S0UNDFINGERS FINAL REPORT

By

August Gress

Kyle Patel

Thomas Driscoll

Final Report for ECE 445, Senior Design, Spring 2020

TA: Dhruv Mathur

8 May 2020

Project No. 8

Abstract

The problem we wanted to solve was the ability for people to make music wherever and whenever they wanted in as natural of a form as possible. We also wanted to provide the user with the choice of who hears the music made, and which songs they wanted to emulate. To solve this, we created a Bluetooth enabled glove with touch sensitive tips that, when tapped, would play a pre-sampled configurable tone through a user's Bluetooth-paired mobile phone.

Contents

1. Introduction

What follows is our implementation of a glove to help users learn to play music and try out new melodies. It is an updated version of a previous ECE 445 project. Our motivation for the updated version of the product was to simplify the original design into something that was easy to pick up and play immediately. We also felt it was important that the user had more freedom for where and when they could tap along to their favorite songs, so we made the glove work with your phone and favorite pair of headphones anywhere you went with a power outlet nearby. We believe our solution fits the needs of our target customer better and, with additional improvements, could be a strong product for market.

1.1 Problem Statement

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. People also might not want to disturb their neighbors or have the space in their house to store instruments, but still want to engage in making music. They also might have physical impairments that hinder them from being able to hold them.

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 must hear the instrument regardless of whether they were interested in hearing it or not [2].

1.2 Solution

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.

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

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.

There is one other competitor product on the market in the form of the MINI.MU Glove Kit, which is essentially a DIY motion driven glove for music production [3]. One of the biggest problems with it however, is that it is hard to accurately track hand movements, as hands can have very fine motor control movements that are difficult for a motion sensor to detect. Our solution improves upon this by being completely force driven, meaning that it's very definite to know when you have made a sound (as you have physically pressed your finger against a sensor).

We envisioned that our solution would be better for the target audience, which would be people interested in *making* music. Our gloves allow the natural use of beating with your fingers to generate music, as well as more mobility. Since our gloves can be plugged into any standard wall socket and connect to the user's phone, they can be played anywhere, allowing for groups to play together simultaneously and easily, unlike the previous solution which required being tethered to a PC.

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 100ms (Total: 250ms 100ms).

1.4 Visual Aid

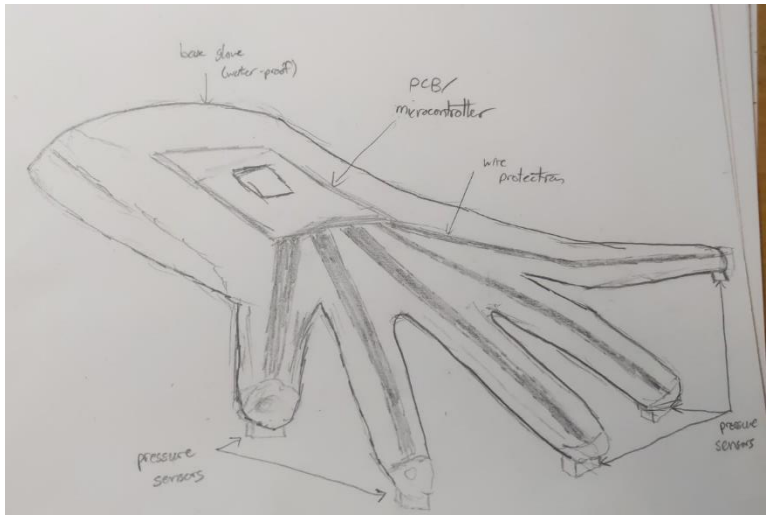


Figure 1: Physical Diagram

Figure 1 is a drawing of what we imagine our prototype to look like. The PCB sits on the back of the glove with a power connection coming off somewhere. Our wired connections are protected with some insulating, waterproof material, as indicated by the black lines on the back of the glove. You can also easily see the force sensors on the tip of each glove tip.

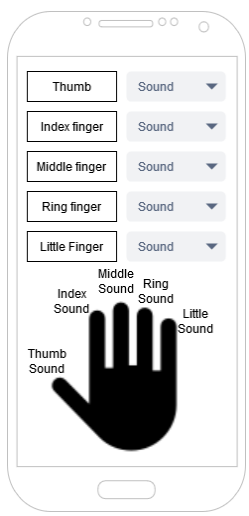


Figure 2: Mobile App Mock-up

Figure 2 is a mock-up of what we imagine the Android app to look like on the user facing side. The user will be able to select a sound for each finger from a dropdown menu and, to visually confirm this, see the sound appear next to each finger on the image below.

