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 electric 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" Controller that maps finger curvature to digital chord profiles and a Right-Hand "Strum" Controller 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, and a Force Sensitive Resistor (FSR) 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 speakers.

Contents

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" Controller that maps finger curvature to digital chord profiles, and a Right-Hand "Strum" Controller that captures rhythmic timing and velocity. By fusing these data streams via the low-latency ESP-NOW protocol to a laptop and outputting standard Bluetooth Low Energy (BLE) MIDI to an external speaker, the device allows users to perform music anytime and anywhere, effectively untethering the musician from the physical constraints of the hardware.

1.3 Visual Aid

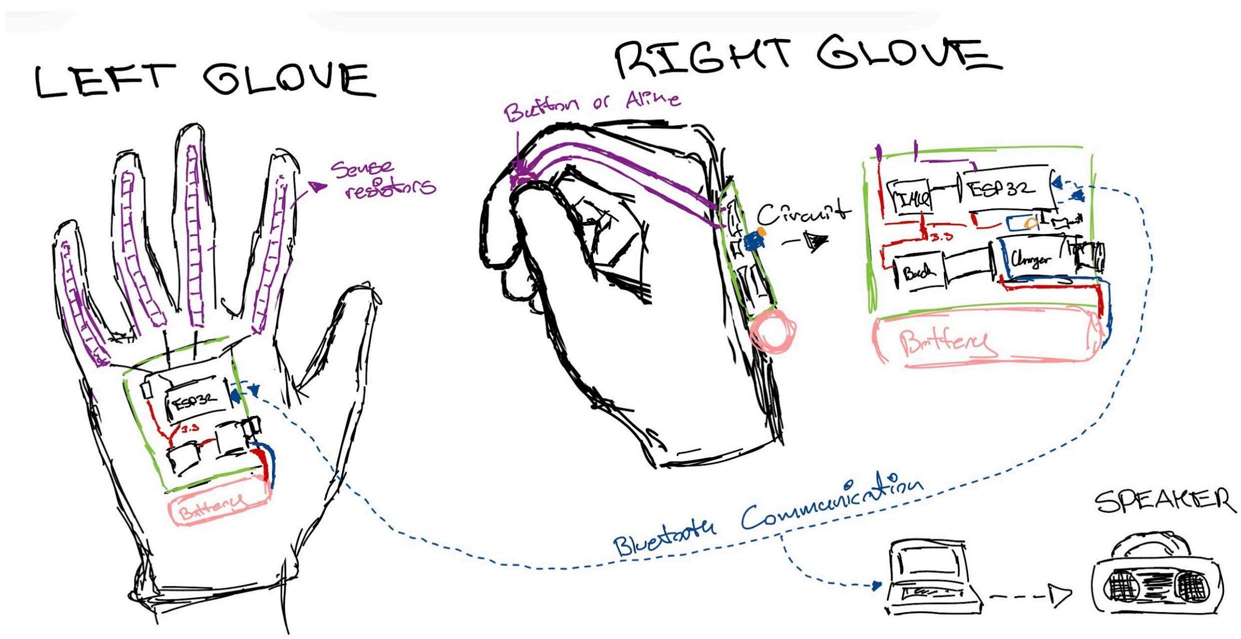

Figure 1. Air Guitar Gloves Illustration

1.4 High-level requirements list

- Latency: The total "Motion-to-Sound" delay must be under 30ms. Anything higher is noticeable to a musician.

- ○ Test Method: We will program a "Test Mode" where a physical button press on the Strum hand toggles a GPIO pin (HIGH) and simultaneously sends the wireless strum packet. Using an oscilloscope, we will measure the delta (t) between the GPIO HIGH signal and the arrival of the MIDI Note On message at the receiver's serial port.

- 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 will send 1,000 packets at the target rate (e.g., 100Hz). The Left-Hand unit will log the sequence numbers; a successful test results in $\leq$ 10 missed packets.

