Software Instruments via IR Sensing

ECE 445 Design Document

Colin Flavin, Helen Swearingen, Nick Russo

Group 33

TA: Eric Liang

1. Introduction

1.1. Objective

Music and movement are big contributors of emotional expression. Traditionally, music is dictated by trained musicians, whose interaction with instruments requires dexterity and coordination, as well as the interaction with a physical instrument. This forces composition and choreography to be separated. The goal of this project is to bridge this gap between composition and choreography via a wireless instrument using IR sensors to track a person's motion.

1.2. Background

Wireless instruments were first introduced through the theremin. The theremin was able to control pitch based on the distance a players' hand was from the pitch antenna. This created a variable capacitance, which would translate to different frequencies in an oscillator. That was then fed into a heterodyne system with a fixed frequency oscillator, which was then heard as the instrument's pitch [1]. However, this had problems, because it was extremely difficult to control the instrument's pitch.

In a completely separate set of circumstances, composer John Cage collaborated with Merce Cunningham to create an instrument based on dance, through the use of 12 antennas to sense the proximity of a dancer, which would then play different notes [2]. This was the first attempt at the coordination of composition and choreography.

This design project aims to combine the ideas of wireless instruments, with the bilinear nature of choreography and composition as seen through Cage's work. In addition, this project differs fundamentally from the IMUsic project. Rather than using wearables to track motion, IR sensors will track hand gestures. Also, rather than mixing the music on a computer with software, our plan is to mix the sound with a custom made mixer.

1.3. High-Level Requirements

1.3.1. Must be able to register hand gestures and correctly map them to expected motions with preset effects.

1.3.2. Must be able to process gestures, mix sounds accordingly, and play them back with a delay of 100ms or less between performed gesture and the played sound.

1.3.3. The output frequency range should be able to play at least two octaves of notes with a dynamic range of 40-70dB.

2. Design

We have three major areas to this project: interfaces with the outside world, gesture processing, and sound mixing. First, off the shelf IR sensors (a Leap Motion Controller) senses the player's hand motions. This information is sent to the computer, where this data is processed and translated to a set of musical tones. From there, this information is sent to a custom built sound mixer for better sound fidelity, before being sent to speaker sets.

2.1. Physical Diagram

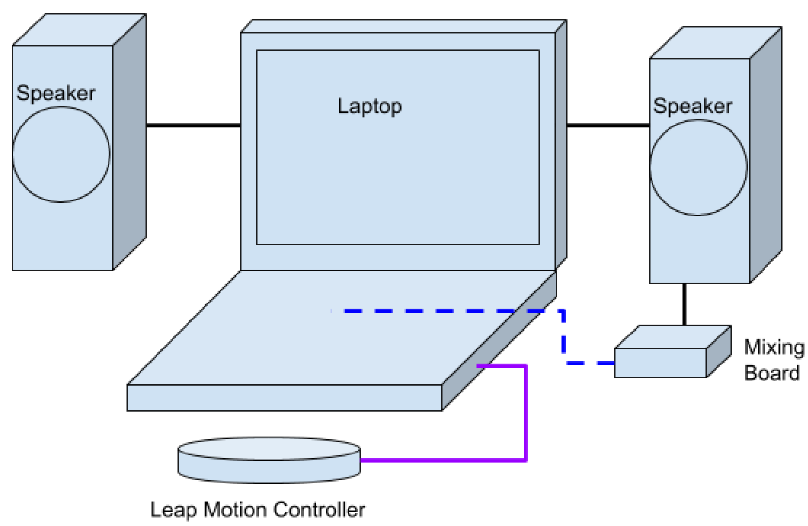

Figure 1: Physical Diagram

A Leap motion controller would sit in front of and connect to a laptop, which would wirelessly connect to the sound mixer on the right, which would process and produce audio signals to be sent to the speakers.