1.5 Block Diagram

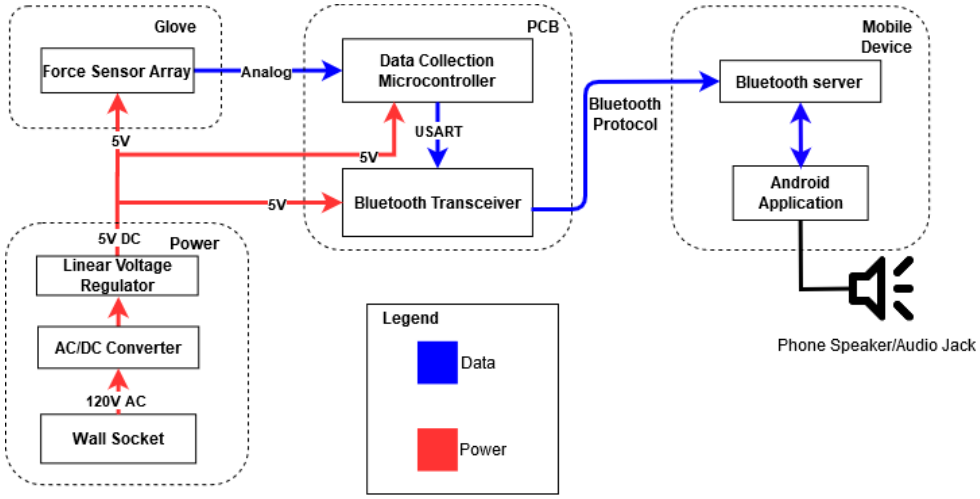


Figure 3: Block Diagram

Our power supply block indicates that voltage will come from a wall socket. We convert from AC to DC for the purposes of our design and put that DC voltage through a linear voltage regulator.

On the glove proper, we simplify our 5 force sensors per glove into a single block that receives power from the power supply and connect via an analog signal to the I/O ports on our PCB.

Our PCB is also simple, connected to the power supply and, via Bluetooth, to our mobile device. The microcontroller processes sensor inputs from the glove, writes the metadata and packet info and sends it to the Bluetooth transceiver.

Our mobile device, most likely a phone, pairs to the Bluetooth transceiver and receives the message containing the sensor input. Our Android application processes this signal and outputs on the Android's speaker.

2. Second Project Implementation

The implementation of our project consists of two major components: the PCB and circuit diagrams. While this is not complete due to the ongoing COVID-19 pandemic, we believed these to be the essential components of the project that we could do remotely.

2.1 PCB

One of the first ways that we wanted to show that our project was feasible was to create the PCB that would be affixed to the backside of the glove (on the flat side of the user's hand). Below are some examples of what the PCB would look like and explanations on each of the subsections.

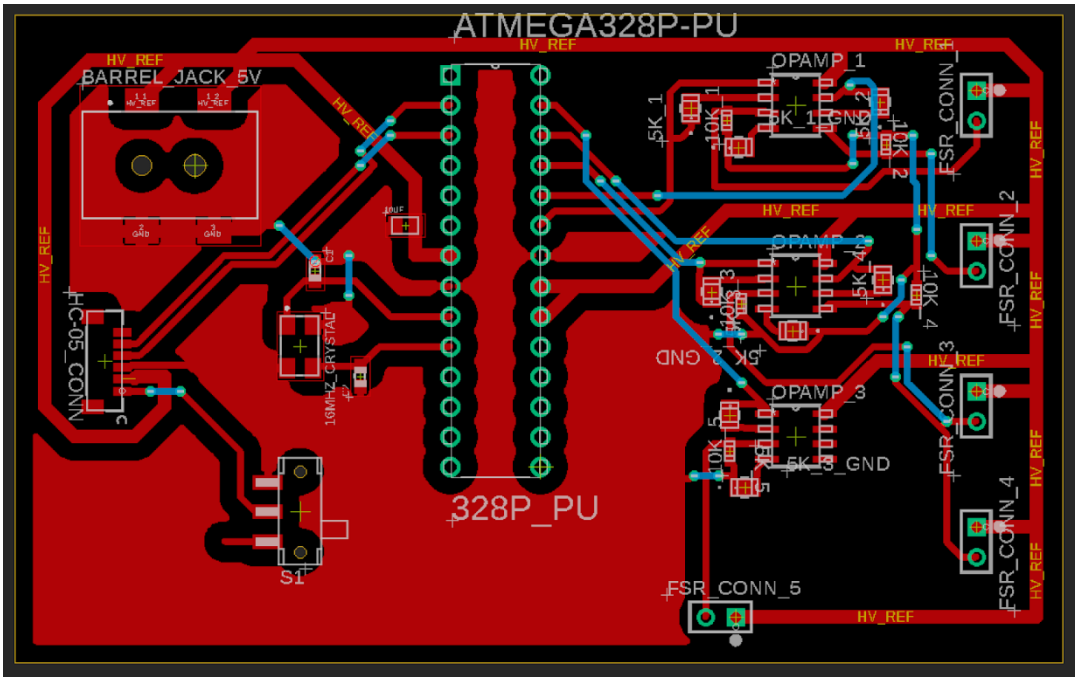


Figure 4: PCB Design

Figure 4 shows what the final PCB would look like. Along the right-hand side and bottom you can see the connectors (FSR_CONN_1 through 5) that would be used for the five FSR402 force resistors to connect to the board. The purpose of these connectors is straightforward - to connect the resistors that the user will be tapping to the main logic of the board. That way the input can be processed and sent via Bluetooth to the mobile phone.

To the left of the connectors are the Op-Amp circuits (OPAMP_1 through 3). These are required to create the voltage divider circuit to accurately measure the resistance across the FSR402 resistors at the tips of the glove for the user. The output of these Op-Amps is what gets connected to the microcontroller for further processing.

