

Air Guitar

By

Arturo Arroyo Valencia

Miaomiao Jin

Youngmin Jeon

Proposal for ECE 445, Senior Design, Spring 2026

TA: Eric Tang

May 2026

Project No. 20

Abstract

The Air Guitar is a dual-wearable sensor system designed to provide a portable, tactile alternative to traditional guitars and screen-based virtual instruments. The system addresses the lack of physical mechanics in existing mobile solutions by utilizing two wirelessly synchronized gloves: a Left-Hand "Fret" Glove that maps finger curvature to digital chord profiles and a Right-Hand "Strum" Glove that captures rhythmic timing and velocity.

The device employs ESP32 microcontrollers communicating via the low-latency ESP-NOW protocol to ensure a motion-to-sound delay of less than 30ms. Key components include flex sensors for chord recognition, a 9-axis IMU for motion detection, a potentiometer for volume control and a button acting as a "dead man's switch" for strum triggering. Powered by separate lithium-ion batteries, the system provides at least 2 hours of continuous operation while outputting standard BLE MIDI to external devices.

Contents

1. Introduction.....	5
1.1 Problem.....	5
1.2 Solution.....	5
1.3 High-level requirements list.....	5
1.4 Block Diagram.....	6
2. Design.....	6
2.1 Subsystem Overview.....	6
2.1.1 Subsystem 1: Left Hand.....	6
2.1.2 Subsystem 2: Right Hand.....	7
2.1.3 Subsystem 3: Power System.....	7
2.2 Subsystem Design.....	7
2.2.1 Subsystem 1: Left Hand.....	7
2.2.1a Resistive Sensing Array.....	7
2.2.1b Signal Conditioning & Processing.....	7
2.2.1c Chord recognition mechanism.....	8
2.2.2 Subsystem 2: Right Hand.....	8
2.2.2a Motion Detection.....	8
2.2.2b Strum Trigger Mechanism (Digital Tactile Switch).....	9
2.2.2c Volume Control (Potentiometer).....	9
2.2.3 Subsystem 3: Power System.....	9
2.2.3a Battery Charging.....	9
2.2.3b Load Sharing and Battery Protection.....	9
2.2.3a Voltage Conversion.....	10
3. Design Verification.....	10
3.1 Subsystem 1.....	10
3.2 Subsystem 2.....	11
3.2.1 Motion Detection.....	11
3.2.2 Volume Linearity.....	12
3.2.3 Polling Latency.....	13
3.3 Subsystem 3.....	13
3.3.1 Battery Protection.....	13
3.3.2 Power supply Operation.....	14
4. Costs.....	15
4.1 Parts.....	15
4.2 Labor.....	16
5. Conclusion.....	16
5.1 Ethical Considerations.....	16
5.2 Future Work.....	16

6. References.....	17
Appendix A Requirement and Verification Table.....	18

1. Introduction

1.1 Problem

Traditional musical instruments, particularly electric guitars, are inherently bulky and difficult to transport, limiting where musicians can practice or perform. While software-based "virtual instruments" exist on mobile devices, they lack the tactile "muscle memory" required for authentic performance, as tapping a glass screen does not replicate the distinct physical mechanics of fretting with one hand and strumming with the other. Consequently, there is a significant need for a wearable system that captures the physical kinetics of guitar playing without the physical footprint of the instrument.

1.2 Solution

To address this challenge, we have developed the "Air Guitar," a dual-wearable sensor system that mimics the ergonomics of a real guitar. The system consists of two wirelessly synchronized gloves: a Left-Hand "Fret" Glove that maps finger curvature to digital chord profiles, and a Right-Hand "Strum" Glove that captures rhythmic timing and velocity. By fusing these data streams via the low-latency ESP-NOW protocol and outputting Bluetooth Low Energy (BLE) MIDI to a phone, the device allows users to perform music anytime and anywhere, effectively untethering the musician from the physical constraints of the hardware.

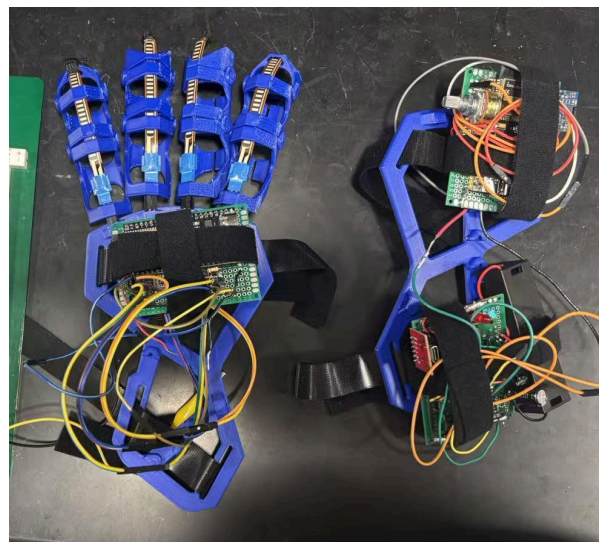


Figure 1. Air Guitar Picture

1.3 High-level requirements list

- Latency: The total "Motion-to-Sound" delay must be under 30ms. Anything higher is noticeable to a musician.
- Chord Recognition: The system must accurately distinguish between at least 5 different chord shapes with a success rate of > 90%.

- **Dynamic Range:** The system must be able to distinguish between a "Soft Strum" and a "Hard Strum," translating that into different speed levels.
- **Battery Life:** The device must operate continuously for at least 2 hours on a single charge.
- **Wireless Stability:** The ESP-NOW link between hands must maintain a Packet Delivery Ratio (PDR) of $\geq 99\%$ within a 2-meter radius (the typical wingspan of a human) over a continuous 10-minute testing window.
 - **Test Method:** The Right-Hand unit and the Left-Hand unit will send 1,000 packets at the target rate (e.g., 100Hz). The laptop will log the sequence numbers; a successful test results in ≤ 10 missed packets.

