S0undFingers

Team - Thomas Driscoll, August Gress, Kyle Patel
ECE 445 Project Proposal 2 - Spring 2020
TA: Dhruv Mathur

1 Introduction

1.1 Objective

Not everyone knows how to play an instrument. And if they do, they might not know how to do it well and be able to stay in key. But they would like to be able to make some music!

Our solution is a bluetooth-enabled glove that has five force sensitive resistors at the end of each of the five finger holes. When you would press the pad of your finger while within the glove against a hard surface, it would play a programmed tone through the speaker on your mobile phone. It would do this by using an internal app that outputs to audio device drivers provided by Android. The tone that would play would be programmable from the app, on a per-finger basis.

1.2 Background

Playing and interacting with music is a pastime that many enjoy, but due to financial, space, or other constraints, can be hard to realize. Coupled with the difficulty of learning a new instrument, there is just a lot of overhead. Many people, however, do make the tradeoffs to get to play and enjoy making music in their homes [1]. This usually leads to it negatively affecting the people around them however, as neighbors or others in the home have to hear the instrument regardless of whether they were interested in hearing it or not [2].

We saw through these issues that there should exist a way for someone to start playing along with their favorite songs, no matter which instrument in the piece they wanted to emulate. We also wanted to make sure that there was a way that the instrument noise could be controlled, so we designed our glove to play its tones using a phone's speaker or anything plugged into the phone's headphone jack.

The design that we propose is different from the original solution in two ways.

First, we do not use any kind of "mode." The original project used different modes to specify how different hand motions (like left to right or finger-bending movements) would affect the notes that would be played, such as in their "piano mode" where moving left to right would simulate playing down and up the piano respectively. We do not use movement to control what notes are played, instead it is based on which finger(s) is(are) currently pressing against a hard surface while within the glove. Further, the note that is played when this happens is completely programmable and not tied to a specific movement or finger.

In the original solution, there was also this concept of note “production”, where the notes produced would always be in the same key so the music produced would be harmonic. We have nothing of the sort, and instead leave it to the discretion of the end user to decide what they would like to have play while they are using the glove.

1.3 High Level Requirements

- Able to recognize finger taps within a pressure-sensitive bluetooth-enabled glove and turn those taps into signals based on which finger is being pressed.
- Able to send those signals from that glove via bluetooth to play a given sound from a mobile phone via an app.
- The latency between a finger tap and sound outputting through the phone is at most Bluetooth protocol latency (200ms) + 50ms for our processing $\pm$100ms (Total: 250ms $\pm$100ms).

