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".

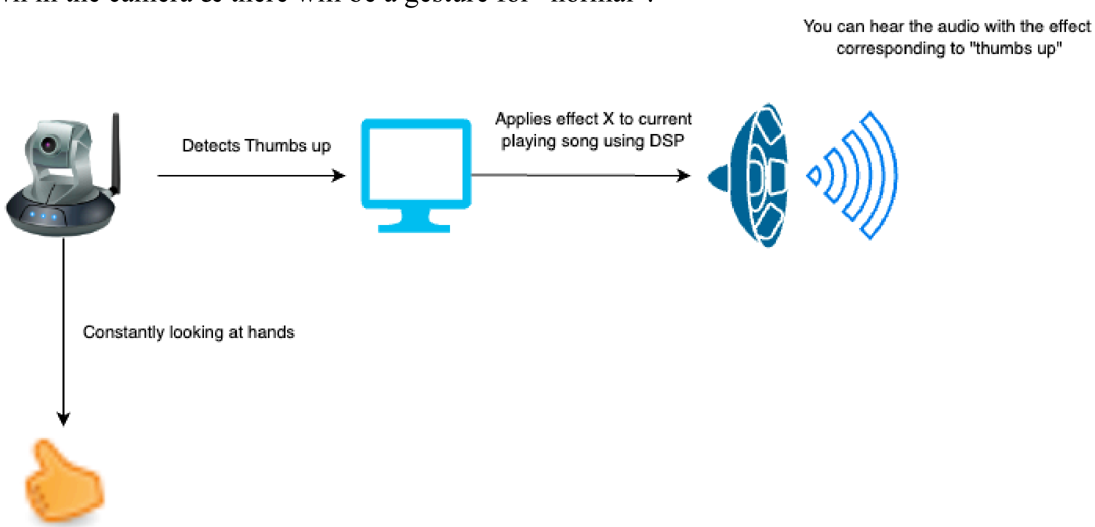

Figure 1. High-Level Overview of the Hands Gesture Audio Effects System

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

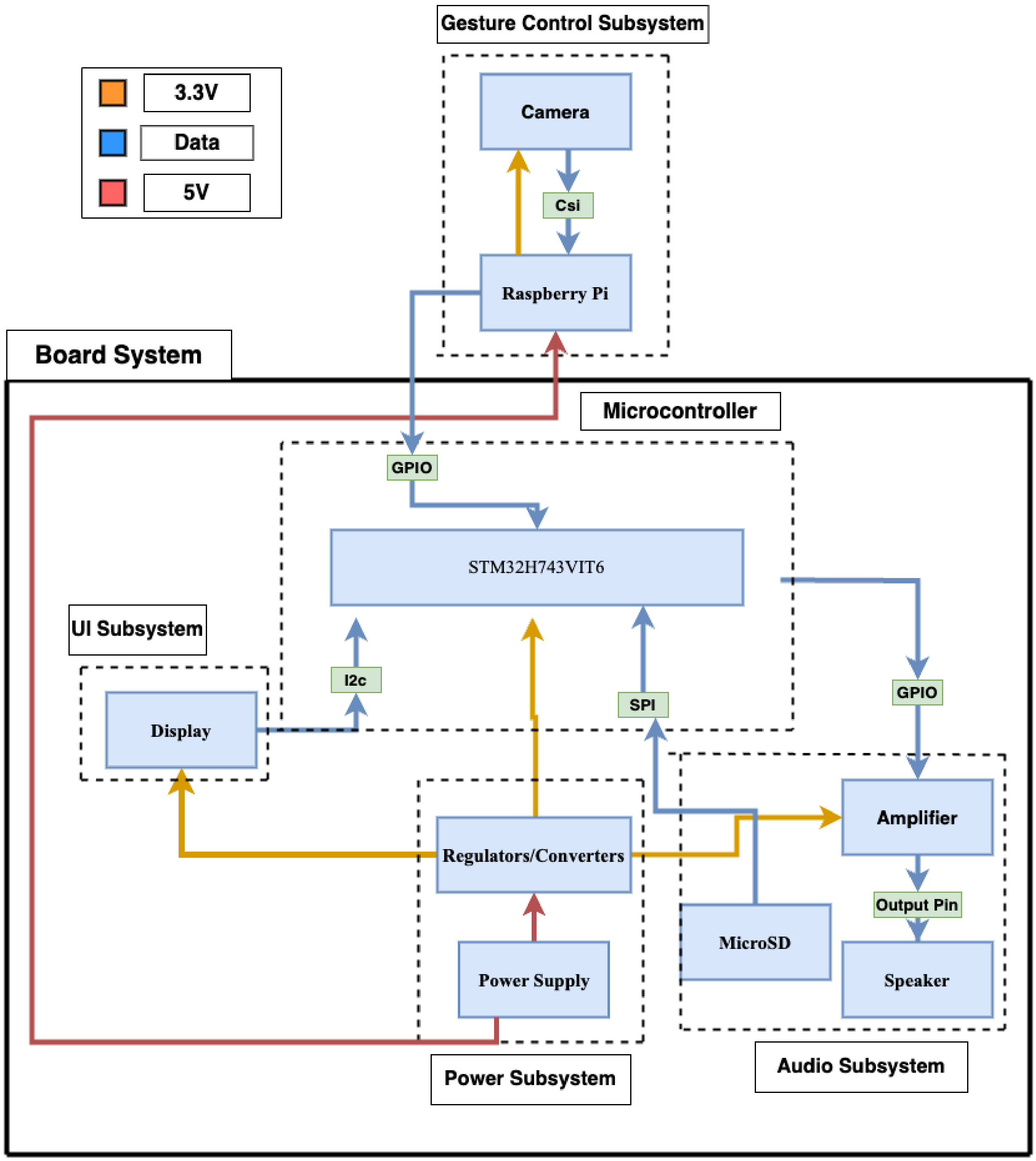

Figure 2. Block Diagram of the Hands Gesture Audio Effects System Design

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.

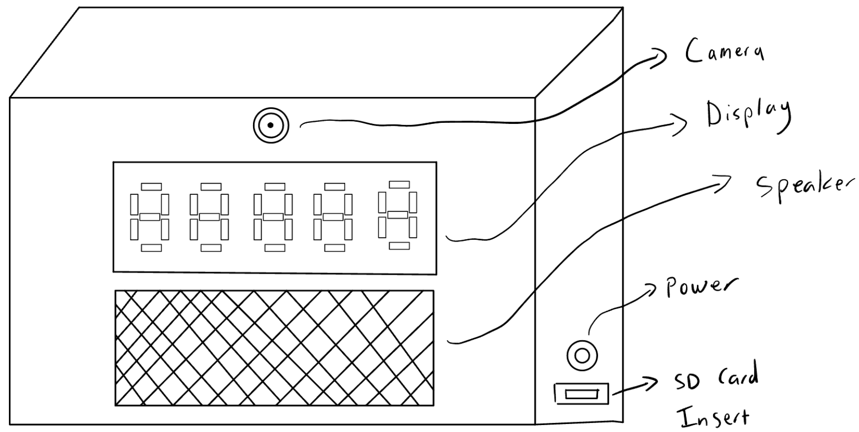

Figure 3. Initial vision for physical design

2.3 UI 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.

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.	We can detect when the signal from the hand gesture module makes it to the PCB using an LED. We can start a time from when that LED goes on to when the display changes. 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.

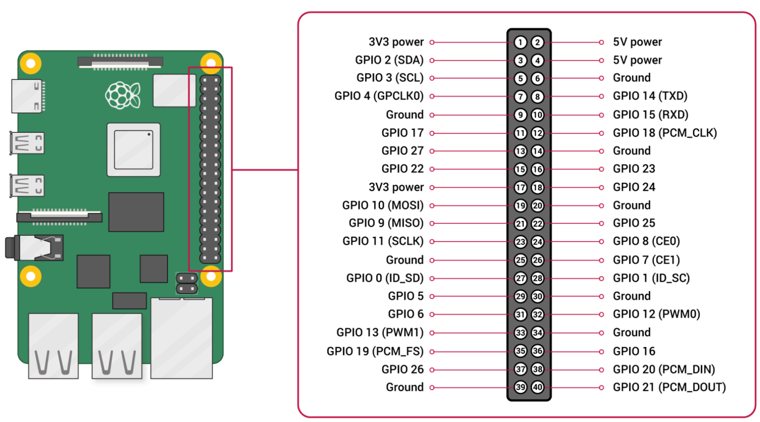

Figure 4. GPIO layout of Raspberry Pi 3B

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. As for the camera module, it was chosen since it was made specifically for the Raspberry Pi board.

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.	One way is to just 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). Then, subtract the variable storing the start time from the variable that stores the end time. Finally, take the reciprocal of that value to get frames 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.	This parameter can be timed using a regular timer (stopwatch). Moreover, the timer starts when the user makes a hand gesture and it will stop when the intended audio effect has been applied to the audio input.
Raspberry Pi must notify the microcontroller to apply an audio effect, 0.2 seconds (at most) after a unique hand gesture corresponding to the audio effect has been detected by the gesture detection algorithm	We can get a timestamp on the Pi of when a gesture is detected and capture a timestamp inside of the STM32H743VIT6 when it detects a change. We can use the difference between these two to see if it fits within the criteria.

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 4.5A wall power adapter) that will connect to a DC power jack. The output of the DC power jack is filtered by a capacitor. Then, the filtered 5V power source will connect to a USB-A connector and the input of a linear regulator. The USB-A connector is required for the microUSB-to-USB cable that powers the Raspberry Pi. On the other hand, the fixed output voltage (3.3V) of the linear regulator will be used as the power supply for all the components in the custom PCB. Also, there will be a test point at the filtered 5V node and at the fixed 3.3V so the ripple of each voltage can be observed for verification.

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.	Using an oscilloscope, connect to the 3.3 V test pin and measure the voltage such that the output voltage stays within 0.5% of 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 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 4.5 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. 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 1.8 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 4.5 A to the entire system. Although this supply current is bigger than what is needed (~4.3 A), it is fine just in case the needed current is slightly above the estimation.