1.4 Block Diagram

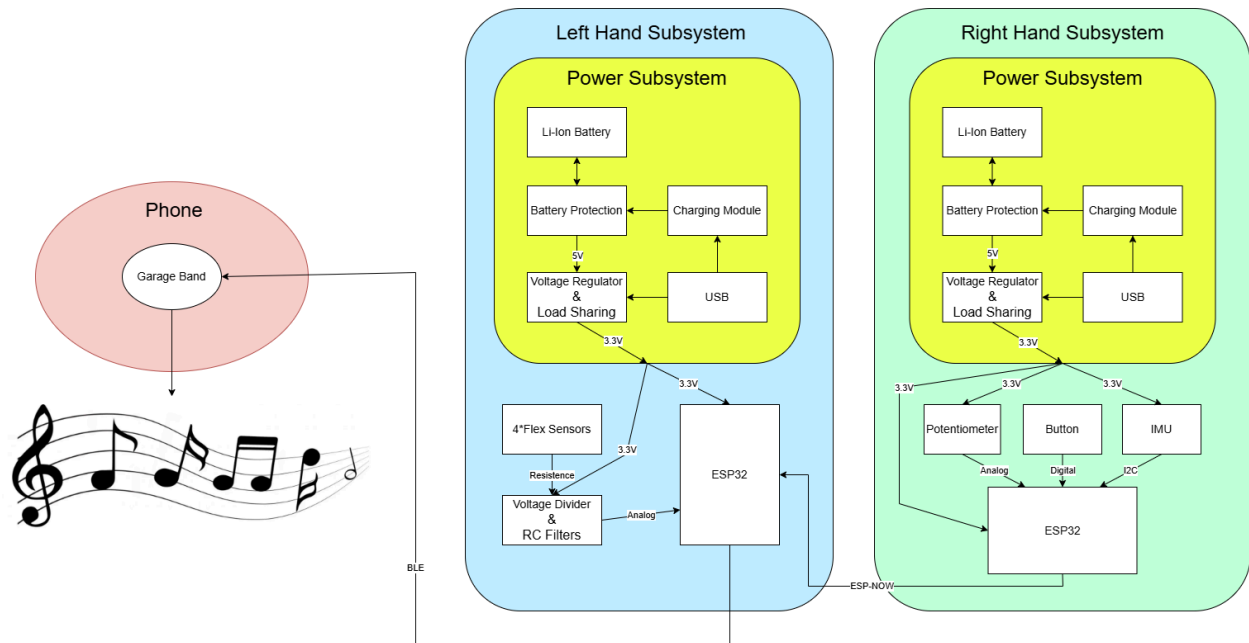


Figure 2. Air Guitar Block Diagram

2. Design

2.1 Subsystem Overview

2.1.1 Subsystem 1: Left Hand

This subsystem identifies the chord the user is trying to play. It maps the curvature of each finger to a specific digital profile (e.g., specific bend angles = C Major). It utilizes four resistive flex sensors mounted along the Index, Middle, Ring, and Pinky fingers. Because the ESP32-WROOM-32E microcontroller cannot measure resistance directly, each sensor is integrated into a Voltage Divider Circuit. This converts the mechanical bending of the finger into a variable analog voltage V_{out} suitable for the ESP32's 12-bit Analog-to-Digital Converter (ADC).

2.1.2 Subsystem 2: Right Hand

This subsystem functioned as the rhythmic trigger for the Air Guitar, determining when a note was played and controlling its intensity. It integrated three primary components to capture user intent: a BNO055 9-axis IMU to detect the precise timing and direction of strumming motions, a digital tactile switch on the thumb that acted as a "dead man's switch" to prevent accidental triggers, and a potentiometer to provide manual, adjustable volume control. The subsystem was driven by a dedicated ESP32 microcontroller, which processed these sensor inputs in real-time and transmitted valid strum events to the Left-Hand Glove using the low-latency ESP-NOW wireless protocol.

2.1.3 Subsystem 3: Power System

This system will be integrated into both of the prior subsystems and will be the main form of power to both the left and right gloves. Therefore, each of the gloves will have their own separate lithium-ion battery. These batteries will be equipped with their own battery management system composed of a charging unit with a buck converter or LDO to ensure a steady voltage of 3.3 volts to the ESP32's on each glove, as well as any other components utilized in attaining the MIDI signal for a specified chord of the user choosing.

2.2 Subsystem Design

2.2.1 Subsystem 1: Left Hand

2.2.1a Resistive Sensing Array

This layer captures the raw physical deformation of the fingers.

Hardware components:

- Flex Sensors (4x): These 2.2" sensors increase in resistance as they bend.
- Voltage Divider Network: A series of 22kΩ high-precision resistors (0.1% tolerance) used to convert the resistance changes into a voltage range optimized for the ESP32's ADC (0.5V to 2.5V). The below voltage divider equations and values are used.

$$V_{out} = V_{cc} \cdot \left(\frac{R_1}{R_{flex} + R_1} \right)$$

Where

$$V_{cc} = 3.3V$$

$$R_{flex, flat} \approx 10k\Omega$$

$$R_{flex, bent} \approx 30k\Omega$$

2.2.1b Signal Conditioning & Processing

Each of the four ADC channels is equipped with a passive RC filter. This serves two purposes:

- Anti-Aliasing: Ensuring the signal bandwidth is lower than half the sampling rate ($f_s/2$).
- Noise Suppression: Filtering out the high-frequency 2.4GHz "burst noise" generated by the ESP32's Wi-Fi/Bluetooth antenna, which can otherwise cause "jitter" in the chord detection.

We choose a cutoff frequency (f_c) that is high enough to capture rapid finger movements (shredding/fast chord changes) but low enough to block electrical interference.

Hardware components:

- Low-Pass Filtering: Passive RC filters (10kΩ / 0.1uF) on each ADC input to remove high-frequency noise and "jitter" caused by hand tremors.
- Microcontroller [P/N: ESP32-S3-WROOM-1]: Processes the 4-channel analog data, applies a calibration Look-Up Table (LUT) to linearize the sensor data, and determines the chord.
- ESP-NOW Wireless: Low-latency (<5ms) protocol to sync with the Strumming hand.

2.2.1c Chord recognition mechanism

To implement chord recognition using four flex sensors, we treat each finger as a binary input: 1 (Bent/Fretted) or 0 (Straight/Open). Below is a logic table for 5 essential guitar chords tailored for the 4-finger glove (Index, Middle, Ring, Pinky). We aim to design the chord patterns to resemble the real hand shapes when playing the guitar.

Table 1. Chord Pattern Profile

Chord	Index	Middle	Ring	Pinky	Binary State	Hex State
F Major	1	1	1	1	1111	0x0F
C Major	1	1	1	0	1110	0x0E
D Major	0	1	1	1	0111	0x07
E Minor	1	1	0	0	1100	0x0C
G Major	1	0	0	1	1001	0x09

2.2.2 Subsystem 2: Right Hand

2.2.2a Motion Detection

To detect the rapid "up and down" motion characteristic of a guitar strum, we utilized the BNO055 9-axis absolute orientation sensor. Unlike standard accelerometers which require computationally expensive raw data filtering on the main processor, the BNO055 featured an on-board Cortex-M0+ processor that performed sensor fusion internally. The IMU communicated with the main ESP32 unit via the I2C protocol. The system monitored the linear acceleration vectors, specifically isolating spikes in the total magnitude that exceeded our defined $6.0 \frac{m}{s^2}$ threshold to trigger a note, while utilizing the polarity of the X-axis vector to determine strum directionality (Up vs. Down). The strum velocity magnitude (V_{strum}) was derived from the acceleration vector during the strum event using the following equation:

$$V_{Strum} = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

2.2.2b Strum Trigger Mechanism (Digital Tactile Switch)

To prevent accidental notes from triggering when the user was simply moving their hand without intent to play, our original design proposed a Force Sensitive Resistor (FSR 402) mounted on the thumb. However, integration testing revealed that the analog membrane was too fragile for the aggressive sheer stress of strumming. Consequently, the FSR was replaced with a mechanical digital tactile switch. This sensor acted as a reliable "dead man's switch," mimicking the physical sensation of holding a guitar pick. The switch was configured using the ESP32's internal pull-up resistor, outputting a deterministic logical LOW when pressed. The system registered a valid strum only when the IMU acceleration exceeded the $6.0 \frac{m}{s^2}$ motion threshold simultaneously with the digital switch being actively engaged.