2. Design

2.1 Block Diagram

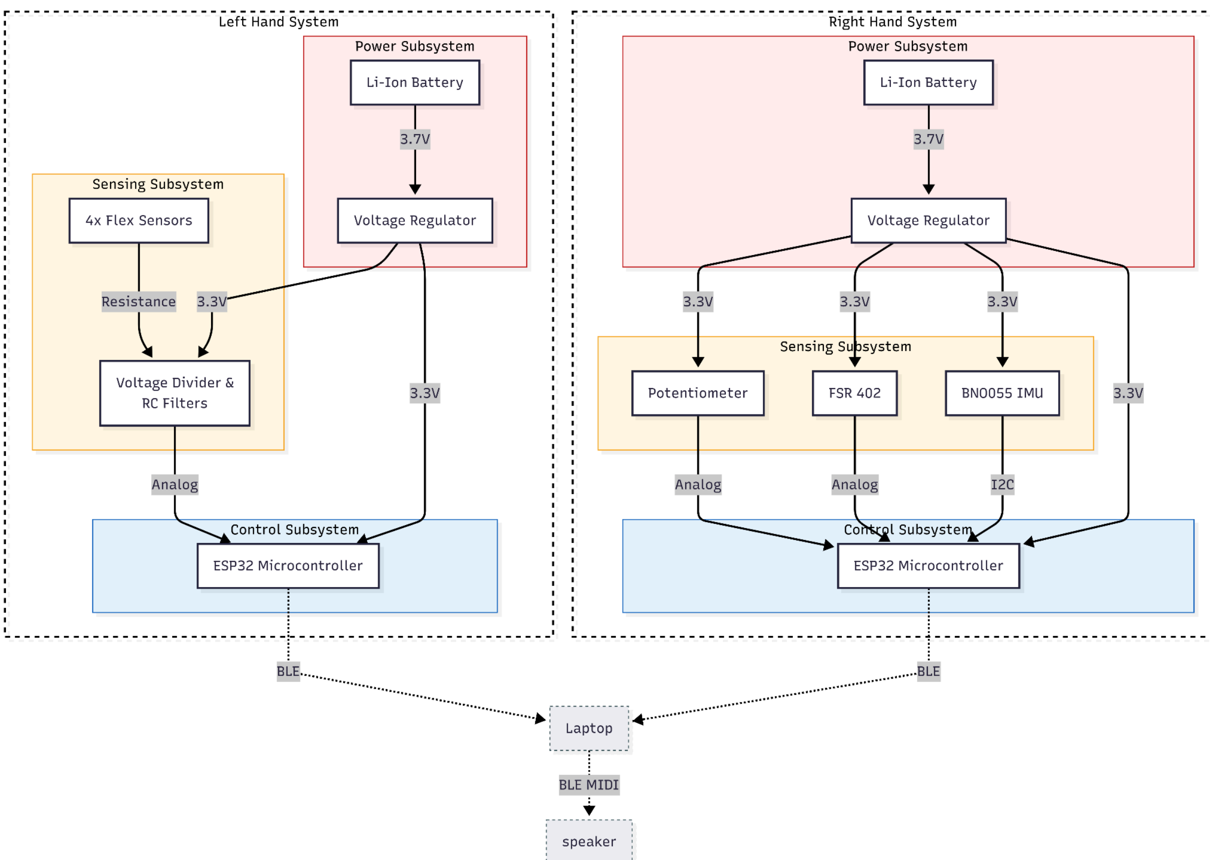

Figure 2. Air Guitar Block Diagram

2.2 Subsystem Overview

2.2.1 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).

2.2.2 Subsystem 2: Right Hand

This subsystem functions as the rhythmic trigger for the Air Guitar, determining when a note is played and controlling its intensity. It integrates three primary sensors to capture user intent: a BNO055 9-axis IMU to detect the precise timing of strumming motions, a Force Sensitive Resistor (FSR) on the thumb that acts as a "dead man's switch" to prevent accidental triggers when the user is not simulating a pick grip, and a potentiometer to provide manual, adjustable volume control. The subsystem is driven by a dedicated ESP32 microcontroller, which processes these sensor inputs in real-time and transmits the valid strum events to the Left-Hand Controller using the low-latency ESP-NOW wireless protocol.

2.2.3 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/controllers. Therefore, each of the gloves/controllers 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 controller, as well as any other components utilized in attaining the MIDI signal for a specified chord of the user choosing.

