𝄞MIDI Mentor 𝄞

Prepsa Ghimire, EE, Neil Guan, EE, Megan Mileski, EE, and Paulina Vu, CE

***Abstract*—Piano is one of the most difficult instruments to play well due to the various requirements. Due to such restrictions, it can be daunting for any musician to pick piano as their first instrument. Many of the existing teaching methods are quite tedious in their methodology and traditionally are fairly expensive. The MIDI Mentor is designed to solve these issues by providing users with an easy method of learning piano at the fraction of the cost of traditional lessons. Our system has a working prototype that is designed to work with 61 key piano and it includes two function glove systems that receives bluetooth signals and turns on haptic motors according to the finger data process on the laptop. Additionally, our system also has an LED strip that indicates to the user what keys they should press along with a GUI that allows the user to have different learning modes and a song selection menu.**

I. Introduction

Piano is notoriously difficult to learn due to a multitude of reasons. Learning sheet music requires a large amount of memorization and music theory knowledge that can discourage beginners. Even if a beginner is able to read music, understanding optimal and correct piano fingering is difficult and not included on sheet music. Knowing which finger to use to play different notes within a song is acquired through experience, creating a barrier of entry for new students. Another challenge is that piano lessons are costly and not accessible to everyone. These challenges can often deter students from learning to play the piano. Our system eliminates these challenges and eases the process of learning piano. The system allows students to learn piano without the knowledge of music theory and the ability to read sheet music. The graphical user interface has a selection screen providing song selection, beats per minute settings, notes per minute settings, and different learning modes. The user will receive multiple cues from the system. The user will receive visual cues from the LED strip attached to the keyboard and also from the falling notes graphic displayed on the computer to indicate which notes should be played and when to play them. The haptic vibration motors attached to the gloves allow the user to feel which finger(s) should be playing the indicated notes. Also, the user will have auditory feedback from the generated sound of the correct notes as they are playing. The scoring feature of the system will provide users the ability to quantify their progress and analyze their performance.

A. Significance

Despite the difficulties, piano is a great instrument for users to start learning. Learning an instrument can improve a student's psyche by engaging the student's creativity. Also, learning the piano can relieve stress. It also provides a healthy activity for young people that can instill good work ethic and patience. Piano is not always a student's top choice when choosing to play an instrument. For lower income families and school districts, piano lessons are inaccessible because of the required cost of lessons needed to learn sheet music and the correct fingerings [1]. According to Piano Belloso, if a student pays $30 per lesson that is 30-45 minutes long weekly for 40 weeks out of the year, they will spend up to $1200 annually just on piano lessons alone [2]. In addition to this, there is an issue over the difficulty to practice since students are given a limited amount of access to such instruments if they don't own one. Users can't practice over weekends or in free time unlike smaller instruments that can be carried. Users of our system could study a piano song without access to a piano. By wearing the gloves they could feel the vibrations from the gloves while listening to the correct note sounds coming from the computer while also watching the falling notes graphic. While the user wouldn't get the same experience of playing the actual instrument they can at absorb the correct notes, fingerings, rhythm, tempo, and sound from wearing the gloves and watching the computer screen. A lot of times musicians refer to the concept of "muscle memory" which could be obtained from our system even without access to a piano at all times.

The current method of learning piano through instructor lessons involving sheet music and music theory is not accessible for everyone. With a different method of learning the piano that doesn't rely on instrument lessons and the acquisition of music theory, learning to play the piano would be attainable to more people.

Figure 1: Traditional image of a student learning the piano with sheet music and a teacher. [3]

B. Context and Competing Solutions in Marketplace

In the past, traditional methods for learning piano were taught by an instructor with a large focus on learning to read sheet music (see Figure 1). This method is still very prevalent and can deter potential students from learning to play the piano. Some of the modern solutions for this problem comes from online music learning programs like Synthesia. Synthesia allows users to learn piano notes for specific songs. Although they avoid the problem of learning sheet music, there is still the problem of determining which fingers should be used for the corresponding notes. These online programs can only offer visual and audio cues, whereas our system can offer vibration sensations indicating the optimal fingerings. Also, even though online programs such as Synthesia can connect to midi pianos, users have to look back and forth between their computer showing which keys to play and the physical piano when playing. With our system, the user is able to identify the correct notes from the LED strip that lights up above the corresponding note allowing users to never have to look away from the keys. This is an important feature of our system because beginners don't have finger placements memorized yet and need to look at the piano while playing.