2.2.2c Volume Control (Potentiometer)

To provide consistent dynamic control, a linear potentiometer was used as a manual volume knob. It was configured as a voltage divider connected to an ESP32 ADC pin. During testing, we identified electrical noise at the physical extremes of the potentiometer's rotation. To correct this, the raw 12-bit digital reading was first passed through a software constraint function to bound the values between 100 and 4000. Values below 100 were clamped to filter noise floor jitter, and values above 4000 were clamped to prevent hardware overshoot. This constrained reading ($ADC_{constrained}$) was then mapped to the standard MIDI velocity range (0–127) using the following linear scaling equation:

$$Velocity_{MIDI} = \frac{ADC_{constrained} - 100}{4000 - 100} \times 127$$

2.2.3 Subsystem 3: Power System

2.2.3a Battery Charging

To make a reliable battery system for the glove, there is a need for a handful of things, with the first and foremost being that of a charger. For that, we choose the TP4056 due to it being designed for 18650 Li-ion batteries, having a configurable charging preset, as well as for providing pins that inform the user of the device's charge. Also, the charge is set to .5 amps for the safety of the user.

2.2.3b Load Sharing and Battery Protection

Since the device will have a usb port to connect it to an external power source, we should assure that it be safe as possible, and the best way is to further provide battery protection through load sharing and isolation. The former is done simply with a diode and some PMOS logic to prevent the battery from acting as the main source when the device is plugged in. As for the isolation, with the use of a DWO1A and some NMOS logic, we are able to take advantage of the pins offered by the former to prevent over charge and discharges from happening.

2.2.3a Voltage Conversion

As stated, it is crucial that we convert any voltage we are using to power the gloves into a stable 3.3 volts. Moreover, we would like it to be as small as possible, and that is feasible with the use of an LDO designed for such voltage. Utilizing other power converters, like a buck converter would just take more space due to the need for a driver, as well as for the inductor. Furthermore, the LDO can provide up to 600 mA of current at 3.3 volts, well enough current than what we need.

3. Design Verification

3.1 Subsystem 1

- Chord Threshold: Resistance of each flex sensor was recorded when the finger was bending 0, 45 and 90 degrees. The thresholds are set to the midpoints between 0 degree and 90 degrees for each finger.

	0 degree F0	0 degree F1	0 degree F2	0 degree F3	45 degree F0	45 degree F1	45 degree F2	45 degree F3	90 degree F0	90 degree F1	90 degree F2	90 degree F3
0	1512	1561	1574	1261	1119	1244	1362	1033	837	845	858	766
1	1513	1561	1575	1260	1121	1242	1363	1029	837	847	858	767
2	1511	1563	1573	1261	1119	1243	1363	1022	837	845	858	772
3	1513	1561	1575	1261	1123	1241	1355	1022	837	847	859	767
4	1512	1562	1575	1259	1118	1243	1354	1023	837	845	859	766
5	1513	1562	1577	1261	1119	1242	1355	1031	837	846	859	766
6	1512	1561	1575	1261	1122	1243	1353	1029	837	846	859	767
7	1513	1561	1575	1262	1119	1243	1355	1023	837	845	859	767
8	1512	1562	1575	1260	1117	1241	1355	1021	838	847	859	767
9	1513	1563	1575	1262	1081	1233	1308	1007	837	847	858	767
10	1513	1561	1575	1260	1081	1230	1308	1006	837	849	859	767
11	1511	1563	1575	1261	1083	1231	1309	1007	837	847	858	767
12	1507	1561	1576	1275	1085	1235	1311	1007	839	847	859	773
13	1507	1560	1575	1273	1085	1234	1311	1007	831	847	859	766
14	1507	1560	1576	1273	1085	1231	1311	1007	845	855	870	763
15	1507	1559	1575	1275	1087	1232	1314	1007	844	853	870	765
16	1507	1560	1575	1274	1086	1233	1312	1007	843	855	868	765
17	1507	1561	1576	1277	1087	1235	1311	1006	843	855	871	766
18	1506	1560	1575	1275	1087	1236	1316	1007	843	855	871	764
19	1508	1561	1575	1276	1087	1235	1315	1006	844	855	871	775
20	1507	1560	1575	1274	1092	1235	1312	1007	843	855	871	775
21	1506	1559	1575	1274	1088	1235	1315	1005	844	856	871	775
22	1507	1560	1575	1275	1092	1235	1315	1005	845	855	873	774
23	1508	1560	1575	1276	1097	1237	1319	1007	844	855	871	775
24	1532	1532	1576	1309	1099	1237	1318	1007	843	857	872	776
25	1532	1532	1576	1308	1209	1287	1371	1016	845	856	873	775
26	1533	1533	1575	1309	1209	1287	1371	1017	861	905	878	786
27	1533	1532	1577	1309	1209	1288	1371	1017	861	908	882	787
28	1533	1531	1576	1309	1209	1286	1370	1017	860	907	880	785
29	1533	1531	1576	1307	1207	1287	1370	1017	862	908	883	787
30	1533	1533	1576	1309	1208	1287	1371	1017	861	907	881	785
31	1532	1532	1577	1307	1207	1287	1371	1015	861	907	879	787
32	1533	1533	1575	1309	1207	1288	1371	1017	861	908	883	787
33	1532	1532	1576	1308	1208	1287	1371	1015	862	909	882	788
34	1540	1532	1577	1309	1208	1286	1371	1017	862	908	883	787
Average	1517.085714	1551.857143	1575.4	1280.542857	1130.285714	1251.314286	1340.8	1014.371429	845.4857143	865.1142857	868.6857143	773.4857143

Figure 3. Resistance data from flex sensors

Threshold(midpoint) 0	Threshold 1	Threshold 2	Threshold 3
1181.29	1208.49	1222.04	1027.01

Figure 4. Thresholds for four fingers

- Repeatability: Each of the four chords were attempted multiple times and sensor resistance and chords detected were recorded. We achieved 100% accuracy in chord recognition.