2.3 Subsystem Requirements

2.3.1 Subsystem 1: Left Hand

2.3.1a Resistive Sensing Array

This layer captures the raw physical deformation of the fingers.
Hardware components:

- Flex Sensors (4x) [P/N: FS-L-0054-103-ST]: 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

$$V_{cc} = 3.3V$$
$$R_{flex,\,flat} \approx 10k\Omega$$
$$R_{flex,\,bent} \approx 30k\Omega$$

2.3.1b 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-WROOM-32E]: 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.

2.3.1c 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

2.3.2 Subsystem 2: Right Hand

2.3.2a Motion Detection

To detect the rapid "up and down" motion characteristic of a guitar strum, we will utilize the BNO055 9-axis absolute orientation sensor. Unlike standard accelerometers which require computationally expensive raw data filtering on the main processor, the BNO055 features an on-board Cortex-M0+ processor that performs sensor fusion internally. The IMU communicates with the main ESP32 unit via the I^2C protocol. The design monitors the linear acceleration vectors, specifically looking for spikes in the vertical axis (relative to the hand's orientation) that exceed a defined threshold. The strum velocity magnitude (v_{strum}) is derived from the acceleration vector during the strum event:

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

2.3.2b Strum Trigger Mechanism (FSR)

To prevent accidental notes from triggering when the user is simply moving their hand without intent to play, a Force Sensitive Resistor (FSR 402) will be mounted on the thumb. This sensor acts as a "dead man's switch," mimicking the physical sensation of holding a guitar pick. The FSR is implemented in a voltage divider configuration with a fixed $10k\Omega$ resistor (R_{fixed}). The output voltage (V_{out}) read by the ESP32's ADC is given by:

$$V_{out} = V_{cc} \times \frac{R_{fixed}}{R_{FSR} + R_{fixed}}$$

The system registers a valid strum only when the IMU acceleration exceeds the motion threshold and the FSR voltage exceeds the "pick holding" threshold.

2.3.2c Volume Control (Potentiometer)

To provide consistent dynamic control, a linear potentiometer is used as a manual volume knob. It is configured as a voltage divider connected to an ESP32 ADC pin. The 12-bit digital reading (0–4095) is mapped to the standard MIDI velocity range (0–127) using the following equation:

$$Velocity_{MIDI} = \frac{ADC_{reading}}{4095} \times 127$$

2.3.3 Subsystem 3: Power System

The power system must ensure that the gloves/controllers are powered for at least 2 hours, and that they pose no threat to the wearer when utilized and placed under immense stress. As stated, each of the controllers will hold their own lithium battery with a battery management system to accompany it. Doing so will allow for the controllers to be wireless, which is a big advantage in comfortability and usage of the air guitar. Nevertheless, the power systems implemented must meet the following requirements to deliver satisfactory performance.

Requirements:

1. Voltage Regulation: Each battery must be regulated to maintain a stable 3.3 voltage with a .1 ripple. Going any higher than this might lead to damages to the microcontrollers, which might result in deterioration of the controllers, or even complete malfunction altogether.
2. Current Regulation: Each battery, and administering system, must support at the very least 150 milli-Amps of continuous current, as the bluetooth features of the ESP32 will demand such.
3. Battery Capacity: Due to the decently high current requirements for many of the ESP32s features, each lithium battery must at the very least support up to two hours of continuous operation, which should be possible with the utilization of lithium ion batteries for each controller.
4. Safety protection: Each battery management system implemented should account for possible overcharge, over-discharge, overcurrent, and overheating, due to the relative closeness of such potent batteries to the user. Neglect of such battery operations could lead to possible safety hazards and bodily harm.