2 Design

2.1 Block Diagram

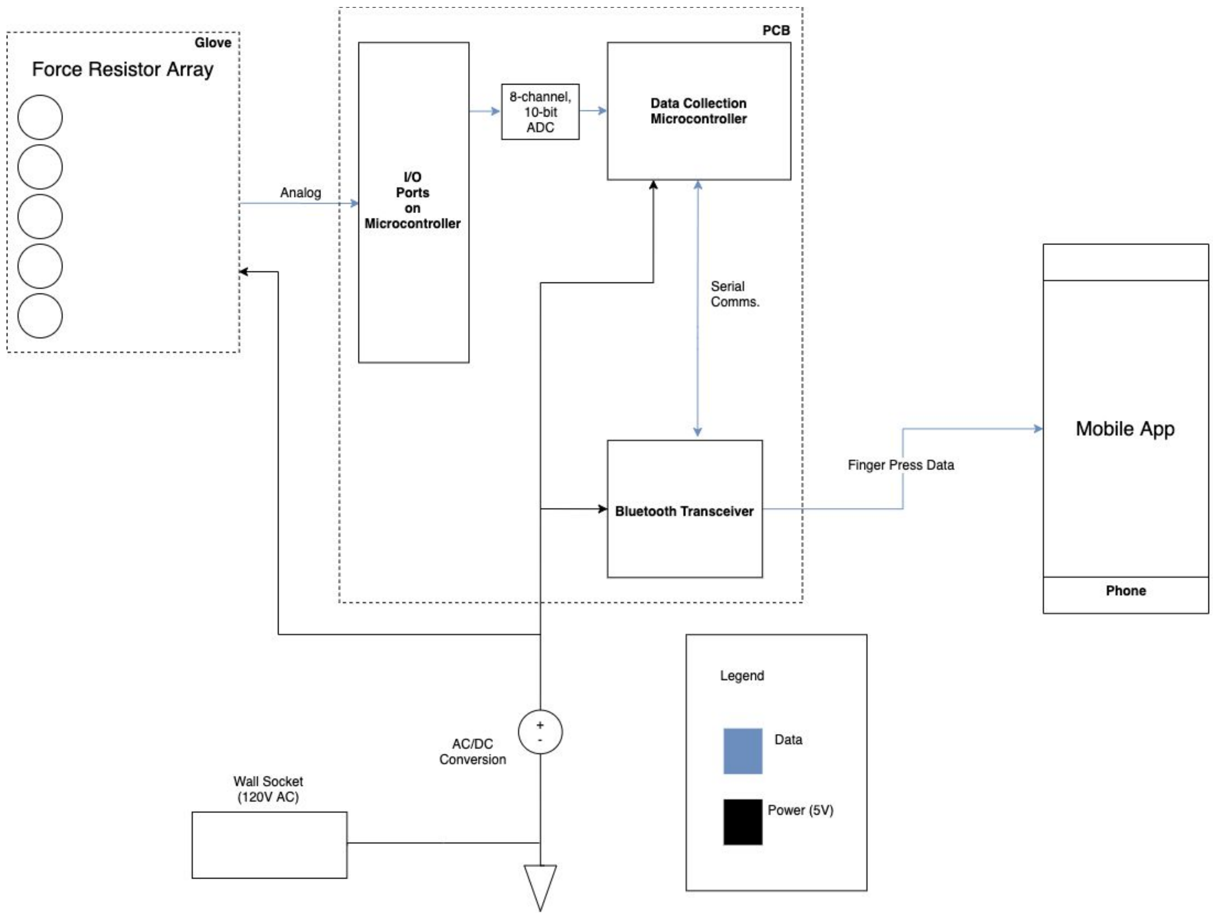

Fig. 1: Block Diagram ranging from electronics to PCB to phone

2.1.1 Description of Block Diagram

Attached to the fingertips of the glove will be 5 FSR 402 resistive force sensors. The resistors will be connected across the analog I/O ports of the data microcontroller. The analog I/O ports connect to an ADC on the data microcontroller (ATMega328P). When a force is applied to the sensor (100g-10kg) [3] the measured resistance across the sensor's terminals decreases. When the measured resistance value for a sensor is less than an experimentally obtained value, then the program running on the ATMega328P considers that the finger attached to that sensor is pressing down on a surface. The voltage source for the sensors will be regulated and provide 5V ± 0.3V for each sensor.

The ATMega328P will send data of which fingers are pressed down to the Bluetooth transceiver module (HC-05) via a serial communication protocol (most likely SPI/I2C). The HC-05 will send this data to the mobile device it is paired with. The HC-05 operates with a voltage of 5V.

From the phone component, our app acts as a server and searches for clients to connect to. If in range, it will detect the Bluetooth transceiver and pair with it. Once it is paired, the app on the phone will create and accept a data stream from the bluetooth transceiver with minimal latency. Upon receipt of input, the data stream will be output to the phone's speakers. The user will be able to customize the sounds associated with each fingertip in a user-facing interface on the phone.

2.2 Functional Overview

In the above diagram, our design consists of four major environments. What follows is a break-down of each of those environments and the modules that work inside them.

2.2.1 Glove

The glove would have 5 pressure sensitive pads that connect directly to the microcontroller, relaying information about whether or not they are currently being pressed. These sensors would also be powered off of the same 5V connection as the HC-05 and microcontroller.