In the middle is the ATMEGA328P-PU microcontroller, a socketed version of the ATMEGA328P to allow for easier development. This component will do all the processing on the input from the FSR402 before sending the finger press data to the HC-05 Bluetooth Module.

On the middle left you will see the 16MHz Crystal, the crystal oscillator we will be using to control timing for our microcontroller. The purpose of this is to remove any variances from the internal crystal on the 328P-PU and have a consistent clock to read the user input data from the FSR402.

On the top left corner is our 5V Barrel Jack Connector, which is where the user will plug in the glove to give all our components power.

Finally, on the bottom right corner is the connector (HC-05_CONN) and switch (S1) for the HC-05 Bluetooth Breakout Board. The purpose of the connector is straightforward, to facilitate a connection between the 328P-PU and the mobile phone app via Bluetooth, and the switch is to change the mode of the Bluetooth adaptor between reading data from the 328P or a separate programming mode.

2.2 Circuit Design

Below I will go into more detail on each of the subsystems in the order explained and give further analysis for their implementation where needed.

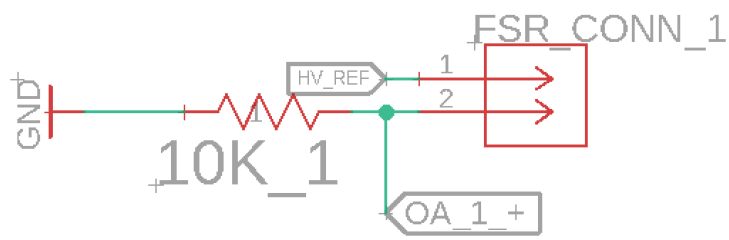
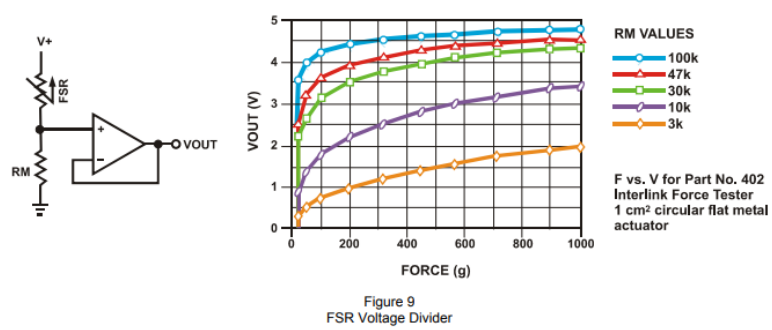


Figure 5: FSR402 Connector Schematic



FSR Voltage Divider

For a simple force-to-voltage conversion, the FSR device is tied to a measuring resistor in a voltage divider configuration. The output is described by the equation:

$$VOUT = (V+) / [1 + RFSR/RM].$$

Figure 6: OP-Amp Voltage Divider

Figure 5 shows the schematic for the FSR402 connector not including the Op-Amp portion. The (10K_1 [4]) is the biasing resistor (RM) from the typical applications example in the FSR402 datasheet. We have chosen this higher biasing RM resistance of 10kΩ as an extra assurance that the user cannot potentially allow too much current to pass through the FSR402 causing it to short and harm the user

(this corresponds to the purple line on the graph in Figure 6). Pin 1 just takes the 5V (HV_REF) input and puts it across the resistor with the Op-Amp configuration shown in Figure 6. The FSR402 also currently would connect to the board through male 2-position through-hole header pins (specifically, the 77311-118-02LF [5]) soldered directly onto those positions on the board (FSR_CONN_1 through 5 in Figure 4). As an improvement in the future, we could upgrade to a true 2-position connector of some kind with proper male and female terminated endings on the board and FSR402 wires respectively.

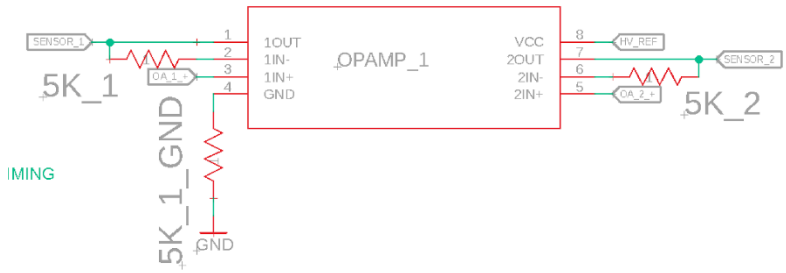


Figure 7: Op-Amp Schematic

Figure 7 shows the LM358 Operational Amplifier (Op-Amp [6]) portion of the FSR402 reading circuit. Once again, Figure 6 shows the circuit that we are recreating on our board. The SMD we chose for the Op-Amp contains two separate Op-Amps on it, so we can handle 2 FSR402 connections per Op-Amp SMD, which explains why we only have OPAMP_1 through 3. The output of this Op-Amp circuit is then connected to the ATMEGA328P via SENSOR_1 through 5. Pin 3 (1IN+) is the input from the FSR402, and Pin 1 (1OUT) is the output to the analog in on the 328P-PU microcontroller.