2.4 Tolerance Analysis

2.4.1 Subsystem 1: Left Hand

Our design places the “Flat Finger” voltage at $V_{out} = 2.27V$ and “Bent Finger” at $V_{out} = 1.39V$. To Analyze the Threshold Margin, define the “Chord Trigger” threshold at the midpoint:

$$V_{thresh} = \frac{2.27+1.39}{2} = 1.83V$$

The Error Budget:

- Resistor Tolerance (1%): ±0.02V
- ADC Quantization Error: ±0.0008V (Negligible)
- ESP32 ADC Non-linearity (approx. 5%): ±0.09V
- LDO Voltage Drift (1%): ±0.03V

Total Worst-Case Error (E_{total}):

$$E_{total} = 0.02 + 0.0008 + 0.09 + 0.03 = 0.1408V$$

Feasibility Demonstration:

Our "Safety Margin" (S_m) is the distance from our operating points to the threshold:

$$S_m = \left|V_{flat} - V_{thresh}\right| = |2.27 - 1.83| = 0.44V$$

Since our Safety Margin (0.44V) is significantly larger than our Worst-Case Error (0.14V), the design is mathematically feasible. Even with component drift and ADC non-linearity, the system will not "misfire" a chord.

2.4.2 Subsystem 2: Right Hand

The primary functional requirement for the Subsystem is to detect and transmit a strum event with a total motion-to-sound latency of less than 30ms. The most significant bottleneck within this subsystem is the time required to poll the sensors: reading data from the BNO055 IMU via the I^2C bus and converting analog signals from the FSR and Potentiometer via the ADC. If the cumulative sensor polling time t_{poll} is significant, it leaves insufficient time for wireless transmission, causing the system to fail the latency requirement.

To demonstrate feasibility, we calculate the worst-case time required to acquire one complete "frame" of data from all three sensors.

2.4.2a IMU Polling Time t_{IMU}

The BNO055 communicates via the I^2C protocol. We configured the bus to "Fast Mode" (400 KHz).

- Data Size: The system reads the Linear Acceleration Vector (x, y, z). Each axis is a 16-bit integer (2 bytes), plus register addressing overhead.
- Total Transaction ≈ 6 bytes (data) + 2 bytes (overhead) = 8 bytes
- Bus utilization: I^2C includes start/stop/ack bits, effectively doubling the bit count to ≈ 128 bits.

$$t_{IMU} = \frac{Total\ Bits}{Bus\ Speed} = \frac{128\ bits}{400,000\ Hz} = 0.00032s = 320\ \mu s$$

2.4.2b ADC Conversion Time t_{ADC}

The ESP32 must sample two analog channels: the FSR (Trigger) and the Potentiometer (Volume).

- According to the ESP32 technical reference, the Successive Approximation Register (SAR) ADC requires approximately 15 clock cycles per sample.
- With a default ADC clock of 2 MHz (conservative setting), the sampling time per channel is:

$$t_{samples} = \frac{15\ cycles}{2{,}000{,}000\ Hz} = 7.5\ \mu s$$

Since we read two channels (FSR + Potentiometer):

$$t_{ADC} = 2 \times 7.5\,\mu s = 15\,\mu s$$

2.4.2c Total Subsystem Latency t_{total}

We sum the polling times plus a conservative 50% safety margin for microcontroller processing overhead (CPU instruction cycles).

$$t_{poll} = t_{IMU} + t_{ADC} = 320\mu s + 15\,\mu s = 335\,\mu s$$

$$t_{total} = t_{poll} \times 1.5 = 0.502\,ms$$

The worst-case sensor acquisition time for Subsystem 2 is approximately 0.5 ms. This is orders of magnitude lower than the 30ms system budget (utilizing only ≈ 1.6% of the available time window).

2.4.3 Subsystem 3: Power System

To have a better grasp and understanding of requirement 3, the estimated current consumption of each module and component will be listed out.

Estimated consumption per component.

- ESP32: approximately 150 to 500 mA
- MPU-6050 (IMU): approximately 3 to 4 mA
- Left hand voltage divider: 1 to 5 mA
- Potentiometer: 1 to 5 mA
- Miscellaneous components (extra RLC components as needed): 10mA

So we could expect what is up to a maximum of 514mA, if the ESP32s are placed under massive amounts of stress. Nevertheless, what should be expected to be around the minimum current consumption is about 165 to 200 mill-amps for each controller.

Therefore, by utilizing a 3000mAh 18650 battery, we are able to meet the requirement of a minimum operation time of 2 hours, well reaching what could be 15 hours, at most, or 6 to 5 at the very least, meeting the minimum requirement.