We would also have a soft membrane between the finger and the sensor for comfort, and to transfer the pressure from the user's finger to the pressure sensor.

2.2.2 Microcontroller/PCB

We would use an ATMega328P microcontroller to control the glove, and a HC-05 Bluetooth module to communicate between the glove and the mobile phone attached to it via Bluetooth. We would power the glove using a 5V wall adapter. The code would be written using the Arduino IDE, and the PCB substrate would be FR-4 using 2 layers.

2.2.3 Mobile/Real-Time App

Our app will be completely stored and run on a user's phone, serving three purposes. The first would be to connect via Bluetooth to the gloves themselves and receive digital input from them. The second purpose would be to process in real-time the input received from the gloves and output that to an audio jack. Finally, the app would also have a user-facing interface to allow customizable sounds. The user would select the instrument desired and a corresponding sound from that instrument for each finger.

Due to the need to access the underlying operating system and device drivers, we plan to use the Android MediaPlayer API to guarantee native Android development (our app will not be accessible using Apple systems for our demo). To initialize and interact with the Bluetooth transceiver, the app, on startup, will also utilize the Android native Bluetooth API, which will configure a majority of our settings and allow us to customize the Bluetooth connection for our needs.

2.3 Physical Design

Fig 2. is what we plan for our physical representation of the glove to look like from a top down view. Essentially, we want to have pressure-sensitive resistors in the fingers to capture the finger press input from the user, and a box on the top of the hand that will contain all the necessary wiring and controllers.

We modelled the size of the hand and fingers based on historic averages [4]. We also took care to ensure that the size of our pressure-sensitive resistors would be small enough to fit within those averages.

Fig. 3 is another view from the side, where you can see how high the box on the top of the hand would be. We wanted to strike a balance between having it be large enough that we could fit all the electronics we needed but not so large that it would impede a user’s ability to tap effectively.

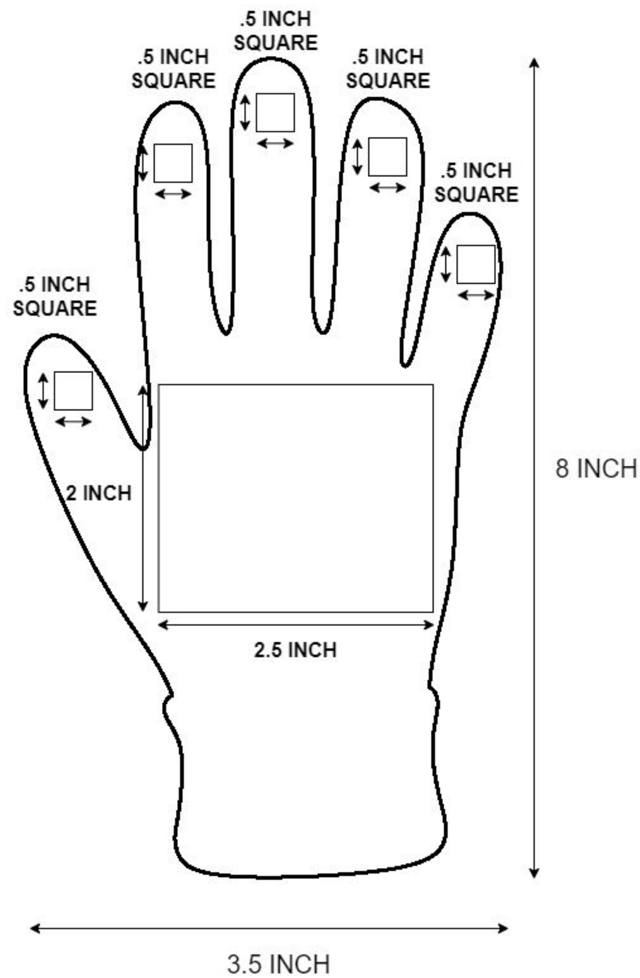

Fig 2. Physical Diagram (Top Down View)