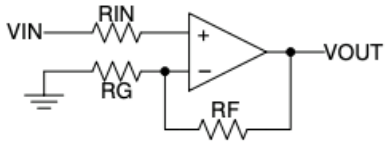


Figure 55. Operational Amplifier Schematic for Noninverting Configuration

Figure 8: Typical Op-Amp Application Circuit

Figure 8 shows how we set up the Op-Amp portion of the circuit, and why we put resistors other than the biasing RM resistor (which was included in Figure 5) in the schematic. The two 5kΩ resistors (RG and RF [7]) are required to vary the op-amp's gain and 5kΩ was chosen to limit current to reasonable levels. Otherwise, we just followed the diagram shown in Figure 6.

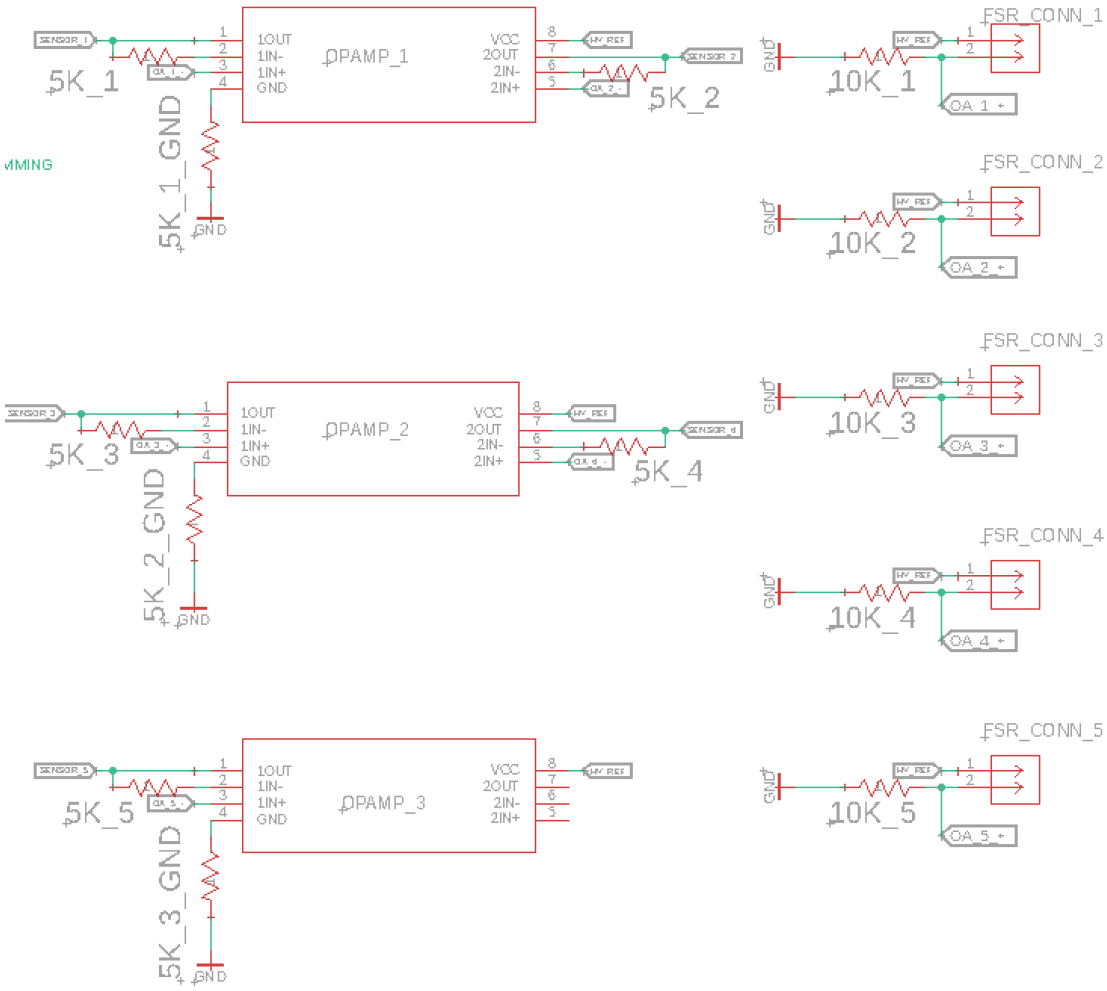


Figure 9: 5 FSR402 Voltage Divider Circuits

Figure 9 shows the full FSR402 portion of the schematic, which corresponds to the entire side of the PCB to the right of the ATMEGA328P in Figure 4. All the outputs from the Op-Amps are then connected to analog inputs on the ATMEGA328P through lines SENSOR_1 through 5. Since we are considering the finger pressure force to be roughly equal across all fingers, we are simply recreating the same circuit 5 times for each of the user's 5 fingers.