The MIDI Mentor builds on the approach of self learning implemented in the online programs but includes additional guidance. The optimal piano fingerings is a feature of our system that would previously only be attainable through an instructor's oversight. Also, our system allows users to study piano and develop "muscle memory" even when they are away from a piano. By hearing, seeing, and feeling the notes of a given song, the player can develop their "muscle memory" by associating the correct fingerings to the correct keys while hearing the correct note sounds and rhythm.

C. Societal Impacts

Our constituencies would be educational institutions such as schools and lower income communities, as this system will give them more access to learning how to play piano. This system also targets beginner piano students, students who wish to rekindle their piano skills or students who enjoy self learning. Additionally, this system is for students who want to learn the piano without having to learn sheet music, music theory, or from an instructor.

With these consumers in mind, this is a learning method that is designed to be more accessible than private tutoring. Our system design has a focus on beginner students resulting in multiple aspects of the system functionality geared towards that demographic. We have built our gloves to accommodate users ages 12 and up. It was important that the gloves fit different hand sizes to maximize the user age range.

A potential negative impact from our design is that the demand of piano instructors could decrease if this type of technology takes precedence. This is not concerning because the main target demographic of our system are students that wouldn't take music lessons. Lower income students that can't afford an instructor and students that are opposed to being taught sheet music. Therefore, even if technology similar to our system becomes popular it wouldn't affect the demand for piano instructors. Most students who can afford music lessons will most definitely still want the chance to have individualized lessons with an expert. The instead would extend the opportunity of piano learning to students that otherwise wouldn't have the opportunity to learn piano. Overall, the only possible concern with our system is that it could take jobs from piano teachers, but this seems very unlikely given the target demographic of the system and given that the system could be used in parallel with music lessons.

D. System Requirements and Specifications

We carefully curated a list or requirements for our system to enhance the user experience and useability (see Table 1). It is important that our system is portable and transportable. We designed the system such that it would have a sufficient battery life of at least an hour. We based this requirement to reflect the typical practice time duration. We've designed the gloves with a focus on joint mobility. The gloves should allow the user to bend their fingers and wrist comfortably. Also, the gloves need to have a minimum bluetooth broadcast range of 5 meters to allow separation between the user and computer. This would allow the user to stay paired from across a room and increase the portability of the system.

Apart from the portability requirements, we wanted the system to have multi-sensory learnability functionality. We decided to indicate information through auditory, visual, and touch functionality. These sensory applications are expected to be very precise and synchronized. We set the requirement that the maximal amount of delay between subsystems should be less than 100ms. Another requirement is that the system must be replicable if a user has a different size keyboard piano that deviates from the 61 key piano that we are using.

Requirement	Specification	Value
Portable	Battery life	at least 1 hour
	Convenience	freely bend joints, while secure
	Range	at least 5m for gloves
Functionality	Auditory assist	delay within 100ms
	Sensory assist	delay within 100ms
	Visual assist	delay within 100ms
	Overall system	delay within 100ms
	Versatile	replicable for different keys

Table 1: Requirements and Specifications

II. Design

A. Overview

Our system is composed of four main subsystems which includes: the laptop computer, the LED strip attached to the keyboard, the gloves with the haptic motors and bluetooth circuit, and the MIDI piano keyboard (see Figure 2 for the Hardware Block Diagram). The bulk of the processing is done on the laptop computer. The reason the computing requires a laptop and not a microcontroller, is that a significant amount of processing power is needed to parse the MIDI data and for the fingering generation. After the fingers are generated the data needs to be transmitted over two separate bluetooth low energy protocol connections. The laptop then displays the GUI of the generated falling notes and provides the corresponding note audio. The main python program computes the scoring based on user input (see Figure 3 for the Software Block Diagram). For the synchronization and generation of these applications, the use of a laptop computer is critical and necessary. The LED subsystem includes a custom PCB, which drives an LED strip to light up above the corresponding keys. The haptic gloves subsystem is composed of two custom PCBs (one for each glove) which drives the haptic motors to vibrate the corresponding fingers based on the MIDI file. The MIDI keyboard outputs MIDI data after the user finishes playing that then is used to calculate the user's score.