2.2. Block Diagram

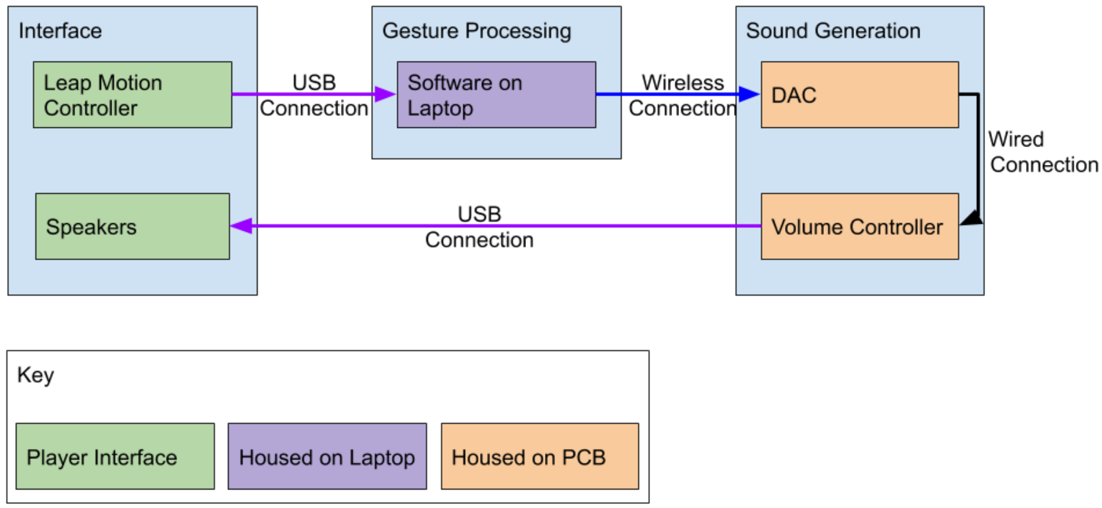

Figure 2: Block Diagram

2.3. Functional Overview

2.3.1. Interfaces

The player must be able to interact with the project: to give it input, and hear the results.

2.3.1.1. Leap Motion Controller

The Leap Motion Controller detects the player's hand motions. It communicates with the laptop via USB. To play the instrument well, 13 unique hand positions must be identifiable so that each possible note in a chromatic can be played.

Requirement	Verification
It must be able to detect at least 13 different hand positions on each hand	A. Set up motion controller with the laptop B. Make 13 different hand positions, write software to detect the hand position, and verify 13 different values occur

Table 1: Motion Controller RV

2.3.1.2. Speakers

The speakers play the sounds with given tones, volume and frequencies from the player, as sent from the sound mixer. This will be an off the shelf speaker(s), so no further work should be required other than plugging it into the sound mixer and power.

Requirement	Verification
It should have a dynamic range of 40-70dB with a tolerance of 5%	A. Setup speakers with a computer B. Send a 40dB sound through speakers and ensure it is within tolerance using a decibel meter C. Repeat with a 70dB sound
It should be able to play two octaves, from 200Hz to 10kHz with tolerance of 1%	A. Setup speakers with a computer B. Send a 200Hz sound through the speakers and ensure it is within tolerance with a musical tuner C. Repeat with a 10kHz sound

Table 2: Speaker RV

2.3.1.3. Volume Controller

The user should have dynamics control through the instrument, to differentiate loud and soft notes. However, in different environments, the user might want a different master volume to control how loud a normal pitch would be. This would be controlled through a dial on the sound generation box.

Requirement	Verification
Max volume allowed should be 70dB	A. Send a signal of over 70dB (roughly 85dB) through the sound generation module B. Set volume controller to max volume and ensure with a decibel meter that the sound never surpasses 70dB.

Table 3: Volume Controller RV

2.3.2. Gesture Processing

Once hand gestures are sensed by the Leap motion controller, this information will be sent to a laptop. The laptop then needs to process this to convert a hand gesture to a pitch with frequency, timbre, and volume. From there, it will communicate with a sound generation module via bluetooth for further processing and conversion to an audio signal.

2.3.2.1. Laptop

Collecting gesture data and mapping detected motions to ones the project will use will happen on a laptop, due to the Leap Motion Controller's system requirements.

Requirement	Verification
It must be able to take an identified hand gesture and map it to a given frequency, timbre, and dynamic	A. Setup Leap Motion controller B. For every right hand gesture, ensure the correct frequency is the output C. For every left hand gesture, ensure the correct timbre is the output D. For multiple distances from the controller, ensure different dynamics are displayed

Table 4: Laptop RV

2.3.3. Sound Generation

When the laptop decides the pitch and volume of the sound, this is all done in the digital space. This module receives this data, and then converts it to an analog audio signal, which is then placed through potentiometers for a master volume control.