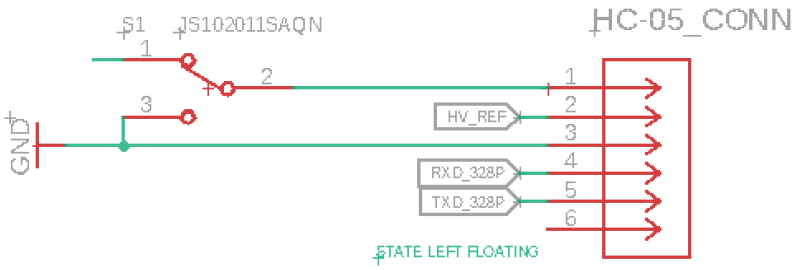


Figure 10: HC-05 Bluetooth Connector

On the left side of Figure 4 is the connector for the HC-05 Bluetooth Breakout Board. The HC-05 will be connected in a similar manner as the FSR402 (via solder pins [8]), but again this can be upgraded in the future to a more robust female-male connector setup. Pin 1 (KEY) is the one which has the switch connected as explained above. Pins 4 (RXD_328P) and 5 (TXD_328P) are the communication pins that connect to the 328P to receive finger press data and send out to the connected mobile phone.

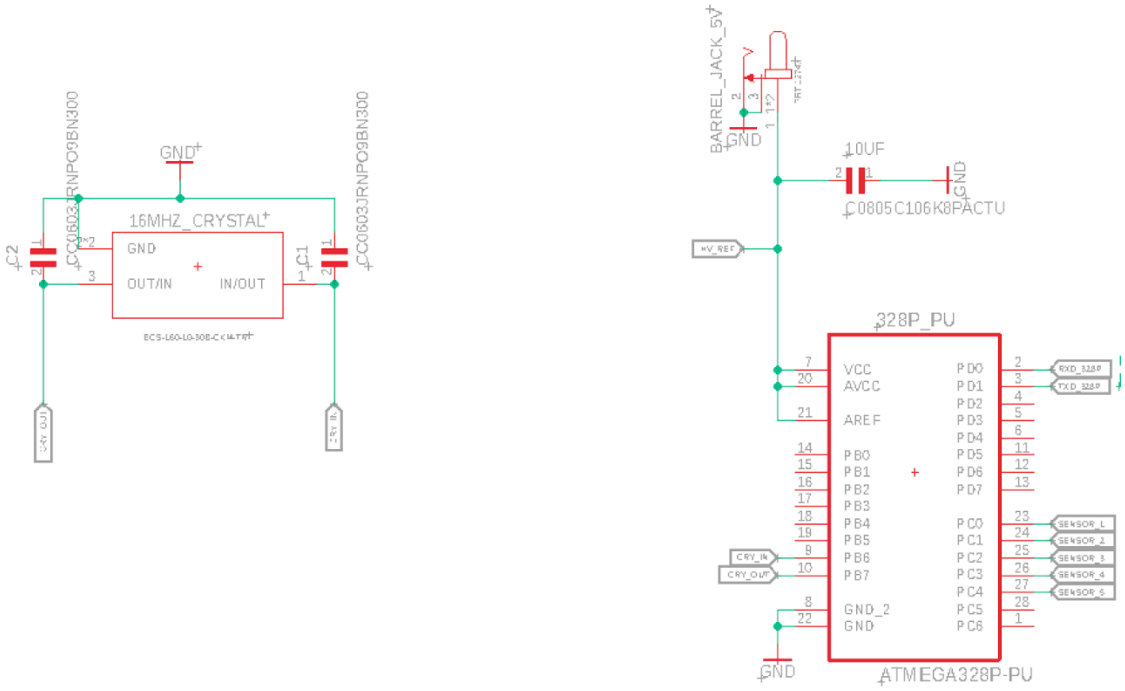


Figure 11: 328P and Oscillator

In the middle of Figure 4 is the microcontroller, the ATMEGA328P-PU [9]. As explained earlier, this chip variant was selected to allow for easy development (as it is socketed, we can program on a debugger board and then insert it into our completed PCB above). Pins 9 and 10 (CRY_IN and CRY_OUT respectively) are where our 16Mhz crystal [10] connects to the 328P-PU. As explained earlier, this external crystal is used to remove any variances in the internal crystal of the 328P-PU and provide a consistent clock for the circuit. The crystal has inductive properties and parallel capacitors are chosen

based on their capacitance to create an oscillator circuit based on the resonance between inductive and capacitive elements. Pin 2 and 3 (RXD_328P and TXD_328P respectively) are where the HC-05 Bluetooth module connects, and Pins 23-27 are all the separate connections for the FSR402 circuits.

We also connected a single 10µF capacitor [11] across the power input to all our subsystems. This was to help reduce some of the ripple from the noisy AC/DC power converter we were using for this project.