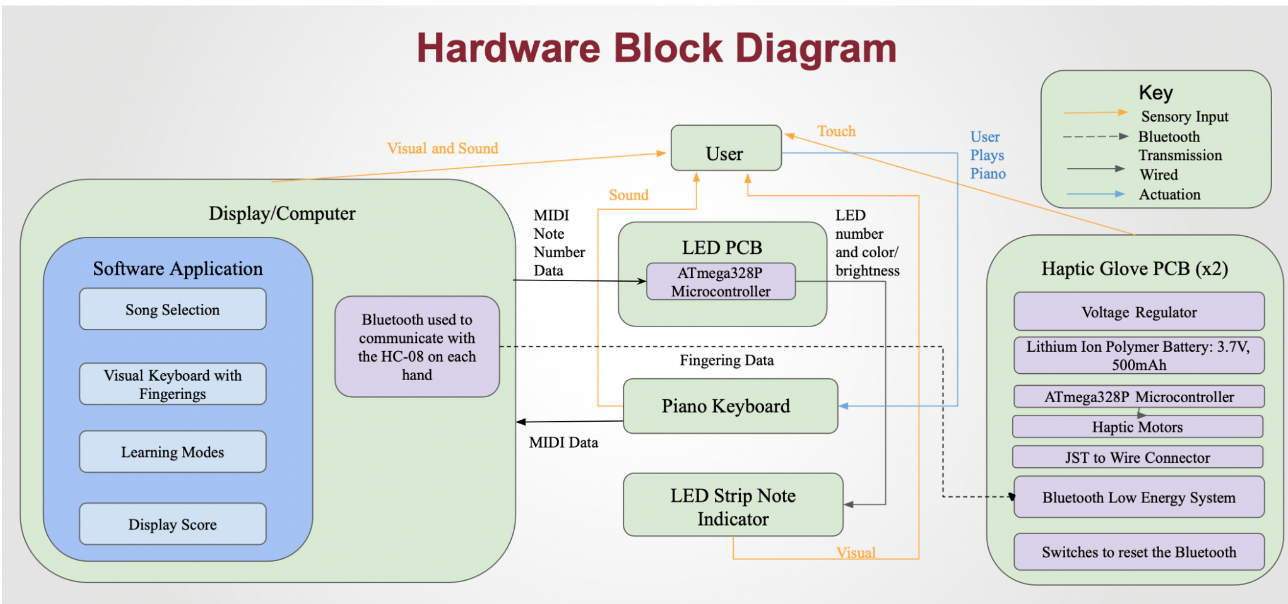

Figure 2: Hardware Block Diagram

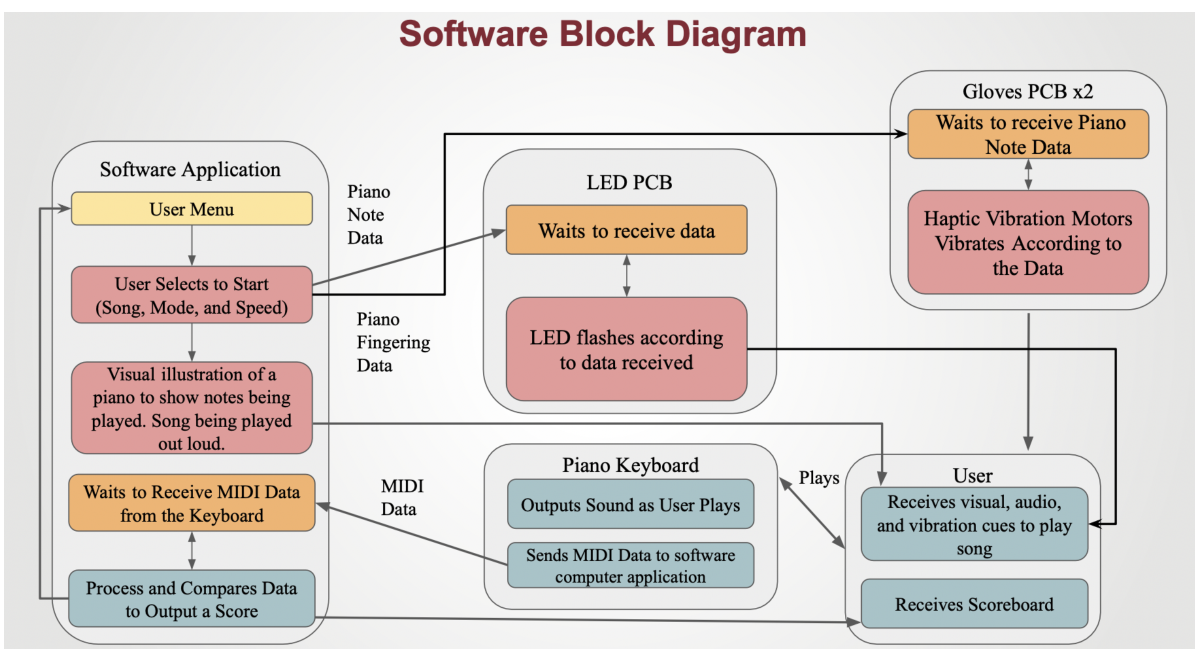

Figure 3: Software Block Diagram

B. Automatic Finger Generation (Computer)

The computer does most of the processing in our system after the user selects the song and learning style. The automatic fingering generation takes in the user's song selection (MIDI or MusicXML file) and generates piano fingerings for each note based on a HMM implementation published in 2020 [4], which is the current state of the art. The library used is Auto Fingering, sourced from a Github repository. Additional work was performed on the source code to make the main file be able to obtain data in a workable format and parse the data to be sent to the Bluetooth Transmitter. This automatic finger generation was also used for the LED Sub System to gather the note number that is sent in bytes via serial communication to the microcontroller on the LED PCB.

C. Graphic User Interface

The Graphic User Interface was created to allow the user to see the length at which they should be pressing the keys as well as to allow users to have autonomy while learning how to play. The GUI displays an 88 key keyboard from the Yamaha website. It was created using Pygame and is synchronized with the LEDs and Haptic Motors. It sends Fingering Data via Bluetooth and MIDI note number data via FTDI Converter Cable. It allows the player to control the beats per minute (BPM), note speed, and single stepping.

The program is organized in a finite state machine that alternates between Game state and Menu state depending on user input. During the Game state, rectangles fall down the screen and reach the keyboard image to visually indicate which key(s) should be played. Once the rectangle hits the piano image, the note sound, which is generated by a software synthesizer, is played. Each frame, the rectangle list is updated based on the relative timing of notes, and various functions and methods keep track of the pausing, single-stepping, and measure-stepping functions. The Menu state, on the other hand, was implemented via the pygame-menu library [5]; the program gives the user the option to select the game speed, BPM of the song, gamemode, and the song itself.