2.3.3.1. Bluetooth Module

Not all laptops have more than one USB port. To compensate this, we plan on making a wireless connection to our sound generation system via an ESP32 microcontroller.

Requirement	Verification
The sound generator must be able to communicate with a laptop via Bluetooth	A. Establish connection between ESP32 and laptop B. Send a random block of data to the ESP32 and verify that it is correct.

Table 5: ESP RV

2.3.3.2. D/A Converter

The D/A converter takes in the digital data sent by the Bluetooth chip and translates it into analog sound data.

Requirement	Verification
The D/A converter must be able to convert a 16 bit digital signal to an analog signal within 2% tolerance	A. Send a 16 bit sinusoid signal to the ESP32, test through simulation what the sinusoid should be. B. Through a test point on PCB, measure the voltage after the D/A converter and verify the signal is within tolerance.

Table 6: DAC RV

2.3.3.3. Power Module

The sound generation box is the only thing that does not have its own power module to come with it. We plan to power this box using a rechargeable 5V lithium ion battery, with the voltage stepped up or down using a logic level converter.

Requirement	Verification
Battery must supply 5±0.5V at max 2.4A for at least one hour.	A. Measure the open-circuit voltage and ensure it doesn’t surpass 5.5V with a voltmeter B. Use resistor load to force 2.4A from the battery, check with an ammeter C. Leave for an hour and ensure power specifications can still be supplied by checking voltage and current with voltmeter and ammeter