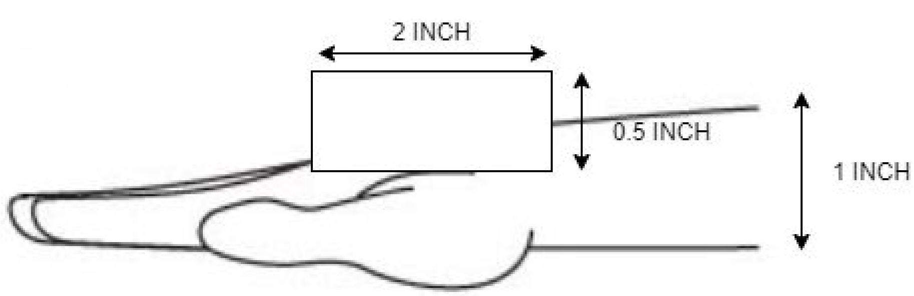

Fig 3. Physical Diagram (Side View)

2.4 Block Requirements

Module	Requirements
Glove: FSR 402 Force Sensitive Resistors	1. Each force sensitive resistor must receive 5 ± 0.3V by way of a linear voltage regulator.
PCB: Microprocessor (ATMega328P)	1. Must be able to facilitate sequential collection of quantized FSR 402 data. 2. Must be able to send digital sensor readings to Bluetooth Microchip via SPI (serial port).
PCB: Bluetooth Microchip (HC-05)	1. Has a discoverable Bluetooth profile. 1. Must be able to maintain connection with a bluetooth-connected phone with 95% ±5% reliability. 2. The microchip must receive 3.6-5V input.
User-Interactive Software: React Native Front-End	1. Users should be able to view supported instrument sounds. 2. Users should be able to pair a particular sound to a particular finger. 3. Front-end should have an icon showing the current status of Bluetooth pairing with gloves. 4. >80% unit test coverage for every React component.
M2M Software: Bluetooth Module[5]	1. Service discovery process successfully caches the security key for bluetooth operation. 2. Process only accepts recognized Bluetooth profiles. 3. When paired, connection lasts until it is outside of range (30m ± 5m). 4. When paired, <1% data loss. 5. Data stream is received server-side and manipulatable.
Sound Processing: Android Media Player[6]	1. Create MediaPlayer object successfully in the Idle state. 2. Read .mp4 files containing sounds successfully. 3. Process data stream from bluetooth module. 4. Output data stream to audio speakers. 5. Successfully release MediaPlayer object when Bluetooth connection ends; i.e. end gracefully.

2.5 Risk Analysis

The biggest risk to the success of this project would have to be the communication between the glove and the mobile app. There are two main aspects within this communication that we must consider during our design and implementation.

The first aspect to consider is latency. We want to make sure that the delay between a person tapping their finger in the glove and the sound being played is as short as possible. This timing delay would ideally be the delay of bluetooth transmission from the glove to the phone (typically 40ms [7] for aptX, or 200ms $\pm 10\%$ [8]) plus at most 50ms $\pm 10\%$ for our processing. Keeping the delay as low as possible is critical for music and rhythm based-applications, so making sure this latency is low is paramount to the project's success.

The other aspect we must consider is the transmission of information between the bluetooth module itself and the mobile app. We need to make sure that our transmission of data is robust enough to handle the transfer of multiple sounds at the same time (in the case that the user is pressing with multiple fingers) and fail gracefully in the case that connection is lost between the app and the glove.

3 Ethics & Safety

The ethical or safety issues with our project pertain to the physical gloves themselves, the microcontroller and Wi-Fi chips.