The game state also has two classes, which are scorer and timeTracker, which handles the scoring and pausing functionalities of the game.

For the game state of the GUI, the program keeps track of the correct notes by spawning rectangles at the correct time, which is handled by the timeTracker class. The song selection dictionary [6] that has all of the rectangle X coordinates that have corresponding MIDI valued keys and every previous rectangle's Y coordinate is stored in a list. For every game loop the Y coordinate increments for every index in the list. Within each game loop a rectangle is created for list index values that have Y coordinates less than 531.2 which is the top of the keyboard image.

This interface also displays a score which is a tally of the total of correct notes played at the right timing. The system calculates this through taking the midi note data from the keyboard inputs and compares it to the correct note data displayed. The midi input from the keyboard is stored in a list which is compared to a dictionary containing the midi note data from the fingering generation algorithm. For each frame of the GUI, the midi input list is updated with the newest values from the piano and then compared with the corresponding expected note value. This is repeated consistently for the duration of the song and then the score is output with the total highest possible score.

D. LED Sub System

The LED strip is attached to the piano and it represents keys that need to be pressed. Every other LED on the strip represents a key ranging from 0-60. This whole system works through serial communication. The techniques used for creating this subsystem was learned in the Junior Design Project as well as The Embedded Systems. The LED PCB receives data from the FTDI to Serial Converter Cable then the Atmega328P processes the data and sends it to the LED strip in terms of LED number as well as the color and intensity [7].