3. Design Verification

3.1 Subsystem 1: Left Hand

Table 2. Left Hand Verification

Requirement	Verification Method
Chord Resolution	The system must distinguish between a "half-bend" (45°) and a "full-bend" (90°) with a minimum ADC delta of 400 LSBs.
Repeatability	Perform the same "C-Major" bend 50 times. The identified chord must be correct in ≥98% of trials.
Latency	Measure the time from a finger bend exceeding the threshold to the ESP-NOW packet being sent. Must be <10ms.
Linearity	Verify that the software Look-Up Table (LUT) corrects the flex sensor's inherent non-linearity, resulting in a calculated angle within ±5° of a physical protractor measurement.

3.2 Subsystem 2: Right Hand

Table 3. Right Hand Verification

Requirement	Verification Method
Motion Detection	The system must register a valid strum only when the IMU linear acceleration exceeds a motion threshold and the FSR voltage exceeds a "pick holding" threshold. Methods:

	1. Connect the DMM probes to the FSR analog output pin and the common ground to monitor output voltage. 2. Program the ESP32 to print the I2C data from the BNO055 and the FSR ADC readings to the Serial Monitor. 3. Apply varying levels of pressure to the FSR without moving the hand. Verify that no strum is triggered. 4. Move the hand in a rapid vertical strumming motion without pressing the FSR. Verify that no strum is triggered. 5. Hold the FSR tightly to simulate a pick grip and execute a rapid vertical strum. 6. Observe the Serial Monitor to ensure the calculated acceleration magnitude crosses the coded acceleration threshold simultaneously with the FSR voltage threshold, resulting in a successful trigger state.
Volume Linearity	The linear potentiometer must provide an adjustable analog voltage that accurately maps the 12-bit ADC reading (0–4095) to the standard MIDI velocity range (0–127). Method: 1. Connect the DMM probes to the wiper pin of the potentiometer and the common ground. 2. Rotate the manual volume knob to five distinct physical positions: 0%, 25%, 50%, 75%, and 100% rotation. 3. At each position, record the steady analog voltage from the DMM. 4. Simultaneously, read the 12-bit ADC value and the mapped MIDI velocity output from the ESP32 Serial Monitor to ensure it correctly follows the mapping algorithm.
Polling Latency	The time required to poll the BNO055 via I2C and the analog channels via the ADC must not create a bottleneck, keeping the total "Motion-to-Sound" delay under 30ms.

	Method: 1. Program a specific Test Mode on the ESP32 where a physical test button (or the FSR trigger crossing its threshold) forces a GPIO pin to toggle HIGH. 2. Connect Channel 1 of the oscilloscope to this GPIO pin to capture the exact moment of physical input initiation. 3. Connect Channel 2 of the oscilloscope to a second GPIO pin programmed to toggle HIGH the exact microsecond the ESP32 finishes its sensor polling and initiates the ESP-NOW wireless transmission. 4. Perform a strum action to trigger the sequence. 5. Freeze the waveforms on the oscilloscope and use the time cursors to measure the delta (t) between the rising edge on Channel 1 and the rising edge on Channel 2.

3.3 Subsystem 3: Power System

Table 4. Power System Verification

Requirement	**Verification Method**
Battery Voltage	The TP4056 will have two pins that will be utilized to let the user know when the battery is charging and when it is completely charged. Moreover, the module should be able to charge the battery up to 4.2 volts, the upper limit and max charge of the 18650 single cell lithium ion batteries. Therefore, utilizing a volt-meter for the voltage of the battery will be a good way to verify this function and behaviour.
Battery Protection	The designed battery protection should prevent the battery from overcharge and discharge. For this to be tested, the pins dedicated

	for such protections on the DW01A could be tested to see which NMOS gates they enable,and if they do so correspondingly.
Voltage Output	The battery, once connected to the LDO, should supply 3.3 volts to the ESP32, and other components, with a fluctuation of 100 millivolts, and no more. This can be taken with a simple oscilloscope reading of the output of the LDO, as well put under constant use, of up to about 2 hours.
Current Output	The LDO should allow for up to 500 milli-Amps of current, at maximum. As for general use, it should easily provide from 150 to 200 milli-Amps without much strain. We could apply some heavier commands to see if it can handle the upper limits of current, but repeated testing with both left and right hand and some measuring with an oscilloscope could show its reliability.