0 F	1 C	2 D	3 Em	4 G
F0.879 F1: 914 F2: 894 F3: 967 Active Chord: 0	F0.893 F1: 927 F2: 921 F3: 1333 Active Chord: 1	F0.1613 F1: 943 F2: 888 F3: 773 Active Chord: 2	F0.841 F1: 954 F2: 1598 F3: 1307 Active Chord: 3	F0.883 F1: 1547 F2: 1454 F3: 973 Active Chord: 4
F0.884 F1: 916 F2: 894 F3: 968 Active Chord: 0	F0.893 F1: 931 F2: 921 F3: 1332 Active Chord: 1	F0.1613 F1: 943 F2: 891 F3: 773 Active Chord: 2	F0.839 F1: 953 F2: 1598 F3: 1304 Active Chord: 3	F0.885 F1: 1547 F2: 1453 F3: 973 Active Chord: 4
F0.884 F1: 916 F2: 895 F3: 967 Active Chord: 0	F0.895 F1: 931 F2: 920 F3: 1331 Active Chord: 1	F0.1612 F1: 943 F2: 891 F3: 776 Active Chord: 2	F0.839 F1: 954 F2: 1598 F3: 1304 Active Chord: 3	F0.885 F1: 1548 F2: 1453 F3: 973 Active Chord: 4
F0.885 F1: 917 F2: 900 F3: 967 Active Chord: 0	F0.895 F1: 932 F2: 921 F3: 1333 Active Chord: 1	F0.1611 F1: 943 F2: 891 F3: 776 Active Chord: 2	F0.839 F1: 953 F2: 1597 F3: 1307 Active Chord: 3	F0.887 F1: 1549 F2: 1455 F3: 973 Active Chord: 4
F0.883 F1: 916 F2: 895 F3: 968 Active Chord: 0	F0.895 F1: 933 F2: 921 F3: 1331 Active Chord: 1	F0.1609 F1: 943 F2: 891 F3: 776 Active Chord: 2	F0.840 F1: 952 F2: 1597 F3: 1305 Active Chord: 3	F0.887 F1: 1547 F2: 1453 F3: 975 Active Chord: 4
F0.884 F1: 917 F2: 901 F3: 969 Active Chord: 0	F0.901 F1: 933 F2: 921 F3: 1331 Active Chord: 1	F0.1608 F1: 945 F2: 892 F3: 778 Active Chord: 2	F0.839 F1: 955 F2: 1598 F3: 1305 Active Chord: 3	F0.886 F1: 1547 F2: 1454 F3: 977 Active Chord: 4
F0.885 F1: 917 F2: 901 F3: 968 Active Chord: 0	F0.901 F1: 931 F2: 920 F3: 1327 Active Chord: 1	F0.1607 F1: 943 F2: 892 F3: 779 Active Chord: 2	F0.831 F1: 953 F2: 1597 F3: 1305 Active Chord: 3	F0.888 F1: 1547 F2: 1453 F3: 978 Active Chord: 4
F0.885 F1: 917 F2: 902 F3: 968 Active Chord: 0	F0.901 F1: 933 F2: 920 F3: 1327 Active Chord: 1	F0.1607 F1: 944 F2: 894 F3: 781 Active Chord: 2	F0.837 F1: 955 F2: 1596 F3: 1303 Active Chord: 3	F0.890 F1: 1548 F2: 1455 F3: 987 Active Chord: 4
F0.885 F1: 917 F2: 903 F3: 968 Active Chord: 0	F0.903 F1: 933 F2: 920 F3: 1326 Active Chord: 1	F0.1607 F1: 943 F2: 893 F3: 778 Active Chord: 2	F0.837 F1: 955 F2: 1596 F3: 1304 Active Chord: 3	F0.890 F1: 1547 F2: 1453 F3: 995 Active Chord: 4
F0.886 F1: 918 F2: 903 F3: 968 Active Chord: 0	F0.901 F1: 933 F2: 921 F3: 1327 Active Chord: 1	F0.1607 F1: 943 F2: 892 F3: 781 Active Chord: 2	F0.837 F1: 953 F2: 1594 F3: 1303 Active Chord: 3	F0.891 F1: 1547 F2: 1453 F3: 995 Active Chord: 4
F0.885 F1: 919 F2: 903 F3: 967 Active Chord: 0	F0.895 F1: 932 F2: 919 F3: 1325 Active Chord: 1	F0.1605 F1: 943 F2: 893 F3: 782 Active Chord: 2	F0.837 F1: 955 F2: 1595 F3: 1302 Active Chord: 3	F0.892 F1: 1547 F2: 1453 F3: 996 Active Chord: 4
F0.887 F1: 919 F2: 903 F3: 969 Active Chord: 0	F0.894 F1: 934 F2: 920 F3: 1325 Active Chord: 1	F0.1605 F1: 944 F2: 894 F3: 782 Active Chord: 2	F0.838 F1: 953 F2: 1595 F3: 1302 Active Chord: 3	F0.892 F1: 1549 F2: 1452 F3: 995 Active Chord: 4
F0.885 F1: 919 F2: 903 F3: 969 Active Chord: 0	F0.893 F1: 933 F2: 920 F3: 1325 Active Chord: 1	F0.1603 F1: 944 F2: 894 F3: 786 Active Chord: 2	F0.839 F1: 954 F2: 1593 F3: 1304 Active Chord: 3	F0.892 F1: 1548 F2: 1452 F3: 1000 Active Chord: 4
F0.885 F1: 919 F2: 903 F3: 968 Active Chord: 0	F0.895 F1: 933 F2: 921 F3: 1325 Active Chord: 1	F0.1599 F1: 943 F2: 893 F3: 787 Active Chord: 2	F0.841 F1: 955 F2: 1593 F3: 1303 Active Chord: 3	F0.893 F1: 1547 F2: 1447 F3: 999 Active Chord: 4
F0.885 F1: 920 F2: 903 F3: 968 Active Chord: 0	F0.895 F1: 933 F2: 921 F3: 1325 Active Chord: 1	F0.1599 F1: 943 F2: 893 F3: 787 Active Chord: 2	F0.841 F1: 953 F2: 1595 F3: 1303 Active Chord: 3	F0.893 F1: 1547 F2: 1447 F3: 999 Active Chord: 4
F0.888 F1: 919 F2: 904 F3: 969 Active Chord: 0	F0.891 F1: 935 F2: 921 F3: 1323 Active Chord: 1	F0.1599 F1: 943 F2: 895 F3: 787 Active Chord: 2	F0.841 F1: 955 F2: 1593 F3: 1303 Active Chord: 3	F0.893 F1: 1547 F2: 1446 F3: 999 Active Chord: 4
F0.889 F1: 919 F2: 903 F3: 969 Active Chord: 0	F0.893 F1: 935 F2: 922 F3: 1323 Active Chord: 1	F0.1599 F1: 943 F2: 894 F3: 788 Active Chord: 2	F0.843 F1: 955 F2: 1592 F3: 1303 Active Chord: 3	F0.894 F1: 1547 F2: 1445 F3: 1001 Active Chord: 4
F0.888 F1: 920 F2: 905 F3: 969 Active Chord: 0	F0.892 F1: 935 F2: 922 F3: 1323 Active Chord: 1	F0.1599 F1: 944 F2: 895 F3: 790 Active Chord: 2	F0.843 F1: 955 F2: 1593 F3: 1304 Active Chord: 3	F0.893 F1: 1545 F2: 1447 F3: 1000 Active Chord: 4
F0.889 F1: 920 F2: 904 F3: 969 Active Chord: 0	F0.893 F1: 935 F2: 923 F3: 1323 Active Chord: 1	F0.1587 F1: 922 F2: 883 F3: 879 Active Chord: 2	F0.843 F1: 955 F2: 1593 F3: 1303 Active Chord: 3	F0.893 F1: 1543 F2: 1447 F3: 999 Active Chord: 4
F0.888 F1: 921 F2: 905 F3: 969 Active Chord: 0	F0.893 F1: 935 F2: 921 F3: 1323 Active Chord: 1	F0.1587 F1: 923 F2: 883 F3: 879 Active Chord: 2	F0.845 F1: 956 F2: 1593 F3: 1304 Active Chord: 3	F0.894 F1: 1543 F2: 1447 F3: 999 Active Chord: 4
F0.889 F1: 920 F2: 905 F3: 969 Active Chord: 0	F0.893 F1: 936 F2: 923 F3: 1324 Active Chord: 1	F0.1589 F1: 924 F2: 883 F3: 879 Active Chord: 2	F0.845 F1: 955 F2: 1593 F3: 1303 Active Chord: 3	F0.891 F1: 1541 F2: 1446 F3: 997 Active Chord: 4
