi serves as the gesture interpretation layer such that it converts recognized gestures into command signals to be sent to the Audio Subsystem (for digital signal processing) and UI Subsystem (for showing the applied audio effect). Also, this subsystem interacts with the Power Subsystem by a USB-A to Micro USB B cable. Moreover, the USB-A is attached to the PCB and the Micro B USB is connected to the power input of the Raspberry Pi. In terms of communication between the Raspberry PI and PCB, three GPIO pins (bits) will be utilized to form a total of 8 unique signals that this subsystem can send to the PCB. The layout of the Raspberry Pi 3B layout pins are shown in Figure 4, and the pins to be used will be GPIO 17, 22, and 27.

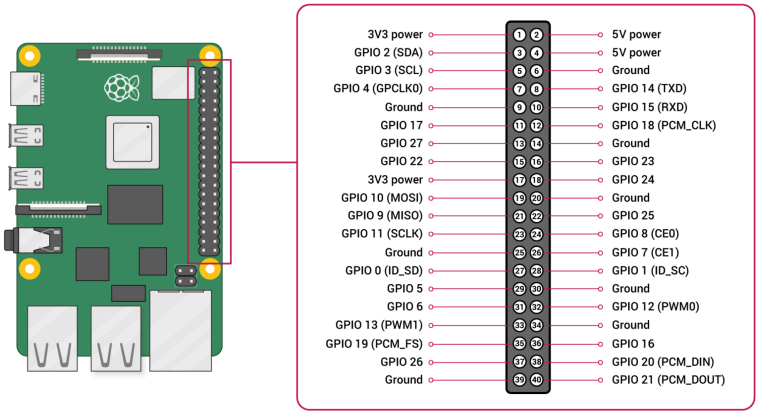

Figure 4. GPIO layout of Raspberry Pi 3B

As a final note, the Raspberry Pi was chosen to handle the image processing aspect of the design since it requires a more complex algorithm that requires a lot of memory. According to its datasheet, it has about 1 Gb of memory which should be plenty for the Gesture Control Subsystem. [10] As for the camera module, it was chosen since it was made specifically for the Raspberry Pi board.

Component List:

1 x Raspberry Pi 3 Model B

1 x RPI-CAM-V2 (Camera module that detects hand gestures)

Table 2. Gesture Control Subsystem RV Table

Requirements	Verification
The image processing algorithm that analyzes the captured images from the camera must have a timing of at least 20 frames per second to get an accurate detection of a hand gesture.	1. Use the <time.h> and time() library to measure the start and end time for a single while loop iteration (used to capture an image from the camera using OpenCV library). 2. Create the variable that stores the start time. 3. Create the variable that stores the end time. 4. Subtract the variable storing the start time from the variable that stores the end time. 5. Take the reciprocal of that value to get the number of frames that got analyzed per second. Note that this procedure is for software done in C++.

The gesture detection algorithm must detect a hand gesture in less than 5 seconds.	1. Connect the female end of a jumper wire to each STATE signal (3 in total) of the Raspberry Pi's pin headers. 2. Connect the male end of each jumper wire to a respective LED's cathode pin on a breadboard. 3. Connect each LED's anode pin to a respective male end of a jumper wire on the breadboard. 4. Connect the female end of each jumper wire (connected to LED anode pin) to its respective STATE pin connector on the custom PCB. 5. Start a timer for when the three LEDs are off (default state) and another person makes a hand gesture to the camera. 6. Stop timer until the LEDs change to an applied audio effect state.

2.5 Power Subsystem

The Power Subsystem provides essential power distribution to all components within the three other subsystems. It is composed of a primary power source (5V 4.0 A wall power adapter) that will connect to a DC barrel jack. This barrel jack will be connected to the USB-A connector to supply the Raspberry Pi with 5 V. In addition, the barrel jack will be connected to an eFuse which determines whether the wall power supply is valid. Otherwise, it will cut off power to the custom PCB components for protection. At the output voltage of the eFuse, it will be the same voltage as the input voltage of the wall adapter. Furthermore, this 5 V voltage will be filtered by a capacitor connected to the input of a linear regulator which will be stepped down to 3.3 V at the output of that linear regulator. The fixed output voltage (3.3V) will be used as the power supply for all components in the custom PCB except the amplifier and USB-A connector. Also, there will be a test point for the input/output voltage ports of the eFuse and at the fixed 3.3V so that the ripple of each voltage can be observed for verification.