Table 7: Power RV

2.4. Schematics

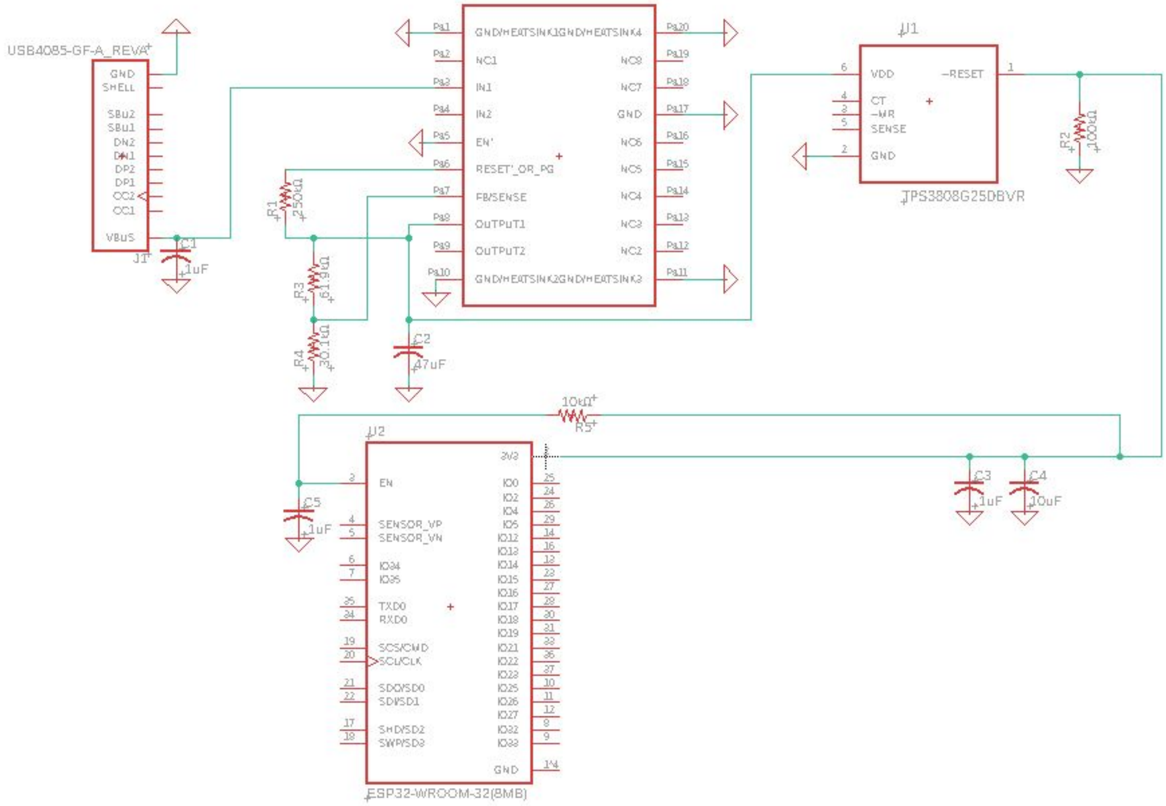

Figure 3: Power and voltage regulation subcircuit

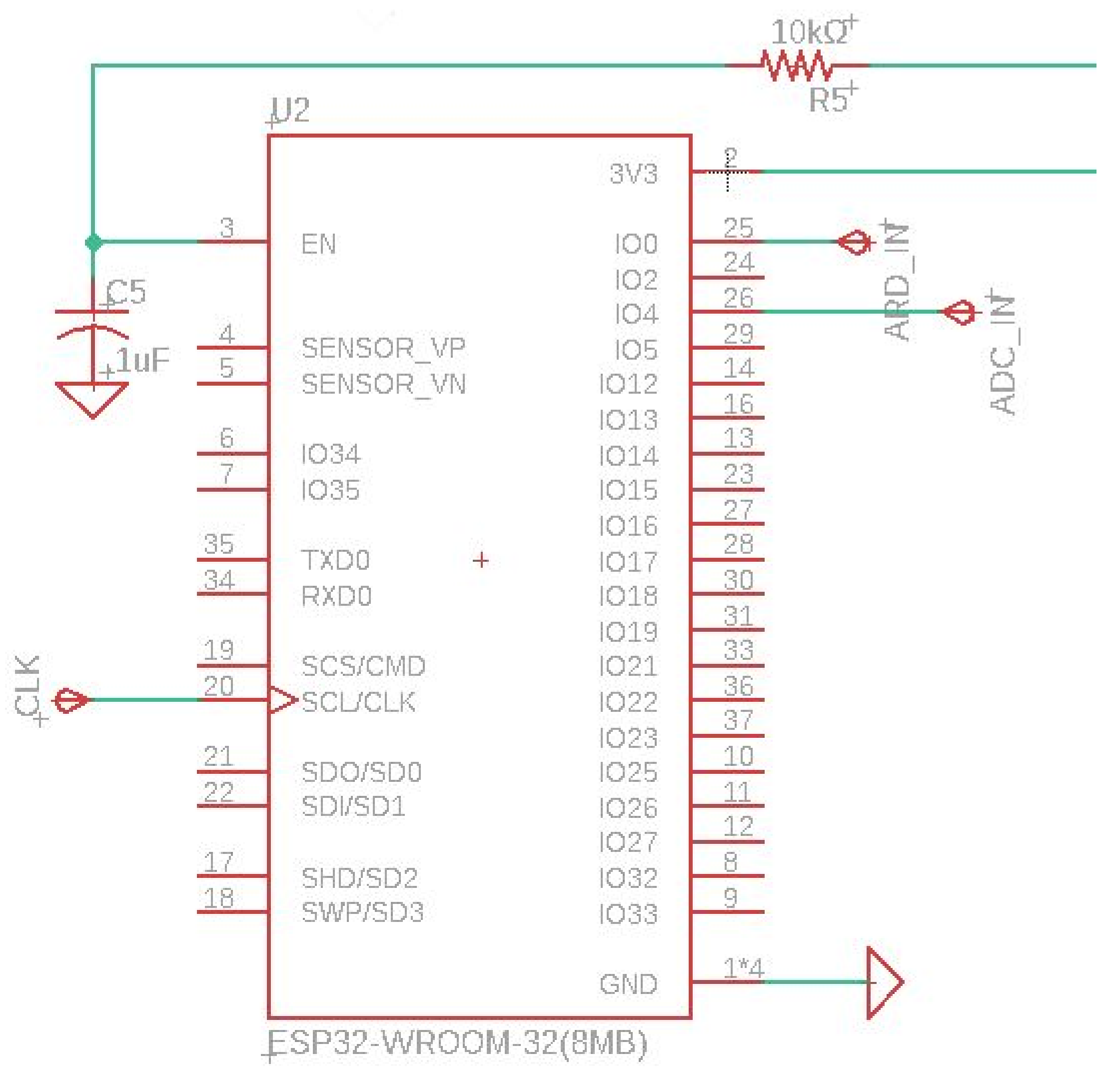

Figure 4: ESP connections

2.5. Software

This project relies heavily on software in several areas. First, the laptop must be able to take the information from the Leap Motion Controller, and be able to map the gestures the user is creating to a given note, timbre, and dynamic. This information must then be used to synthesize an audio signal, which will be done using SuperCollider. Finally, this audio signal must be transmitted to the sound generation module via Bluetooth.