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

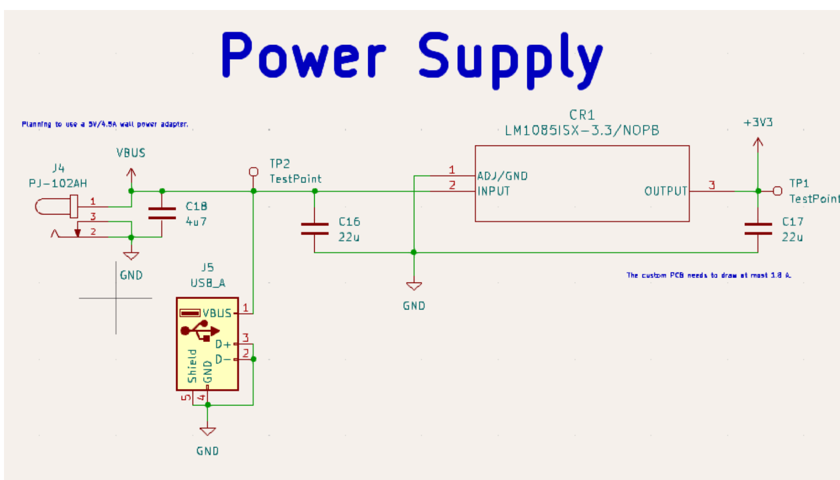

Figure 5. Power Supply Circuit Diagram

As illustrated by Figure 5, the wall adapter will connect to a DC power barrel jack. The input voltage ripple will be minimized by the capacitor at the jack pins and the capacitor at the input of the linear regulator. The voltage ripple will be minimized again at the output 3.3 voltage of the linear regulator. One thing to be noticed is that the data lines of the USB-A connector were set to ground since the only purpose is powering the Raspberry Pi (not transferring data).

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.

Delay: Creates an echo effect by replaying the audio signal after a short period.

Gain Adjustment: Varies the amplitude of the audio signal, controlling volume.

Low-pass Filter: Attenuates frequencies above a cutoff, reducing high-frequency noise or brightness.

High-pass Filter: Removes low-frequency components, reducing rumble or bass.

Table 4. Audio Subsystem RV Table

Requirement	Verification
The sound being outputted by the speaker must be less than 80 dB for safety purposes	-Connect the audio source to the input of the Audio Subsystem. -Set the audio source to produce a continuous sine wave signal at 1 kHz. -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. -Gradually increase the volume of the audio source until the sound level meter stabilizes. -Record the maximum sound pressure level (SPL) reading displayed by the Sound Level Meter. -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	-Connect the audio source to the Audio Subsystem's input and the output to both the Audio Analyzer and headphones/speakers. -Set the audio source to produce a clean sine wave signal at 1 kHz. -Apply the desired audio effect using the microcontroller within the Audio Subsystem. -Conduct a subjective listening test by playing the processed audio through headphones/speakers and noting any static noise or undesired artifacts. -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. Using the worst case scenario current drawn by the PCB (~1.8 A) 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.

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

$$P_{D,worst\ case} \approx (1.8\ A)(5\ V - 3.3\ V) \approx 3.06\ W \quad (1.2)$$

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

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 5.

Table 5. 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
MEAN WELL USA Inc.	GSM36U05-P1J	1	$18.31	5V 23 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
ShenZhen LvXiangYuan Technology Co Ltd	LX18AA-050300-ZU	1	$12	MakerSpot UL-Listed Power Supply 5V3A with Switch 2.5A 2A 1.5A 1A Fast Charge AC Adapter w/ 1.5m Extra Long On Off Power Switch Micro USB Cable for Raspberry Pi 3 Model B/B+
ALLECIN	LM386N	1 pack of 10	$7	8-pin power amplifier IC/Wide supply voltage range:4V–12V or 5V–18V. Voltage Gains from 20 to 200.
uxcell	a13062500ux0495	Pack of 10	$7	PCB Mount Stereo Jack;Type : 3.5mm Female Socket(1/8")
KUIDAMOS	B0BM996CD9	1	$9	3.5mm External Speakers Portable 3.5mm Speaker,Plug in Speaker with Audio Cable Energy Saving Speaker with 3.5mm Jack Wired Speaker Mini Speaker for MP3, MP4, Computer, 60mm Diameter Speaker
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	CL21A226MQQNNNE	2	$0.26	22 µF ±20% 6.3V Ceramic Capacitor X5R 0805 (2012 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 **\$178.35.**

3.1.3 Grand Total Cost

The total cost for this project is calculated below in Table 6.

Table 6. Individual Costs and Total Cost

Labor Cost	\$180000
Part Cost	\$178.35
Total Cost	\$180178.35

3.2 Schedule

A weekly schedule about how tasks will be divided up between each team member is shown in Table 7.

Table 7. 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	Zachary

	computer for 5 effects	
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 display/Sd card	Sergio
	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.

5. Citations

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.