F0.868 F1: 895 F2: 883 F3: 908 Active Chord: 0	F0.815 F1: 840 F2: 988 F3: 1303 Active Chord: 1	F0.1589 F1: 924 F2: 884 F3: 880 Active Chord: 2	F0.895 F1: 940 F2: 1525 F3: 1267 Active Chord: 3	F0.853 F1: 1517 F2: 1429 F3: 967 Active Chord: 4
F0.867 F1: 901 F2: 883 F3: 909 Active Chord: 0	F0.822 F1: 847 F2: 1001 F3: 1305 Active Chord: 1	F0.1588 F1: 924 F2: 884 F3: 879 Active Chord: 2	F0.894 F1: 941 F2: 1525 F3: 1267 Active Chord: 3	F0.853 F1: 1517 F2: 1429 F3: 967 Active Chord: 4
F0.869 F1: 903 F2: 884 F3: 913 Active Chord: 0	F0.827 F1: 856 F2: 1013 F3: 1305 Active Chord: 1	F0.1588 F1: 923 F2: 884 F3: 882 Active Chord: 2	F0.895 F1: 941 F2: 1525 F3: 1269 Active Chord: 3	F0.853 F1: 1517 F2: 1429 F3: 967 Active Chord: 4
F0.869 F1: 901 F2: 884 F3: 911 Active Chord: 0	F0.838 F1: 863 F2: 1019 F3: 1304 Active Chord: 1	F0.1588 F1: 923 F2: 883 F3: 883 Active Chord: 2	F0.893 F1: 941 F2: 1523 F3: 1267 Active Chord: 3	F0.854 F1: 1517 F2: 1429 F3: 968 Active Chord: 4
F0.868 F1: 902 F2: 885 F3: 911 Active Chord: 0	F0.843 F1: 870 F2: 1032 F3: 1304 Active Chord: 1	F0.1587 F1: 925 F2: 876 F3: 875 Active Chord: 2	F0.894 F1: 943 F2: 1523 F3: 1269 Active Chord: 3	F0.855 F1: 1517 F2: 1429 F3: 968 Active Chord: 4
F0.869 F1: 903 F2: 887 F3: 914 Active Chord: 0	F0.853 F1: 873 F2: 1039 F3: 1303 Active Chord: 1	F0.1589 F1: 929 F2: 876 F3: 877 Active Chord: 2	F0.895 F1: 943 F2: 1515 F3: 1269 Active Chord: 3	F0.853 F1: 1517 F2: 1429 F3: 969 Active Chord: 4
F0.871 F1: 903 F2: 887 F3: 915 Active Chord: 0	F0.855 F1: 877 F2: 1043 F3: 1305 Active Chord: 1	F0.1587 F1: 926 F2: 877 F3: 876 Active Chord: 2	F0.895 F1: 943 F2: 1497 F3: 1268 Active Chord: 3	F0.853 F1: 1517 F2: 1428 F3: 969 Active Chord: 4
F0.870 F1: 903 F2: 887 F3: 915 Active Chord: 0	F0.857 F1: 883 F2: 1045 F3: 1305 Active Chord: 1	F0.1589 F1: 927 F2: 879 F3: 879 Active Chord: 2	F0.895 F1: 942 F2: 1483 F3: 1268 Active Chord: 3	F0.853 F1: 1518 F2: 1429 F3: 969 Active Chord: 4
F0.871 F1: 903 F2: 888 F3: 917 Active Chord: 0	F0.859 F1: 885 F2: 1049 F3: 1307 Active Chord: 1	F0.1587 F1: 928 F2: 879 F3: 877 Active Chord: 2	F0.895 F1: 943 F2: 1479 F3: 1263 Active Chord: 3	F0.853 F1: 1518 F2: 1427 F3: 968 Active Chord: 4
F0.873 F1: 903 F2: 889 F3: 916 Active Chord: 0	F0.861 F1: 887 F2: 1051 F3: 1308 Active Chord: 1	F0.1588 F1: 927 F2: 880 F3: 877 Active Chord: 2	F0.901 F1: 944 F2: 1479 F3: 1269 Active Chord: 3	F0.854 F1: 1517 F2: 1429 F3: 970 Active Chord: 4
F0.870 F1: 903 F2: 888 F3: 917 Active Chord: 0	F0.861 F1: 890 F2: 1051 F3: 1308 Active Chord: 1	F0.1588 F1: 927 F2: 883 F3: 879 Active Chord: 2	F0.901 F1: 946 F2: 1481 F3: 1269 Active Chord: 3	F0.855 F1: 1517 F2: 1428 F3: 970 Active Chord: 4
F0.872 F1: 903 F2: 889 F3: 918 Active Chord: 0	F0.861 F1: 892 F2: 1052 F3: 1308 Active Chord: 1	F0.1587 F1: 927 F2: 886 F3: 883 Active Chord: 2	F0.896 F1: 947 F2: 1481 F3: 1271 Active Chord: 3	F0.855 F1: 1517 F2: 1429 F3: 969 Active Chord: 4
F0.872 F1: 906 F2: 889 F3: 917 Active Chord: 0	F0.866 F1: 893 F2: 1053 F3: 1307 Active Chord: 1	F0.1589 F1: 931 F2: 887 F3: 883 Active Chord: 2	F0.896 F1: 943 F2: 1481 F3: 1270 Active Chord: 3	F0.855 F1: 1517 F2: 1427 F3: 968 Active Chord: 4
F0.871 F1: 905 F2: 889 F3: 917 Active Chord: 0	F0.866 F1: 894 F2: 1055 F3: 1305 Active Chord: 1	F0.1589 F1: 931 F2: 888 F3: 880 Active Chord: 2	F0.901 F1: 943 F2: 1481 F3: 1270 Active Chord: 3	F0.853 F1: 1517 F2: 1429 F3: 969 Active Chord: 4
F0.872 F1: 905 F2: 890 F3: 917 Active Chord: 0	F0.866 F1: 895 F2: 1055 F3: 1305 Active Chord: 1	F0.1589 F1: 927 F2: 887 F3: 885 Active Chord: 2	F0.895 F1: 943 F2: 1485 F3: 1270 Active Chord: 3	F0.855 F1: 1517 F2: 1428 F3: 969 Active Chord: 4
F0.875 F1: 906 F2: 889 F3: 917 Active Chord: 0	F0.869 F1: 901 F2: 1055 F3: 1307 Active Chord: 1	F0.1485 F1: 951 F2: 921 F3: 959 Active Chord: 2	F0.901 F1: 946 F2: 1485 F3: 1269 Active Chord: 3	F0.855 F1: 1517 F2: 1428 F3: 969 Active Chord: 4
F0.873 F1: 905 F2: 890 F3: 919 Active Chord: 0	F0.868 F1: 901 F2: 1055 F3: 1305 Active Chord: 1	F0.1485 F1: 951 F2: 923 F3: 959 Active Chord: 2	F0.901 F1: 945 F2: 1485 F3: 1270 Active Chord: 3	F0.855 F1: 1517 F2: 1427 F3: 968 Active Chord: 4
F0.875 F1: 907 F2: 890 F3: 919 Active Chord: 0	F0.869 F1: 902 F2: 1055 F3: 1307 Active Chord: 1	F0.1483 F1: 952 F2: 921 F3: 964 Active Chord: 2	F0.901 F1: 947 F2: 1485 F3: 1271 Active Chord: 3	F0.855 F1: 1515 F2: 1427 F3: 968 Active Chord: 4
F0.875 F1: 907 F2: 890 F3: 915 Active Chord: 0	F0.869 F1: 903 F2: 1055 F3: 1305 Active Chord: 1	F0.1482 F1: 953 F2: 923 F3: 959 Active Chord: 2	F0.895 F1: 944 F2: 1485 F3: 1270 Active Chord: 3	F0.857 F1: 1516 F2: 1427 F3: 969 Active Chord: 4
F0.872 F1: 907 F2: 890 F3: 915 Active Chord: 0	F0.876 F1: 909 F2: 1061 F3: 1305 Active Chord: 1	F0.1481 F1: 953 F2: 923 F3: 959 Active Chord: 2	F0.901 F1: 943 F2: 1487 F3: 1269 Active Chord: 3	F0.856 F1: 1515 F2: 1426 F3: 967 Active Chord: 4
F0.875 F1: 907 F2: 891 F3: 919 Active Chord: 0	F0.875 F1: 910 F2: 1061 F3: 1303 Active Chord: 1	F0.1481 F1: 953 F2: 923 F3: 957 Active Chord: 2	F0.896 F1: 944 F2: 1485 F3: 1271 Active Chord: 3	F0.855 F1: 1517 F2: 1426 F3: 966 Active Chord: 4