2.5.1. Gesture Mapping

In this portion, the laptop will poll for data from the motion controller. As it receives data about the left and right hands, it will do a table lookup to the corresponding timbre, as denoted by the left hand, and corresponding pitch, as denoted by the right hand. The information about the x and y position of each hand will correspond to a discretized dynamic in the table as well. This information will then be sent to the SuperCollider for the synthesis of the note. This will then repeat so long as the instrument is on.

2.5.2. SuperCollider

SuperCollider is a program that will allow us to interface our hardware to the computer, pick notes, and generate sound. Our code should be taking in three different kinds of data from the Leap controller from the two hands. From the right hand, it should take in the gesture performed, and from the left hand, it should be taking in x and y position relative to the Leap. Supercollider will take in the gesture from the right hand and use that to pick the timbre or quality of the sound (i.e., the instrument). The x-position of the left hand will be used to control the pitch, with each pitch taking up 1cm of space relative to the center of the Leap interaction box, for a total of two octaves of pitch range in the optimal Leap range. The notes are discrete between the 1cm sections. The y position will control the note dynamic, with a dynamic range of 40-70dB. This will be a continuous operation, with 40dB corresponding to the bottom of the interaction box at 82.5mm above the Leap and 70dB at 317.5mm above the interaction box.

2.5.3. Bluetooth Communication

The ESP32 has in-built Bluetooth, so this portion is fairly straightforward. Prior to any connections, the ESP32 will be programmed with the Bluetooth BLE already written for ESP32 Bluetooth connection. The device can then be paired with the laptop. Afterwards, the laptop will send the audio data to the ESP32 to be sent through the DAC.

2.6. Tolerance Analysis

The most critical part of this project is going to be the gesture tracking and hand mapping of the Leap controller. The Leap controller maps points to the millimeter in x, y, and z, and has an inverted pyramid field of view. [3] We would need to experiment with the sensor to find comfortable, natural interaction boxes for both hands that minimize one hand accidentally blocking the sensor's view of the other.

3. Project Differences

3.1. Overview

In the previous solution to this project, they implemented four wearable devices for both wrists and both ankles. These were outfitted with an IMU with accelerometers, gyroscopes and magnetometers to track a user's motion. This information was then sent via WiFi to a software synthesizer to create the music.

Our implementation is quite different. There are no wearable devices, and it tracks hand motion via an IR sensor. These motions are associated with pitches and volumes in a table, and are sent to be synthesized in SuperCollider before final trimming of the audio signal on our own PCB.

Our project has no wearables. This makes our design less mobile than their dance geared solution. However, our solution tracks hand movement more precisely and in a more compact amount of space. In addition, our solution will be subject to less interference.

