Mock Design Review: Dual Glove Air Bass
Ying Chen, Pranathi Gummadi, Niranjan Jayanth
ECE 445
TA: Michael Genovese
1. Introduction

1.1 Objective
The dual glove air bass is a portable electronic device that replicates the physical characteristics of the instrument without the physical medium. For this solution, the inspiration of was drawn from the concept of “air guitar”, a performance art in which an individual pretends to play an imaginary instrument with accuracy. In this specific case, we will be implementing a wearable device capable of generating audio output and replicating the stylistic techniques for playing a bass guitar.

1.2 High-level requirements list
a. The air bass must be able to sustain through 6 hours of battery-operated playing.
b. The air bass must have at least 90% accuracy of pluck attempts as played notes.
c. The air bass must have at least 90% pitch accuracy on left-hand finger placement.

1.3 High-Level Validation Plan
a. Upon completion of initial build, leave the MCU on for 6 hours with intermittent use to see battery life.
b. Upon completion of each successive build, individually test right hand accuracy by playing a pre-decided sequence of plucks. Compare the 10 datasets with the correct output (preset).
c. Upon completion of each successive build, individually test left hand accuracy by a pre-decided sequence of notes. Compare the 10 datasets with the correct output (preset).

2. Design

2.1 Block Diagram

![Block Diagram](image)

Figure 1: Block Diagram
2.2 Physical Design

![Figure 2: Right Glove Sensor Placement](image1)

![Figure 3: Left Glove Sensor Placement](image2)

2.3 Block Design

2.3.1 Flex Sensors

The flex sensors will be used to measure the bend angles in the fingers, indicating the press state of the left hand and the pluck state of the right hand. The sensors are optical, using 2 circuits: an LED circuit and a corresponding photoresistor circuit. They will be powered by the 5V rail getting power from a 9V battery. A voltage divider will be used to measure the value of the photoresistor, which then will be used to train which values are which angles. For the left hand, the first string will see a middle knuckle bend of 90°, the second string will see 120°, the third 150°, and the lowest string 180°. The subsequent voltages will not be hard coded, but used to train the system to recognize different gestures. Further, there will be a second flex sensor on the base knuckle of each finger of the left hand, to tell the difference between a finger on the lowest string and a finger pointing upwards. For the machine learning to be most accurate, we will need the flex sensors to see at least a 0.1 V difference in the target angles.

*Requirement 1: The flex sensor must produce voltages distinguished by ±0.1 V when bent at 90°, 120°, 150° and 180°.*

*Requirement 2: The flex sensor voltages at a given angle must not differ more than 50 mV.*

*Requirement 3: The flex sensor voltages must approach their correct value range for each angle within 12 ms.*

**Verification:** Observe voltage states for various LEDs and various tubing methods at key angles. Use MCU to supply voltage to a test circuit to measure the voltage. Python will be used to extract data into a .txt for characteristic graphing in MATLAB.
Figure 4: Schematic of flex sensor test circuit

Left Circuit: LED

\[ R = IV \]
\[ R_{\text{total}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} \]
\[ R_{\text{total}} = \frac{1}{3\Omega} + \frac{1}{3\Omega} + \frac{1}{3\Omega} = 1k\Omega \]

Right Circuit: Photoresistor

\[ I_{\text{right}} = \frac{V - V_p}{R} = \frac{5V - V_p}{100k\Omega} \]
\[ R_p = V_p \cdot I_{\text{right}} \]
\[ R_p = V_p \cdot \frac{V - V_p}{R} \]
2.3.2 Accelerometer

The accelerometer will be used to determine relative position and determine pluck state. It will be powered by the 3.3V rail getting power from the 9V battery. The accelerometer will send analog data to the ADC of the MCU to be integrated to find relative position. The MCU will then interpret this and use these inputs as inputs to a state machine to find the next state. The accelerometer and gyroscope will
also be used to sense the plucking of the string, based on the rotational acceleration and linear acceleration of the finger, detecting whether the user had the intent of plucking. This will handle the case of a bent finger that is stationary, and thus not plucking the string.

**Requirement 1:** The accelerometer must have a minimum range ±2g and the gyroscope must have a minimum range of ±180 dps.
**Requirement 2:** The module must not exceed dimensions of 50mm x 50mm.
**Requirement 3:** The accelerometer must provide real-time data accurate with 12 ms.

**Verification:** Use MCU to supply 3.3 V to accelerometer module. Read accelerometer values for plucking motion and plot using MATLAB. Read position values in real-time to see sample-delay and thus time-delay in obtaining relative position.

### 2.3.4 9V Battery

The battery will be used to power the system. Since our project is designed to be mobile, we cannot use a stationary power supply; further, since the components being powered are relatively low-power, we can use a 9V battery.

### 2.3.5 Power Transformer

The power transformer must be able to supply a 3.3V from the 9V battery. The resulting rails must have a steady voltage with little overshoot for sake of not sending more power than allowed to the flex sensors and accelerometers/gyroscopes.

**Requirement 1:** The step-down power will not exceed ±0.5V below or above intended voltages at 200 mA.

\[
\begin{align*}
3.3 &= \frac{5R_2}{R_1 + R_2} \\
1.7R_2 &= 3.3R_1 \\
R_2 &= 1.941R_1
\end{align*}
\]

![Proposed Circuit](image)
2.3.6 MCU
We are using the MSP432 microcontroller from Texas Instruments. The MSP432 series is known for its low power requirements and high performance. Texas Instruments provides a large library of technical support documents for UART, SPI, IrDA, and I²C interface protocols. The microcontroller also features a 14-bit ADC resolution and 24 channels for analog inputs. Our project falls under the category of wearable, portable, consumer electronics which is a common application for the MSP432. A 14-bit IEEE’s floating point number is capable of representing precise binary values by using 9-bit significant values, 4-bit exponents, and 1-bit for sign. On the MCU there is a total of 256kB of flash memory, 64kB of SRAM, and 32kB of ROM available. There is a 6mA current sink/source for digital inputs. The microcontroller must include an option of an SD card support, needed to store sound samples and calibration data from the PC.

2.3.7 AUX output
The output of our system will be the note played by the user. We will be storing sound samples on an SD card and playing them back out to an auxiliary output. This can be plugged into any given speaker. The audible range for a human being is 20 Hz to 20 kHz.

3. Schedule

<table>
<thead>
<tr>
<th>Date</th>
<th>Task Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>October 2nd</td>
<td>Build initial flex sensors and obtain sensitivity data. Create design review documents.</td>
</tr>
<tr>
<td>October 9th</td>
<td>Have the first build with necessary sensors and get 50% pluck accuracy</td>
</tr>
<tr>
<td>October 16th</td>
<td>Build the left glove with the necessary sensors and initial implementation of string changing</td>
</tr>
<tr>
<td>October 23rd</td>
<td>Obtain necessary sensor data to input into calibration software. Begin software development for recognition algorithms.</td>
</tr>
<tr>
<td>November 6th</td>
<td>Final PCB designed and ordered.</td>
</tr>
<tr>
<td>November 15th</td>
<td>Finish software development. Set up output to speaker. Begin calibration testing with software.</td>
</tr>
<tr>
<td>November 20th</td>
<td>Integrate PCB in final glove set-up.</td>
</tr>
<tr>
<td>December 1st</td>
<td>Viable demo of a song involving notes from E1 to B2. Fine tuning until final demo day.</td>
</tr>
<tr>
<td>December 6th</td>
<td>Demo Day</td>
</tr>
<tr>
<td>December 11th</td>
<td>Final Presentation</td>
</tr>
</tbody>
</table>