Figure 5. Repeatability test results

- Latency: Timestamps were taken when fingers were bent and when BLE data was sent. The time difference was calculated and the average was calculated to be approximately 0.20ms which is well below our requirement of 10ms.

3.2 Subsystem 2

3.2.1 Motion Detection

To verify the motion detection and trigger logic, a custom serial logger was implemented on the ESP32 microcontroller. The system was configured to monitor the calculated IMU acceleration magnitude against the logical state of the digital tactile switch. The subsystem was subjected to physical test cases simulating both intentional strumming and unintentional arm movements. As illustrated in Figure. Motion Detection Tests, the integrated logic successfully filtered out false positives across the evaluated edge cases:

- Motion Too Soft: When the switch was actively engaged but the acceleration remained negligible (e.g., $0.04 \frac{m}{s^2}$ at timestamp 2370), the system correctly ignored the input.
- Pick Dropped: When the device experienced aggressive acceleration spikes (e.g., $8.96 \frac{m}{s^2}$ at timestamp 8624) without the switch being engaged, the system successfully blocked the event.
- Valid Strum: The system accurately generated a "VALID STRUM" event exclusively when both the magnitude threshold (e.g., $6.21 \frac{m}{s^2}$) and the active switch criteria were met simultaneously, as seen at timestamp 16692.

These empirical results confirm that the sensor fusion and trigger logic behaved strictly as specified. The complete Requirement and Verification parameters for this test are detailed in Appendix A.

```

01:53:41.810 -> =====
01:53:41.810 -> MOTION DETECTION VERIFICATION LOGGER
01:53:41.810 -> =====
01:53:41.810 -> TIME (ms)          SWITCH          MAGNITUDE          SYSTEM DECISION
01:53:41.810 -> -----
01:53:42.734 -> 2370              [HELD]          0.04              IGNORED (Motion too soft)
01:53:42.856 -> 2506              [OPEN]          0.06              -
01:53:48.999 -> 8624              [OPEN]          8.96              IGNORED (Pick Dropped)
01:53:48.999 -> 8635              [OPEN]          10.42             IGNORED (Pick Dropped)
01:53:48.999 -> 8646              [OPEN]          7.75              IGNORED (Pick Dropped)
01:53:54.896 -> 14524             [HELD]          0.08              IGNORED (Motion too soft)
01:53:57.061 -> 16692             [HELD]          6.21              VALID STRUM FIRED
01:53:57.205 -> 16850             [HELD]          6.59              VALID STRUM FIRED
01:54:03.896 -> 23537             [OPEN]          0.15              -
01:54:03.933 -> 23582             [HELD]          0.84              IGNORED (Motion too soft)
01:54:03.971 -> 23627             [OPEN]          0.61              -

```

Figure 6. Motion Detection Tests

3.2.2 Volume Linearity

To verify the dynamic range and software mapping of the volume control, the physical potentiometer was swept across its full rotational range. A serial logger was utilized to simultaneously record the raw 12-bit ADC input, the software-constrained value, and the final calculated MIDI velocity output. As illustrated in Figure. Volume Linearity Tests, the empirical data confirmed that the linear scaling algorithm and constraint logic functioned precisely as specified:

- Minimum Sweep (~0%): A raw ADC reading of 152 was correctly constrained and mapped to a nearly silent MIDI velocity of 1.
- Midpoint Sweep (~51%): A raw reading of 2068 was accurately mapped to the center of the MIDI range, yielding a velocity of 64.
- Maximum Sweep (~100%): At the physical limit, the raw reading peaked at 4012. The system successfully executed the hardware overshoot filtering logic, clamping the value to the defined upper boundary of 4000 and outputting a stable maximum MIDI velocity of 127.

These results verify that the system successfully mitigates electrical noise at the physical extremes while providing stable, linear volume control across the entire sweep. The complete Requirement and Verification parameters for this evaluation are located in Appendix A.

```

03:24:27.231 -> VOLUME LINEARITY VERIFICATION LOGGER
03:24:27.231 -> =====
03:24:27.231 -> POSITION          RAW ADC          CONSTRAINED      MIDI VELOCITY
03:24:27.231 -> -----
03:24:43.436 -> ~0%             152             152             1
03:24:58.023 -> ~17%            776            776            22
03:25:03.834 -> ~51%            2068           2068           64
03:25:07.091 -> ~82%            3271           3271           103
03:25:09.264 -> ~100%           4012           4000           127

```

Figure 7. Volume Linearity Tests

3.2.3 Polling Latency

To ensure the Right Hand subsystem did not introduce a hardware bottleneck, the local sensor polling intervals and ESP-NOW transmission times were evaluated utilizing hardware-level microsecond timers. The system was configured to transmit a continuous burst of packets to the Left Hand receiver, logging the MAC-layer acknowledgments to accurately calculate the one-way latency. As illustrated in Figure. Latency Tests, the empirical data confirms the system's high-speed throughput. Across a robust sample size of 187 evaluated packets, the system achieved an average one-way transmission latency of exactly 1.103 ms. This result unequivocally demonstrates that the design goal was met. The local data acquisition and transmission processes consume less than 4% of the overarching 30 ms "Motion-to-Sound" latency budget. This performance validates that the subsystem easily satisfies real-time operational constraints while preserving ample processing overhead for the downstream Bluetooth and MIDI pipelines. The complete Requirement and Verification parameters for this evaluation are located in Appendix A.

```

00:47:23.980 -> =====
00:47:23.980 -> LATENCY TEST SUMMARY
00:47:23.980 -> =====
00:47:23.980 -> Packets Evaluated: 187
00:47:23.980 -> Average Latency: 1.103 ms
00:47:23.980 -> =====

```

Figure 8. Latency Tests

3.3 Subsystem 3

3.3.1 Battery Protection

Testing as to whether operation with a battery will be safe is a bit finicky, especially in the design stage, especially due to how volatile batteries can be, especially lithium ion based ones. Therefore,

the best way to assure proper battery charging behaviour is to utilize a power supply, and test the implemented isolation and load sharing. If it all works correctly, it should look something like this. Current is flowing into the power supply, meaning that is being charged when the device is connected via the USB port.