Component List:

1 x LM1085ISX-3.3/NOPB (Linear regulator for 3.3 V power supply)

1 x WSU050-4000 (5 V 4.0 A wall power supply)

1 x PJ-102AH (DC Barrel Jack Connector for wall power supply)

1 x 87583-3010RPALF (USB-A connector soldered on custom PCB)

1 x USBAM11851M2MICBBK5A (USB-A to microUSB B Cable for Raspberry Pi Power)

1 x TPS259814ARPWR (eFuse for safety protection if user plugs in the incorrect power supply)

Table 3. Power Subsystem RV Table

Requirement	Verification
The linear regulator should provide 3.3 V ± 0.5% at its output to the necessary supply pins of the custom PCB.	1. Connect to the 3.3 V test pin and GND test pin on custom PCB using the connectors coming from an oscilloscope. 2. Add a measurement for the average voltage. 3. Measure the peak-to-peak voltage (ripple) using horizontal cursors of the oscilloscope. 4. Check if the peak-to-peak voltage is at most 0.033 V which is 0.5% above and below 3.3 V.

As shown in Table 3, the power supply for the entire custom PCB board should be about 3.3 V. This power supply voltage was decided beforehand since typically most chips require 3.3 V. However, the Raspberry Pi and amplifier needed 5 V so a linear regulator was required to step down an input voltage of 5 V to 3.3 V. This design decision allows both the Raspberry Pi and PCB to be powered by the same wall adapter that connects to an outlet. A wall adapter (5 V 4.0 A) was selected based on the worst case scenario of current drawn by each board. Looking at the Raspberry 3 Model B's datasheet, it recommends that the power supply should be at 2.5 A. [10] Thus, this current was used as the worst case scenario of current drawn by the Raspberry Pi. On the other hand, the custom PCB draws up to roughly 1.0 A. This estimation was found based on the sum of all our component's rated current stated in their respective datasheet. As a result, a 5V wall power adapter was picked out to supply 4.0 A to the entire system. Although this supply current is bigger than what is needed (~3.5 A), it is fine just in case the needed current is slightly above the estimation.

As for the schematic design of this power subsystem, the connections are shown below in Figure 5.