2.3 Circuit Simulation and Analysis

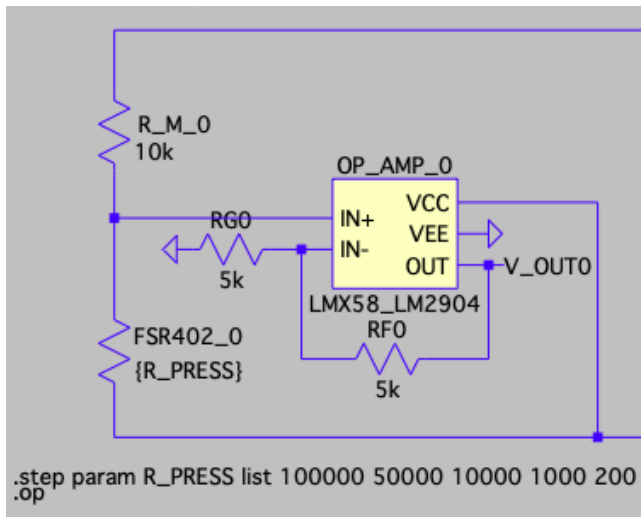


Figure 12: Single branch of FSR402 Circuit

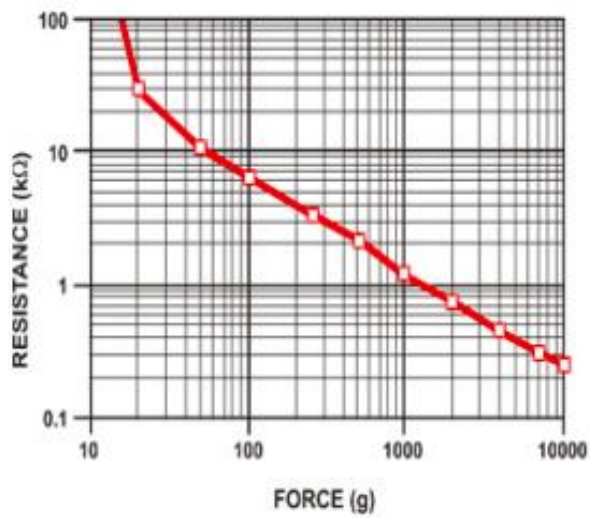


Figure 13: Force vs. Resistance Curve of FSR402 [12]

We created the circuit in LTspice above to model the force-sensing circuit which consists of 5 voltage divider circuits, like that in Figure 12, in parallel. We model the FSR402's resistance with a list parameter containing a likely range of resistances to be seen in real use. To simulate pressing down on the FSR402, the resistance ranges from 100kΩ, which might be analogous to a finger resting on a

surface, down to 200Ω which is analogous to 10kg of force according to Figure 13. It is unexpected for a typical user to press down with 10kg of force, but we also wanted to ensure the sensor would not be overloaded with current so it must be accounted for. 10kΩ corresponds to about 50g of force, the approximate touch weight of a piano key [13], which is the approximate desired force for a press to send the signal to output a sound from a user's mobile device.

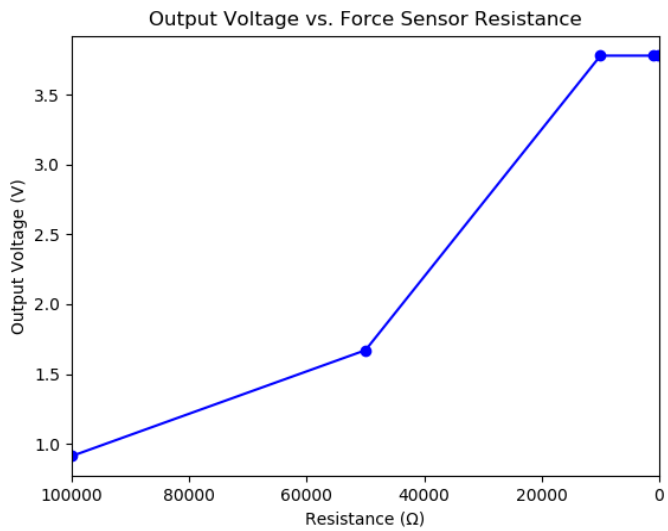


Figure 13: Op-Amp Output Voltage vs. FSR402 Resistance

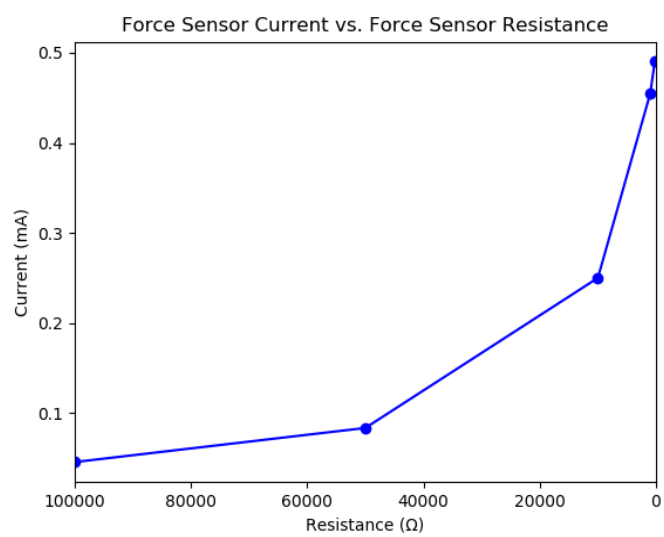


Figure 14: FSR402 Current vs. FSR402 Resistance

Using an external Python library to access raw LTspice data, we plotted the op-amp's output voltage and the current going through the FSR402 against its resistance to yield Figures 14 and 15. Even at 200Ω, the current going through the resistor is about 0.5mA according to Figure 15, which is well below the limit of 1.3mA rated for the FSR402 [12]. With our circuits configured such that components would be at low

risk of overload; we would be able to decide on a digital threshold value in our microcontroller program to constitute a keypress. The ATmega328P has a 10-bit ADC and has a 5V analog reference voltage by default, so if the FSR402's resistance being 10kΩ corresponds to an output voltage of roughly 3.7V, perhaps a good value for a digital threshold would be (3.7 / 5) * 1024 = 758 ± 50.