4. Costs and schedule

4.1 Parts

Table 5. Parts Costs

Part	Number	Manufacturer	Retail Cost ($)	Bulk Purchase Cost ($)	Actual Cost ($)
ESP32-S3-WROOM	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 endeavorous 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. Now, 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.

4.3 Schedule

Table 6. Schedule

Week	Task	Miaomiao Jin	Youngmin	Arturo Arroyo Valencia
3.2	Have all parts ordered, prepare for design review, finish layouting PCBs, order PCBs	PCB design for left hand	PCB design for right hand	Order all components
3.9	Prepare for Breadboard demo, make sure the ground logic works, debug and modify PCB for third order if necessary	Put together necessary circuits for left hand on the breadboard and test all the sensors and logic	Put together necessary circuits for right hand on the breadboard and test all the sensors and logic	Breadboarding and testing the power system circuits
3.16	Spring break			
3.23	Program the MCU for left and right hand to get accurate data	Program left hand logic	Program right hand logic	Program wireless communication between devices
3.30	Test and verify. Debug if needed. Assemble everything onto the gloves.			
4.6	Progress demo			
4.13	Test and verify. Debug if needed			
4.20	Prepare for Mock Demo			
4.27	Prepare for Final presentation			
5.4	Prepare for Final paper			

5. Ethics, safety and societal impact

5.1 Safety Concerns and Regulatory Standards

The most prominent safety concern for the Air Guitar system involves the power delivery mechanism. Because each controller is untethered and wearable, the system relies on lithium-ion batteries located in close proximity to the user's body. If mishandled or poorly regulated, these batteries pose a risk of over-discharge, overcurrent, and overheating, which could lead to thermal runaway and bodily harm.

To mitigate these risks, our project will adhere to the UL 1642 Standard for Safety Lithium Batteries, which provides rigorous requirements for both primary and secondary lithium cells. Compliance with UL 1642 ensures the batteries can withstand environmental, electrical, and mechanical stress—including standardized impact and crush tests—to prevent unsafe reactions like fire or explosion when a cell is obstructed or physically damaged.

Furthermore, because our design relies heavily on wireless communication—specifically the ESP-NOW protocol and Bluetooth Low Energy (BLE)—we will ensure our ESP32 microcontrollers comply with FCC Part 15 regulations. Part 15 governs the operation of low-power, unlicensed radio frequency devices, mandating that such devices must not cause harmful interference to authorized radio services and must accept any interference received. During the development phase, our team will strictly adhere to UIUC campus policies and ECE laboratory safety guidelines, particularly the protocols surrounding soldering, handling exposed circuitry, and testing volatile power sources.

5.2 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 controller subsystem, the integration of the Force Sensitive Resistor (FSR) 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.

5.3 Societal, Economic, and Environmental Impact

The Air Guitar project carries several broader impacts across multiple contexts:

- Societal and Economic: Traditional musical instruments are inherently bulky and often prohibitively expensive. Our solution democratizes music creation by significantly lowering the financial and physical barriers to entry. By translating kinetic motion into digital MIDI data, individuals who may lack the space or budget for a full electric guitar setup can practice and perform anywhere.
- Environmental: The proliferation of IoT devices and wearable electronics contributes to the growing global crisis of electronic waste (e-waste). The microcontrollers, flex sensors, and especially the lithium-ion batteries used in our gloves possess a limited lifespan and require responsible disposal. To mitigate our environmental footprint, we will design the system using durable, modular components to extend the product's life cycle and prioritize rechargeable power systems over disposable alternatives.

6. References

- "Espressif Documentation," *Espressif.com*, 2025. https://documentation.espressif.com/esp32_technical_reference_manual_en.pdf
- 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.
- 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
- Wikipedia Contributors, "Power Glove," *Wikipedia*, Feb. 08, 2026.