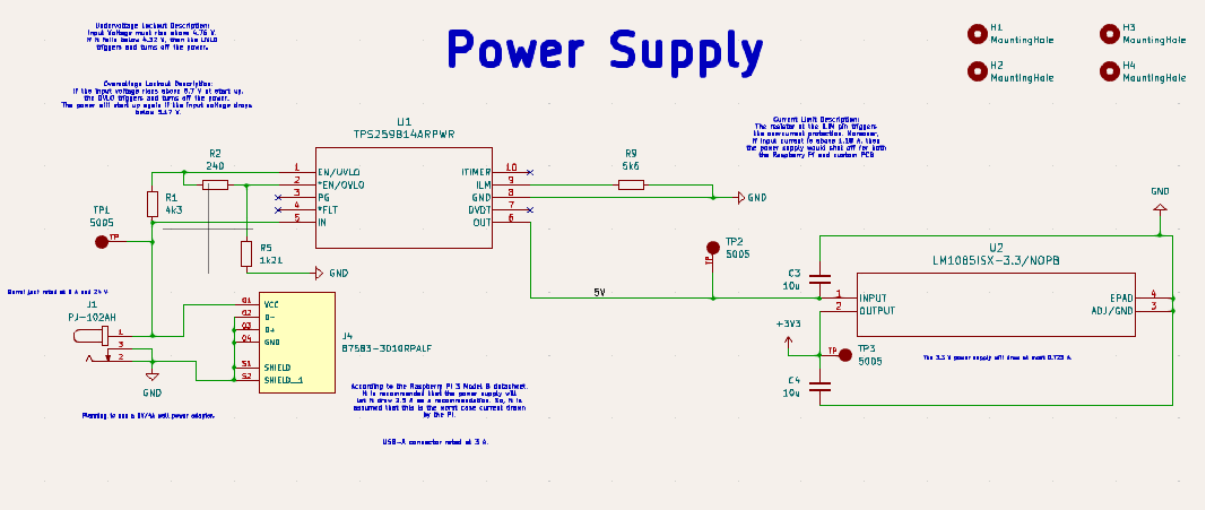

Figure 5. Power Supply Circuit Diagram

As illustrated by Figure 5, the wall adapter will connect to a DC barrel jack. The U1 component is the eFuse. The eFuse triggers when the user either connects a voltage source below 4.76 V or above 5.7 V such that it will shut off the current getting outputted to the linear regulator. This situation can happen if the user mixes up two wall adapters with different voltages to power the system. The eFuse also handles short circuit protection for the components connected to its output. Moreover, if the output current is above 1.18 A then it will shut off the power since a short circuit has occurred within the custom PCB. This will help with short circuits due to poor soldering. The voltage limits and current limit was set due to the given equations referenced in the eFuse datasheet. [9] Note that the overvoltage threshold would have been lowered closer to 5 V. However, it would have required getting special resistors not available to us.

2.6 Audio Subsystem

The focus of the Audio Subsystem is applying an audio effect on a .wav file stored in a microSD, and outputting that new audio signal. First, the microcontroller on the custom PCB interacts with the microSD using SPI to obtain the stored digitized audio signal. Then, the microcontroller applies desired audio effects on the input signal using its processor, and converts the modified signal back to analog form via a digital-to-analog converter (DAC). Finally, the amplifier within this subsystem will boost the processed audio signal to a speaker. This subsystem will only apply an audio effect if told by the Gesture Control Subsystem. Also, this subsystem relies on the Power Subsystem for the energy needed to operate amplifiers and processing units.

Audio effects:

Reverb: Simulates the echo and ambiance of a physical space, adding depth to the audio. To implement, we will use the Schroeder Reverb Algorithm, which uses multiple feedback delay lines and all-pass filters to simulate room ambiance.

Delay: Creates an echo effect by replaying the audio signal after a short period. The basic implementation requires a circular buffer to store the audio samples for the delay period.

Gain Adjustment: Varies the amplitude of the audio signal, controlling volume. Multiply each incoming audio sample by a gain factor. A gain factor greater than 1 increases volume, while a gain factor less than 1 decreases it.

Low-pass Filter: Attenuates frequencies above a cutoff, reducing high-frequency noise or brightness. Implement the filter using a simple discrete time equation: $y[n] = \alpha * x[n] + (1 - \alpha) * y[n-1]$, where $x[n]$ is the input signal, $y[n]$ is the output signal, n is the sample index, and α (alpha) is the filter coefficient related to the cutoff frequency.

High-pass Filter: Removes low-frequency components, reducing rumble or bass. Use a simple high-pass filter equation: $y[n] = \beta * (x[n] - x[n-1]) + \gamma * y[n-1]$, where β and γ are coefficients that depend on the cutoff frequency.

Note: Will Use the CMSIS-DSP library for optimized DSP functions

Component List:

1 x STM32H743VIT6 (Microcontroller)

1 x ECS-400-8-36B-7KY-TR (Crystal Oscillator)

1 x MSD-4-A (MicroSD Connector)

1 x S312TLKJM-C1000-3 (MicroSD 128 Mb)

1 x D6R90 F2 LFS (Push Button for MCU Reset)

1 x LM386N-1/NOPB (Amplifier for Audio DAC output)

1 x SPKM.15.8.A (Speaker to hear audio)

Table 4. Audio Subsystem RV Table

Requirement	Verification
The sound being outputted by the speaker must be less than 80 dB for safety purposes	1. Connect the audio source to the input of the Audio Subsystem. 2. Set the audio source to produce a continuous sine wave signal at 1 kHz. 3. To measure the sound db, a sound level meter app on a phone will be used. Position the Sound Level Meter at a distance of 1 meter from the speaker, aligned with the center of the speaker cone, in a quiet, enclosed room to avoid ambient noise interference. 4. Gradually increase the volume of the audio source until the sound level meter stabilizes. 5. Record the maximum sound pressure level (SPL) reading displayed by the Sound Level Meter. 6. Repeat the procedure three times and calculate the average SPL to ensure reliability. Add SPL value to report
The effect applied to the input audio signal must be heard clearly such that there should not be any static sound unless it is a distortion audio effect	1. Connect the audio source to the Audio Subsystem's input and the output to both the Audio Analyzer and headphones/speakers.