3. Conclusion

Below we discuss the remaining portions of the project. There are several pieces of our project that we would have liked to implement or expand upon under different circumstances. We also discuss the ethical and safety concerns associated with the project.

3.1 Summary of Implementation

In Chapter 2, we laid out the hardware implementation of the project through the design of a PCB. Within this PCB section, we showed which connectors and parts would be required, the PCB layout, and justification for every component selected. We also proved this PCB design through an LTspice simulation of the exact same FSR402 reading circuit created on the PCB within the program. This simulation showed that the voltage, current, and resistor values were within their required specification and would not pose a hazard to the user. The significance of these two accomplishments is straightforward, as they prove that our design will function as intended and that we have a definitive plan for how to execute the hardware portion of the project. August worked on the PCB, schematic, and chose the specific components that would be used on the physical PCB. Thomas worked on a software-suite to create the graphs for the LTspice model, where Kyle created the model itself.

3.2 Unknowns, Uncertainties, Testing Needed

There were several parts of the project that we were unable to complete due to our lack of access to the lab and time remaining in the semester.

One major portion of the project that we are unable to complete was the Android application and Bluetooth mock device. After discussing with our TA, we decided that this implementation was unnecessary for our proof of concept. Further, learning the framework alone would have taken at least a week and given the time limits of the project, we are unable to create a fully functioning app. To complete this within the full timeframe, we would build a small-scale app to teach ourselves the fundamentals of the framework and how to test it, and then move on to the project proper.

We were also unable to build a working prototype of our glove for a variety of reasons. We do not have access to the ECEB shop to discuss with them wearable technological solutions, as well as a lack of access to the waterproof material we need to comply with safety standards. Further, the wires and casings that would run along our glove, as in the visual aid, are not available to any of us. If we had access to the ECEB and the expertise available there, we would talk with the shop managers, buy supplies (from currently closed stores), and then collaboratively construct the wearable tech.

Another aspect of the project unavailable to us at this time is a physical PCB. Due to the shipping delays caused by COVID-19, we would not be able to order a PCB nor the SMDs, connect it to our glove or test it. In fact, we would need the lab and access to oscilloscopes to test the majority of our electrical

connections. Obviously, we would need access to the lab, where we would connect and then test all our circuits.

3.3 Ethical Considerations

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 [14], 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 regard to the ACM Code of Ethics 2.9 [15]. 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 [16]. 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 [17]. 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 [17]. 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.

3.4 Future Work / Improvements

First, and most beneficially, we would include a lithium ion battery supply on the glove. This would remove the dependency of the glove on a wall socket and allow complete mobility for the user.

Another important improvement would be gameplay functionality on the app side. One criticism of our project was that the user should be provided with a sample song to demonstrate how to use the glove. Our thought is to expand that to make the gloves "playable", with levels that the user can try different sounds and combinations with. This would also allow us to teach the user how to keep time musically, further expanding our target consumer into the musical education and gaming fields.

We could also potentially decrease the latency even further than we have already by choosing hardware that supports a faster audio codec (such as aptX with its latency of ~40ms). Further still, we could program an FPGA to interface with a multi-channel ADC to collect digital data to reduce latency.