3.2. Analysis

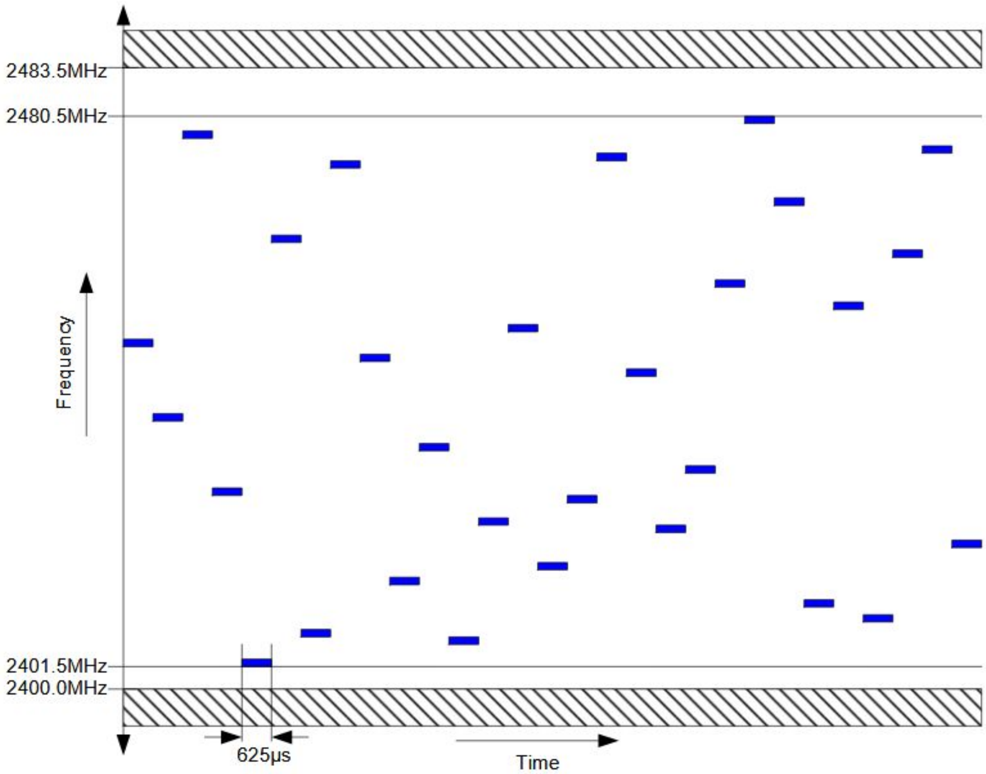

Figure 3: Bluetooth Frequency Occupancy Example [4]

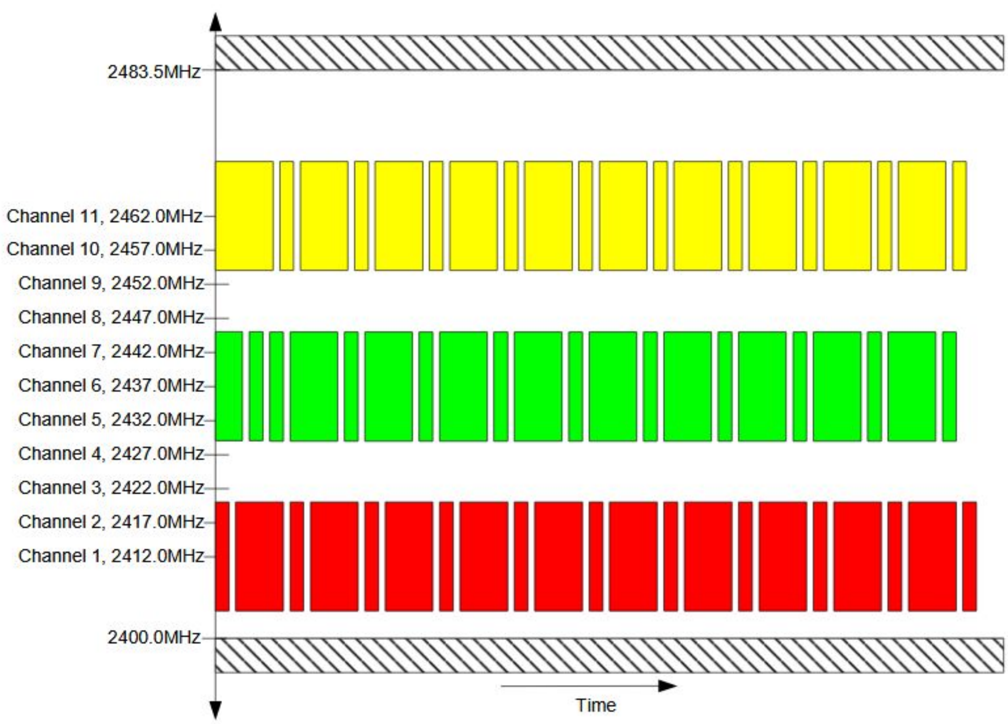

Figure 5: Frequency Occupancy of Three Wifi Networks [4]

Since our design rests on a table and relies only on short-range wireless transmissions, it is subject to less interference than the previous solution. The dance implementation used four wifi transmitters, attached to the dancer's wrists and ankles. Those signals have the potential for interference from each other and from surrounding wifi networks, since wifi networks operate on one of a few fixed sections of the 2.4GHz band. Since Bluetooth signals jump between many, narrower bands, they are much less susceptible to interference from other networks: they will not be constantly overlapping with, for instance, the sections of the band used by preexisting local networks. The base and receiver's proximity to each other will also minimize any interference that does occur, as our signals will still be strong.