	2. Set the audio source to produce a clean sine wave signal at 1 kHz. 3. Apply the desired audio effect using the microcontroller within the Audio Subsystem. 4. Conduct a subjective listening test by playing the processed audio through headphones/speakers and noting any static noise or undesired artifacts. 5. Adjust the effect parameters and repeat the test if static noise is detected in the absence of a distortion effect. Add data to report

2.7 Tolerance Analysis:

One critical part of the project design was deciding the linear regulator part such that it won't reach its power dissipation rating when the worst case scenario input current is going through it. The LM1085ISX-3.3/NOPB was specifically chosen to handle this issue. Moreover, this linear regulator has a maximum voltage dropout of 1.5V at 3.0 A. [7] Using the worst case scenario current drawn by the PCB (~0.753A) and assuming the input current and output current are the same, I can calculate the power dissipation of the linear regulator. Then, compare that value to the rated power dissipation of the chosen linear regulator using Equation 1.1.

i_{out} : Maximum drawn current

v_{in} : Input Voltage

v_{out} : Output Voltage

P_D : Power Dissipation of Linear Regulator

$$P_D = i_{out}(v_{in} - v_{out}) \quad (1.1)$$

$$P_D \approx (0.753\ A)(5\ V - 3.3\ V) \approx 1.28\ W \quad (1.2)$$

$$P_{D,rating} \approx (3.0\ A)(1.5\ V) \approx 4.50\ W \quad (1.3)$$

As seen by Equation (1.2) and Equation (1.3), the power dissipated by the linear regulator during the worst-case scenario is much less than the rated power dissipation for the linear regulator. Thus, a heat sink will not be needed for the design. Furthermore, the junction temperature for the linear regulator has been calculated for an ambient temperature of 38 degrees Celsius (100 degrees Fahrenheit) using Equation (1.4).

T_j : Junction Temperature

$R_{\theta JA}$: Junction-to-Ambient Thermal Resistance

T_a : Ambient Temperature (used a typical hot weather temperature 38 ℃ ≈ 100°F)

$$T_j = i_{out}(v_{in} - v_{out})R_{\theta JA} + T_a \quad (1.4)$$

$$T_j \approx (1.28\,W)\left(22.8\,\frac{℃}{W}\right) + 38\,℃ \approx 67.184\,℃ \quad (1.5)$$

Looking at the calculated result of Equation (1.5), it is much smaller than the maximum recommended junction temperature for the linear regulator which is 125 degrees Celsius stated in its datasheet. [7] Thus, the linear regulator can handle a typical hot weather environment.

As a final note, the worst-case current drawn scenario was calculated from Table 5.

Table 5. Worst-Case Current Drawn at Linear Regulator Output

Component	**Current drawn (worst case) @ 3.3 V**
STM32H743VIT6	700 mA
LC Display	1.5 mA
microSD	50 mA
STM32 Debugger	< 1 mA

3. Cost and Schedule

3.1 Cost Analysis

3.1.1 Labor Cost

For each partner, the typical hourly rate is $50 per hour. Thus, $150 per hour is the hourly rate for all three people in the group. As for the hours needed to complete the project, about an average 20 hours per group member per week for the entire next eight weeks which is a total of 480 hours. As a result, the total labor cost after multiplying by a 2.5x overhead multiplier would be $180000.

3.1.2 Parts Cost

All the parts needed for the project design are listed below in Table 6.