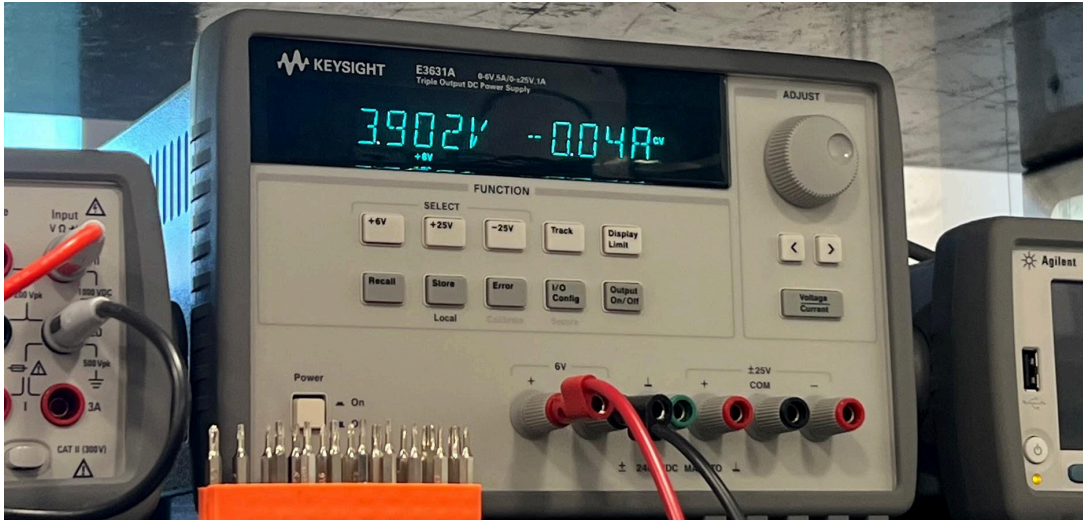


Figure 9. Power Supply (state charging)

3.3.2 Power supply Operation

As for testing if the power system operates correctly when connected to a load, the best way to do so is to simply use a very cheap but informative load, and an LED fits that description. If our power system is operating correctly then it shouldn't falter in any of the various cases where it should deliver power to the load. Such is the following figure that demonstrates three different operating states that the power system should be able to handle power delivery to the load. (the yellow LED being on means that the load is receiving power, and 3.3 volts at it, as it is after the power conversion).

Battery Operation Plugged and Charging Plugged Only

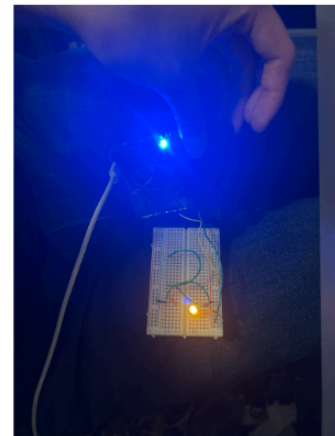
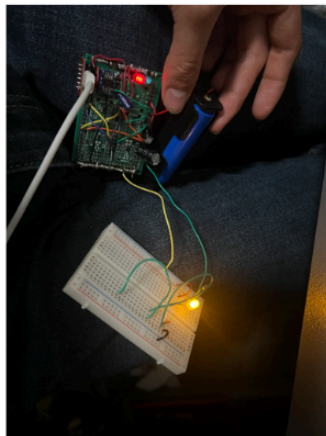
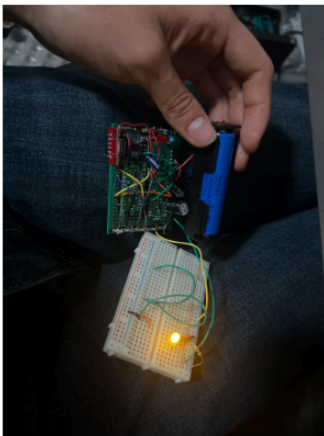


Figure 10. Power System under operation of battery and external power.

Aside from the shining yellow LED, there are two other LEDs that shine under two operations, and those are the red and blue LEDs. These are LEDs utilized to inform the user if the battery is charging, or fully charged/absent, and they're intuitively matched to red and blue, respectively.

4. Costs

4.1 Parts

Table 2. Parts Costs

Part	Number	Manufacturer	Retail Cost (\$)	Bulk Purchase Cost (\$)	Actual Cost (\$)
ESP32-S3-WRO OM	2	Espressif Systems	5.49	3.93	0
22kΩ Resistor	10	DigiKey	0.10	0.08	0
10kΩ Resistor	10	DigiKey	0.10	0.08	0
0.1μF Capacitor	10	Digikey	0.08	0.13	0
USB-C Port	2	Keebio	1.98	N/A	1.98
DW01A	10	DigiKey	0.10	0.23	0.23
FS8205A	10	DigiKey	0.34	2.07	2.07
TP4056	10	Amazon	0.80	7.99	7.99
AP2112K-3.3 TRG1	3	Digikey	0.22	1.53	0
Li-ion 18650, 3.7, 1800mAh	2	Amazon	7.48	N/A	14.96
18650 Battery Holder	5	Amazon	6.99	N/A	6.99
5.1kΩ Resistor	10	DigiKey	0.10	0.08	0
1KΩ Resistor	10	DigiKey	0.10	0.08	0
1.2kΩ Resistor	10	DigiKey	0.10	0.08	0
150Ω Resistor	10	DigiKey	0.10	0.08	0
10μF Capacitor	10	DigiKey	0.10	0.08	0
1μF Capacitor	10	DigiKey	0.10	0.08	0
Amphenol FCI Clincher Connector (2 Position, Male)	4	Sparkfun	11.8	N/A	11.8
Flex Sensor 2.2"	4	Sparkfun	50	N/A	50
LOCCEF Work Gloves	1	Amazon	3.65	N/A	3.65
BNO055	1	DigiKey	12.34	N/A	12.34
FSR402	1	InterLink	2.99	N/A	2.99

10K Thumbwheel Potentiometer	1	Adafruit	0.95	N/A	0.95
Total					115.95

4.2 Labor

For this project, we found that it would be justifiable if our salaries were \$40 per hour, as most pcb designers tend to around such salaries, when starting. Assuming that our work starts from the 4th week of this semester till the last week of the semester, that would make this project 9 weeks long. Moreover, we would be spending about 15 hours on this entire project per week. Therefore, our final expense in regards to salary would be \$16,200, with each of us making about \$5,400 dollars.

5. Conclusion

All requirements of the project were satisfied. We were able to play 5 chords in both up and down strumming directions, control the speed and volume of the output sound, and power both of the gloves completely wirelessly with rechargeable batteries.

5.1 Ethical Considerations

As engineers, our design process is guided by the IEEE Code of Ethics, which has evolved over the last century to help professionals worldwide uphold the highest level of ethical conduct.