4. Progress Made on First Project

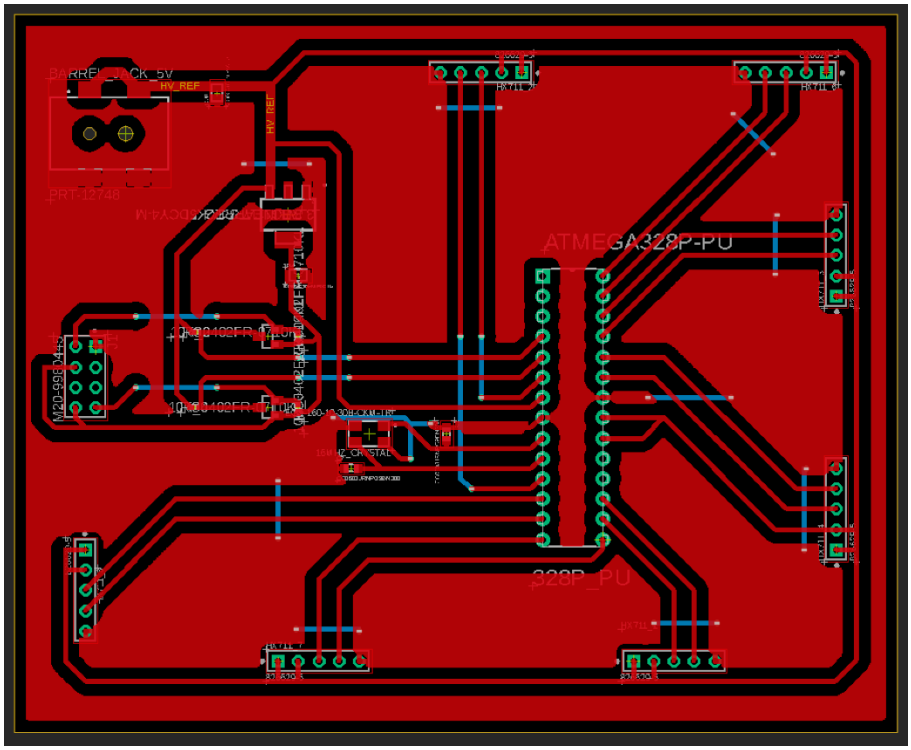


Figure 15 PCB Design Project 1

For our first project, Weightb0ard, we had fully designed out a PCB that would be used to control the electronic portion of the project. Below we will list out each of the components that our board controlled.

Along the top, right, and bottom of the PCB are the 5-position connectors [18] for the HX 711 breakout boards. These connectors directly interface with the ATMEGA328P-PU inputting the digital output from the HX711 about how much weight is being placed on each of the load cells on the board.

Along the left side is the 8-position connector that is used to interface with the ESP8266EX Wi-Fi breakout board [19]. This connects directly to the ATMEGA328P-PU over software serial via a BSS138 logic voltage level converter [20]. This level converter is required to shift the output voltage of 3.3V that the ESP8266EX gives over its data pins to the 5V that the ATMEGA328P-PU requires (and vice versa). This is used to send data about the board's status to the user's phone via a software API.

To the left of the ATMEGA328P-PU there is also the same crystal oscillator [10] used in S0undfingers, as we had the same requirements in both projects. This is also to standardize the clock used within the 328P-PU and remove any potential inconsistencies.

On the top left is the 5V Barrel Jack Connector [21], which is what the user will plug into to give the PCB power.

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] Shop.pimoroni.com. 2020. *MINI.MU Glove Kit – Pimoroni Store*. [online] Available at: <https://shop.pimoroni.com/products/mini-mu-glove-kit?variant=21240542560339> [Accessed 16 April 2020].

[4] "RC0603FR-0710KL," *DigiKey*. [Online]. Available: https://www.digikey.com/product-detail/en/yageo/RC0603FR-0710KL/311-10-0KHRTR-ND/726880. [Accessed: 07-May-2020].

[5] "77311-118-02LF," *DigiKey*. [Online]. Available: https://www.digikey.com/product-detail/en/amphenol-icc-fci/77311-118-02LF/609-4434-ND/1523689. [Accessed: 07-May-2020].

[6] "LM358DR," *DigiKey*. [Online]. Available: https://www.digikey.com/product-detail/en/texas-instruments/LM358DR/296-1014-2-ND/276658. [Accessed: 07-May-2020].

[7] "PNM0805E5001BST5," *DigiKey*. [Online]. Available: https://www.digikey.com/product-detail/en/vishay-thin-film/PNM0805E5001BST5/PNM0805-5-0KBTR-ND/2120328. [Accessed: 07-May-2020].

[8] "1734595-6," *DigiKey*. [Online]. Available: https://www.digikey.com/product-detail/en/te-connectivity-amp-connectors/1734595-6/A99947TR-ND/2077854. [Accessed: 07-May-2020].

[9] "ATMEGA328P-PU," *DigiKey*. [Online]. Available: https://www.digikey.com/product-detail/en/microchip-technology/ATMEGA328P-PU/ATMEGA328P-PU-ND/1914589. [Accessed: 07-May-2020].

[10] "ECS-160-10-30B-CKM-TR," *DigiKey*. [Online]. Available: https://www.digikey.com/product-detail/en/ecs-inc/ECS-160-10-30B-CKM-TR/XC2685CT-ND/8023620. [Accessed: 07-May-2020].

[11] "C0805C106K8PACTU," *DigiKey*. [Online]. Available: https://www.digikey.com/product-detail/en/kemet/C0805C106K8PACTU/399-4925-1-ND/1090920. [Accessed: 07-May-2020].

[12] 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

[13] K. E. Lile, "Piano Finders," Piano Finders: Touchweight [Online]. Available: https://www.pianofinders.com/educational/touchweight.htm. [Accessed: 16-Apr-2020]

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

[15] 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].

[16] 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].

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

[18] "826629-5," *DigiKey*. [Online]. Available: https://www.digikey.com/product-detail/en/te-connectivity-amp-connectors/826629-5/A106751-ND/2276109. [Accessed: 07-May-2020].

[19] R. M. Rogers, "WiFi Module - ESP8266," *WRL-13678 - SparkFun Electronics*. [Online]. Available: https://www.sparkfun.com/products/13678. [Accessed: 07-May-2020].

[20] "BSS138," *DigiKey*. [Online]. Available: https://www.digikey.com/product-detail/en/on-semiconductor/BSS138/BSS138CT-ND/244294. [Accessed: 07-May-2020].

[21] "PRT-12748," *SnapEDA*. [Online]. Available: https://www.snapeda.com/parts/PRT-12748/SparkFun/view-part/?ref=search&t=PRT-12748. [Accessed: 07-May-2020].