Table 6. Project Parts Cost and Description

Manufacturer (Purchase link hyperlinked)	Part #	Quantity	Cost	Description

STMicroelectronics	STM32H743VIT6	1	$15.83	ARM® Cortex®-M7 STM32H7 Microcontroller IC 32-Bit Single-Core 480MHz 2MB (2M x 8) FLASH 100-LQFP (14x14)
ESC Inc.	ECS-400-8-36B-7KY-TR	1	$0.60	40 MHz ±7ppm Crystal 8pF 30 Ohms 4-SMD, No Lead
CUI Devices	MSD-4-A	1	$0.35	9 (8 + 1) Position Card Connector microSD™ Surface Mount, Right Angle Gold
Delkin Devices, Inc.	S312TLKJM-C1000-3	1	$7.80	Memory Card microSD™ 128MB Class 10, UHS Class 1 SLC
Amphenol ICC (FCI)	87583-3010RPALF	1	$1.08	USB-A (USB TYPE-A) USB 2.0 Receptacle Connector 4 Position Surface Mount, Right Angle
Texas Instruments	LM1085ISX-3.3/NOPB	1	$1.99	Linear Voltage Regulator IC Positive Fixed 1 Output 3A TO-263 (DDPAK-3)
CUI Devices	PJ-102AH	1	$0.70	Power Barrel Connector Jack 2.00mm ID (0.079"), 5.50mm OD (0.217") Through Hole, Right Angle
Newhaven Display Intl	NHD-C0220BIZ-FSW-FBW-3V3M	1	$11.41	Character Display Module Transflective 5 x 8 Dots FSTN - Film Super-Twisted Nematic LED - White I2C 75.70mm x 27.10mm x 6.80mm
Triad Magnetics	WSU050-4000	1	$13.32	5V 20 W AC/DC External Wall Mount (Class II) Adapter Fixed Blade Input
C&K	D6R90 F2 LFS	1	$1.53	Pushbutton Switch SPST-NO Keyswitch Through Hole (0.1 A 3V)
SparkFun Electronics	PRT-12796	Bulk	$2.10	Jumper Wire Female to Female 6.00" (152.40mm) 28 AWG
Würth Elektronik	61300311121	Bulk	$0.13	Connector Header Through Hole 3 position 0.100" (2.54mm)
CNC Tech	3220-10-0300-00	1	$0.78	Connector Header Surface Mount 10 position 0.050" (1.27mm)
Seeed Technology Co., Ltd	114991786	1	$8.70	STM8, STM32 - Debugger (In-Circuit/In-System) SIPEED USB-JTAG/TTL RISC-V DEBUG

Raspberry Pi	SC0022	1	$35.00	Raspberry Pi 3 Model B Single Board Computer 1.2GHz 4 Core 1GB RAM ARM® Cortex®-A53, VideoCore
Raspberry Pi	RPI-CAM-V2	1	$35.00	Sony IMX219 image sensor custom designed add-on board for Raspberry Pi
GlobTek, Inc.	USBAM11851M2MICBB K5A	1	$6.08	Cable A Male to Micro B Male 3.94' (1.20m) Shielded
Texas Instruments	LM386N-1/NOPB	1	$1.28	Amplifier IC 1-Channel (Mono) Class AB 8-PDIP
Taoglas Limited	SPKM.15.8.A	1	$1.74	8 Ohms General Purpose Speaker 500 mW 10 Hz ~ 11 kHz Top Round
Texas Instruments	TPS259814ARPWR	1	$1.37	Electronic Fuse Regulator High-Side 10A 10-VQFN-HR (2x2)
Stackpole Electronics Inc	CF14JT10K0	10	$0.53	10 kOhms ±5% 0.25W, 1/4W Through Hole Resistor Axial Flame Retardant Coating, Safety Carbon Film
Cal-Chip Electronics, Inc.	GMC02X5R105M10NT	3	$0.30	1 µF ±20% 10V Ceramic Capacitor X5R 0201 (0603 Metric)
Samsung Electro-Mechanics	CL10A475KP8NNNC	2	$0.20	4.7 µF ±10% 10V Ceramic Capacitor X5R 0603 (1608 Metric)
Murata Electronics	GRM0335C1H2R0BA01D	2	$0.20	2 pF ±0.1pF 50V Ceramic Capacitor C0G, NP0 0201 (0603 Metric)
Samsung Electro-Mechanics	CL10B225KP8NNNC	4	$0.40	2.2 µF ±10% 10V Ceramic Capacitor X7R 0603 (1608 Metric)
Samsung Electro-Mechanics	CL05B104KP5NNNC	10	$0.15	0.1 µF ±10% 10V Ceramic Capacitor X7R 0402 (1005 Metric)

After summing the cost for each part, the total part cost is **$148.63.**

3.1.3 Grand Total Cost

The total cost for this project is calculated below in Table 7.

Table 7. Individual Costs and Total Cost

Labor Cost	$180000
Part Cost	$148.37
Total Cost	$180148.37

3.2 Schedule

A weekly schedule about how tasks will be divided up between each team member is shown in Table 8.