Most critically, the code mandates that engineers must "protect the safety, health, and welfare of the public". We will avoid ethical breaches related to user safety by treating the wearable nature of this device with the utmost caution. We will not conduct human testing with the gloves until the power subsystem and Battery Management System (BMS) have been rigorously tested to operate within safe thermal limits.

Additionally, specifically within the right-hand glove subsystem, the integration of the button acts not only as a functional strum trigger but as a critical safety "dead man's switch." This ensures the device only transmits data when actively engaged by the user's thumb, minimizing unintended outputs. We must also consider the potential for accidental or intentional misuse. Because the device promotes active physical movement and untethers the user, a musician could become distracted in inappropriate environments. We have an ethical responsibility to provide clear operational warnings regarding situational awareness in our user manual.

5.2 Future Work

To elevate the project's performance and design, we will integrate machine learning to enable significantly more precise movement recognition. Additionally, we will utilize TPU for high-quality 3D printing, ensuring the housing is both durable and flexible. Finally, we will develop custom PCBs to consolidate the electronics, resulting in a sleeker, more compact device.

6. References

- [1] “Espressif Documentation,” *Espressif.com*, 2025.
https://documentation.espressif.com/esp32_technical_reference_manual_en.pdf
- [2] R. Aalund and M. Pecht, “The Use of UL 1642 Impact Testing for Li-ion Pouch Cells,” *IEEE Access*, vol. 7, pp. 176706–176711, 2019, doi: <https://doi.org/10.1109/access.2019.2957814>.
- [3] E. W. Pugh, “Creating the IEEE Code of Ethics,” *IEEE Xplore*, Aug. 01, 2009.
https://ieeexplore.ieee.org/abstract/document/5337855?casa_token=bcaOY10Nki4AAAAA:jADx2UDuB2w1DbO-RSxYNSrzdrefoXr5hueKFwerItgb8ZOocDGf_N_aJ5xUP62DoJUa_GW2NW2m
- [4] Wikipedia Contributors, “Power Glove,” *Wikipedia*, Feb. 08, 2026.
- [5] Bosch Sensortec, *BNO055 Intelligent Absolute Orientation Sensor, 9-Axis Sensor Fusion Data Sheet*, Rev. 1.8, BST-BNO055-DS000-18, Oct. 2021. [Online]. Available:
<https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bno055-ds000.pdf>
- [6] *TP4056 Datasheet*, Top Power ASIC, Corp. China.
- [7] *DW01A Datasheet*, SLKOR, China.
- [8] *AP2112 Datasheet*, DIODES, Inc., Plano, Texas, 2017.
- [9] *FS8205A Datasheet*, EVVOSEMI, Hong Kong.
- [10] *ElectronicsHacks*, “How to Use the TP4056?,” *ElectronicsHacks*. Available:
<https://electronics hacks.com/how-to-use-the-tp4056/>.
- [11] Designing a Battery Protection Circuit With DW01A,” *Instructables*. Available:
<https://www.instructables.com/Designing-a-Battery-Protection-Circuit-With-DW01A/>.
- [12] A. Banerjee, “SEPIC Converter,” class notes for ECE 464, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, 2025.

Appendix A Requirement and Verification Table

Table 3. Left Hand Verification

Requirement	Verification Method
Chord Threshold	The system must distinguish between a "half-bend" (45°) and a "full-bend" (90°).
Repeatability	Perform the same "C-Major" bend 50 times. The identified chord must be correct in $\geq 98\%$ of trials.
Latency	Measure the time from a finger bend exceeding the threshold to the ESP-NOW packet being sent. Must be $< 10\text{ms}$.

Table 4. Right Hand Verification

Requirement	Verification Method
Motion Detection	<p>A valid strum is registered only when the IMU linear acceleration magnitude crosses the defined motion threshold ($> 6.0 \text{ m/s}^2$) while the tactile switch is actively engaged, functioning as a "pick holding" dead man's switch.</p> <p>Methods:</p> <ol style="list-style-type: none"> 1. Program the ESP32 to print the calculated IMU acceleration magnitude and the digital state of the tactile switch (Pressed/Released) to the Serial Monitor. 2. Press and hold the switch without moving the hand. Verify that no strum event is triggered on the Serial Monitor. 3. Move the hand in a rapid vertical strumming motion without pressing the switch. Verify that no strum event is triggered on the Serial Monitor.

	<p>4. Press and hold the switch to simulate a pick grip and execute a rapid vertical strum.</p> <p>5. Observe the Serial Monitor to ensure the calculated acceleration magnitude crosses the coded acceleration threshold simultaneously with the switch being in the "Pressed" state, resulting in a successful strum trigger output.</p>
<p>Volume Linearity</p>	<p>The linear potentiometer must accurately map its full physical rotation (represented by a 12-bit ADC reading) to the standard MIDI velocity range (0127) using a linear scaling algorithm.</p> <p>Method:</p> <ol style="list-style-type: none"> 1. Program the ESP32 to continuously sample the potentiometer ADC value, apply boundary constraints (100–4000), and calculate the mapped MIDI velocity. 2. Rotate the physical volume knob to five approximate target positions: 0% (Minimum), 25%, 50%, 75%, and 100% (Maximum). 3. Observe the Serial Monitor at each resting position to verify that the raw ADC values scale linearly across the full range and accurately map to the correct MIDI velocity bounds (0 to 127) without overflowing or underflowing.
<p>Polling Latency</p>	<p>The time required to poll the BNO055 via I2C, read the digital switch, and complete the wireless transmission must not create a bottleneck, keeping the total one-way hardware latency under 30ms.</p> <p>Method:</p> <ol style="list-style-type: none"> 1. Program the Right Hand ESP32 to capture a start timestamp using the hardware-level micros() timer immediately before executing the esp_now_send command. 2. Utilize the ESP-NOW OnDataSent callback interrupt to capture an end timestamp the exact microsecond the MAC-layer Acknowledgement (ACK) arrives from the Left Hand receiver.

	<p>3. Program the ESP32 to calculate the Round Trip Time (RTT), divide it by 2 to determine the one-way latency, and keep a running aggregate of the latency across multiple packets.</p> <p>4. Press and hold the tactile switch to initiate continuous packet transmission, perform a strum action, and then release the switch.</p> <p>5. Observe the Serial Monitor output upon button release to verify that the calculated average one-way latency across all transmitted packets remains strictly under 30.0 ms.</p>
--	---

Table 5. Power System Verification

Requirement	Verification Method
Battery Voltage (charge)	User intuitive LEDs that inform if the device (battery) is charged or or standby, red and green respectively. Should also be capable of up to 2 hours of charge. Voltage of battery signifies its charge.
Battery Protection	To prevent overcharge and discharge battery protection in forms of a DWO1A and load sharing will be implemented to isolate the battery and prevent sudden shorts when charging and utilizing the device. Thus the device should operate normally even if the device is plugged to another power source. Easily testable with an LED.
Voltage Output	The power system should convert the 3.7 volts of the battery to a stable 3.3 volts, required to power the ESP32. Signified by the other modules working as intended.
Current Output	The power system should be able to deliver up to 200 milli Amps of current, as the bluetooth operation will require up 150 milli Amps, and any other GPIO interaction will at worst take up 50 milli Amps.