Citing the IEEE Code of Ethics #1 [9], we will work to ensure that the construction of our gloves is structurally sound such that a user will not be concerned with electrical hazards such as exposed wires or static shock, or any harm from burning ICs or plastic. Further, a likely source of potential harm would be liquids spilling on the glove, ruining the circuity and causing an electrical hazard to form near the user's hands. To prevent this, all circuits in our glove will have a protective layer on the top of them that prevents any spillage into the sensitive electronics underneath.

An additional source of safety concern is the user-facing application, specifically in regards to the ACM Code of Ethics 2.9 [10]. While we expect the user of our prototype to load the application from source code provided by the designers, bad actors could potentially hijack the Bluetooth connection in the app itself to download malware onto a user's phone [11]. These concerns, while valid, are an extremely low risk as our application will not be downloaded outside of the authors knowledge for the duration of the project. Further, we will be whitelisting the gloves such that the app will reject any interaction that is not associated with that Bluetooth identifier (BD_ADDR).

Finally, regarding regulatory standards, since we are creating a receive-only device, we are exempt from type approval [12]. If we were to take the product to market, we would need to test at an accredited testing house, followed by an application for Part 15 certification [12]. However, since we are not, we do not need to address those issues at this time. Our understanding is that this would be a relatively simple process that would require time and money to pay for accreditation, neither of which are available to us.

References

[1] Rock, L. and Rock, L., 2020. *Music Lessons Doncaster, Melbourne | Guitar, Singing & Piano - Invest In Learning Guitar: How Much Does It All Cost?*. [online] Music Lessons Doncaster, Melbourne | Guitar, Singing & Piano. Available at: <https://www.leadersofrock.com/posts/2015/7/27/invest-in-learning-guitar-how-much-does-it-all-cost> [Accessed 3 April 2020].

[2] Kay, L., 2020. *Bombed By Bass: The Neighbours' Music Turned My Flat Into A Hellhole | Laura Kay*. [online] the Guardian. Available at: <https://www.theguardian.com/commentisfree/2014/dec/05/noisy-neighbours-drug-taking-my-flat-hellhole> [Accessed 3 April 2020].

[3] Interlink Electronics, "FSR 400 Series Round Force Sensing Resistor," FSR 402 Data Sheet, Oct. 26, 2010. Available: https://cdn.sparkfun.com/assets/8/a/1/2/0/2010-10-26-DataSheet-FSR402-Layout2.pdf

[4] Healthline. (2020). *Average Hand Size: For Adults, Children, Athletes, And More*. [online] Available at: <https://www.healthline.com/health/average-hand-size#adults> [Accessed 3 April 2020].

[5] Android Developers. (2020). *Bluetooth Overview | Android Developers*. [online] Available at: <https://developer.android.com/guide/topics/connectivity/bluetooth#kotlin> [Accessed 3 April 2020].

[6] Android Developers. (2020). *Mediaplayer | Android Developers*. [online] Available at: <https://developer.android.com/reference/android/media/MediaPlayer> [Accessed 3 April 2020].

[7] Qualcomm® aptX™ Low Latency Synchronised Audio. (n.d.). Available: https://www.aptx.com/aptx-low-latency. [Accessed: 3 April 2020].

[8] Available: https://www.electronicdesign.com/industrial-automation/article/21808300/how-to-pick-the-best-bluetooth-protocol-for-your-application. [Accessed: 3 April 2020].

[9] ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 3 April 2020].

[10] Acm.org. (2020). *The Code affirms an obligation of computing professionals to use their skills for the benefit of society.*. [online] Available at: https://www.acm.org/code-of-ethics [Accessed 3 April 2020].

[11] Chicago Tribune. (2020). *Chicago Tribune - Widespread Android Virus Could Hide in Popular App Updates*. [online] Available at: https://www.chicagotribune.com/news/ct-xpm-2012-04-16-sns-201204161154usnewsusnwr201204130413androidapr16-story.html [Accessed 3 April 2020].

[12] Blueradios.com. (2020). [online] Available at: <https://www.blueradios.com/Bluetooth_Global_Certification_Requirements.pdf> [Accessed 3 April 2020].