4. Costs

Our development costs are estimated at around $40/hour, 10hours/week, for 3 people. We consider there are 10 weeks left in the course to complete the project. This neglects weekly TA meetings, and comes to $12,000 in labor costs.

3 x $40/hour x 10 hour/week x 10 weeks = $12,000

Part	Cost
LeapMotion Controller (UltraLeap)	$89.95
ESP32 Wifi Module (ESP)	$3.75
Anker PowerCore Battery (Anker -- Amazon)	$25.99
Total:	$119.69

Table 8: Costs

5. Schedule

Week #	**Nick Russo**	**Colin Flavin**	**Helen Swearingen**
1	Finish D/A converter schematics	Finish schematics and start PCB design	Research Leap Motion Controller
2	Finish PCB Design	Finish PCB design	Research coordinate and gesture mapping for Leap controller
3	Research Bluetooth communication system	Research coordinate mapping for Leap	Program basic gestures

		controller	
4	Begin coding Bluetooth communication system	Research coding in SuperCollider	Program and test communication with SuperCollider
5	Finish coding Bluetooth communication system, test functionality	Code note picking algorithm and sound mapping in SuperCollider	Work on matching human hand motion to preset gestures and variants
6	Debug week for any problems that arise	Debug problems in code or PCB	Continue gesture matching
7	Test at system level	Test code with motion controller	Test with SuperCollider
8	Begin final presentation	Catch up if behind schedule	Catch up if behind schedule
9	Begin working on Final Report	Work on final presentation	Work on final presentation
10	Finish working on Final Report	Finish working on Final Report	Finish working on Final Report.

Table 9: Schedule

6. Safety and Ethics

IEEE code of ethics states that our project should “improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies” [4]. Our project satisfies this as it promotes the bilinearity of the formation of music and dance. In addition, the IEEE code of ethics also states we should seek, offer and accept criticism for our technical work [4]. By participating in this course, going to weekly meetings with course staff, and presenting our ideas to our peers, we are actively following this code, while seeking to improve the project idea with the intent of creating the best possible end result. In addition to ethical concerns, our project also has some safety concerns.

When dealing with audio, one main safety concern is dealing with the listener's hearing. If unbounded, high volumes of noise can cause prolonged hearing loss. According to OSHA, sounds over 120dB can cause the listener immediate pain, while 85dB of sound over prolonged periods of time can sustain permanent hearing damage [6]. Our mixer will have a max volume of 70dB allowed to account for this.

References

[1] "How the Classic RCA Theremin Works", *Tuvalu.santafe.edu*, 2014. [Online]. Available: http://tuvalu.santafe.edu/projects/musicplusmath/index.php?id=30. [Accessed: 02- Apr- 2020].

[2] "Variations V - Merce Cunningham Trust", *Mercecunningham.org*, 2019. [Online]. Available: https://www.mercecunningham.org/the-work/choreography/variations-v/. [Accessed: 02- Apr- 2020].

[3] "Coordinate Systems", *Leapmotion.com*, 2019. [Online]. Available: https://developer-archive.leapmotion.com/documentation/csharp/devguide/Leap_Coordinate_Mapping.html. [Accessed: 13- Apr- 2020].

[4] "Wi-Fi (IEEE 802.11b) and Bluetooth Coexistence Issues and Solutions for the 2.4 GHz ISM Band", *Ti.com*, 2001. [Online]. Available: https://www.ti.com/pdfs/vf/bband/coexistence.pdf. [Accessed: 16- Apr- 2020].

[5] "IEEE Code of Ethics", *Ieee.org*, 2016. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 03- Apr- 2020].

[6] "How Loud is Too Loud?". [Online]. Available: https://www.osha.gov/SLTC/noisehearingconservation/loud.html. [Accessed: 03- Apr- 2020].