Table 8. Weekly Schedule

Week	Task	Person
February 26	Order All	Everyone
	Assign high level subsystems to each member	Everyone
	Schematic Completed	Sergio
	Install prerequisite software on Raspberry Pi	Sarthak
	Finalize 5 audio effects for our project	Zachary
March 4	PCB Done & audited	Sergio
	Be able to view camera from Pi	Sarthak
	Write/Test DSP code on computer for 5 effects	Zachary
March 11	Spring Break	
March 18	Test power system	Sergio
	Test Audio System	Zach
	Be able to detect a hand	Sarthak
March 25	Begin coding connections between the chip and the	Sergio

	display/Sd card	
	Order PCB	Everyone
	Write at least one effect on the microcontroller	Zach
	Successfully detect a single hand gesture and be able to output information to a breadboard	Sarthak
April 1	Hand gestures work	Sarthak
	Finish/Test display SD card code	Sergio
	Integrate SD card & speaker	Zach
April 8	Integrate Software with microcontroller	Sarthak
	3d design enclosure	Sarthak
	Buffer time for anything above not done	Sergio & Zach
April 15	Soldering	Sergio
	Additional Buffer Time	Everyone
	Print 3d enclosure	Sarthak
April 22	Final demo prep	Everyone
April 29	Final Report Work	Everyone

4. Discussion of Ethics and Safety

There were no immediate ethical and safety implications of this project. However, for the sake of being comprehensive, potential scenarios will be addressed below.

The first is the IEEE ethics code II.7: “to treat all persons fairly and with respect, and to not engage in discrimination based on characteristics such as race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression”. Since this project can be applied to help those with certain disabilities and could be even necessary for someone with a disability to DJ or work with audio effects. As a result, the price would be raised incredibly high to make it unaffordable for many(similar to how prescription drugs like insulin are prices) due to it being a necessity. We understand how unethical this would be and if we decided to sell this product we would ensure its pricing is fair to all

those who need it.

The second is IEEE ethics I.1: “1. to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to promptly disclose factors that might endanger the public or the environment;”. What specifically stands out here is the privacy issue since we are using a camera. Although we have no reason to record the camera other than a live feed, a malicious party could potentially try to use the camera to invade privacy. We will address this issue by not connecting any parts of this device to the internet meaning there is no chance of an “attack”, All camera processing will be local to the unit.

Lastly, is IEEE ethics II: To treat all persons fairly and with respect, to not engage in harassment or discrimination, and to avoid injuring others. Our product has the potential to cause harm as it has a speaker which could potentially damage someone's eardrums should the audio be malicious or too loud. We will ensure that only audio files put into the device will be played and that those files will not be harmful. Furthermore, we will ensure the speaker we are using has no possibility of getting loud enough to cause any hearing damage.

5. Citations

[1] IEEE. “IEEE - IEEE Policies: Section 7 - Financial Operations, Article 8 - 7.8 IEEE Code of Ethics.” IEEE, www.ieee.org/about/corporate/governance/p7-8.html.

[2] Raspberry Pi Foundation. "Raspberry Pi Camera Module v3 Product Brief." [Online]. Available: https://datasheets.raspberrypi.com/camera/camera-module-3-product-brief.pdf

[3] Raspberry Pi Foundation. "Remote Access." [Online]. Available: https://www.raspberrypi.com/documentation/computers/remote-access.html

[4] STMicroelectronics. "STM32H743BI Datasheet." [Online]. Available: https://www.st.com/resource/en/datasheet/stm32h743bi.pdf

[5] Digi-Key Electronics. "WSU050-4000 Datasheet." [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/328/114991786_Web.pdf

[6] Triad Magnetics. "WSU050-4000 Datasheet." [Online]. Available: https://catalog.triadmagnetics.com/Asset/WSU050-4000.pdf

[7] Texas Instruments. "LM1085ISX-3.3-NOPB Datasheet." [Online]. Available: https://www.digikey.com/en/products/detail/texas-instruments/LM1085ISX-3-3-NOPB/366710?s=N4IgTCBcDaIDIFkCMAGAHAVgJIGUAaAtAMwB0RA9AHIDyACgEIgC6AvkA.

[8] Taoglas, "15 mm Miniature Speaker Datasheet," 2019. [Online]. Available: https://www.taoglas.com/datasheets/SPKM.15.8.A.pdf.

[9] Texas Instruments, “TPS259814ARPWR Datasheet.” [Online]. Available: https://www.ti.com/lit/ds/symlink/tps25981.pdf?ts=1694724042849&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTPS25981

[10] Raspberry Pi, “Raspberry Pi 3 Model B Datasheet. ” [Online}. Available: https://us.rs-online.com/m/d/4252b1ecd92888dbb9d8a39b536e7bf2.pdf