E. Glove Subsystem

Using USART configuration of the ATmega328P, the computer will wirelessly transmit the fingering data to the HC-08 bluetooth module [8] using the low energy protocol [9]. We initially used the Adafruit Bluetooth Bluefruit Friend that was low energy. For the proceeding prototype we decided to use the HC-05 bluetooth module. The HC-05 was very ineffective because it was dropping a lot of packets and there were irregular transmission and receive delays. We quickly realized that the HC-05 was not going to work and decided to experiment with the HC-08. After implementing the HC-08, we had really good results. There weren't any packets being dropped and even with the most advanced songs, there were no noticeable transmission delays. Thus, we decided to move forward with the HC-08. We were able to find the castellated SMD version of the HC-08 and implemented this with our custom PCBs. Each glove has a 3D printed box concealing the PCB with the bluetooth module. Attached to each PCB are 5 haptic motores. All of the subsystems are synchronized without any noticeable delays.

III. The Refined Prototype

A. Prototype Overview

Our refined prototypes include everything from our original block diagram, including two different PCBs that help control the haptic motors and the LED according to the user's choice of songs and mode of learning. The GUI consists of a song selection menu that allows the user to pick the song BMP, Notes per minute and other different learning modes that outputs a score. The main two boards, which are fully assembled PCBs, both receive data via Serial Communication or Bluetooth Low Energy. They both are able to either send data to the LED strip or the Haptic motors based on the user's choice of song and other parameters they pick. Below is attached the prototype that we created before the PCBs. Figure 3 includes the prototype for the LED PCB and Figure 4 includes the Prototype for the Haptic Glove PCBs.

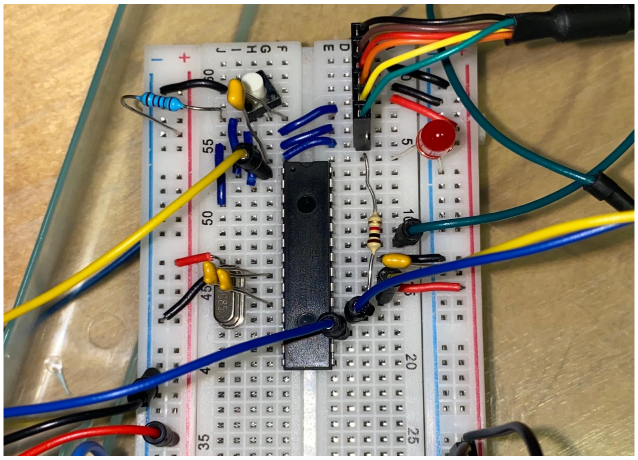

Figure 3: Prototype for the LED PCB.

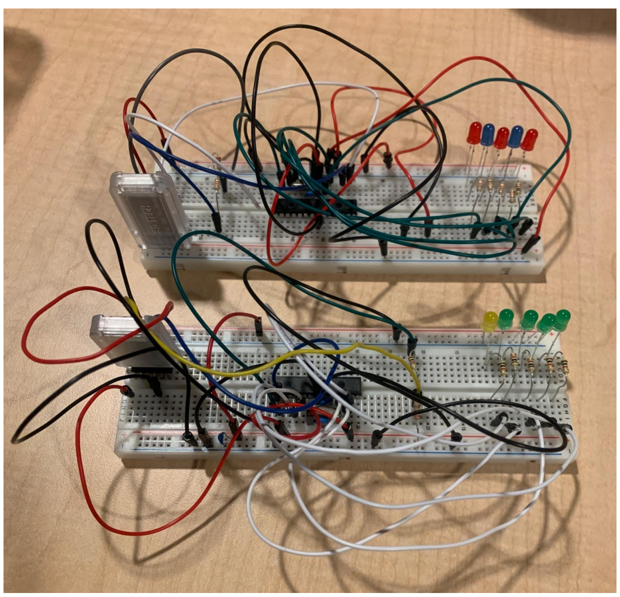

Figure 4: Prototype for the Haptic Glove PCBs.

B. *List of Hardware and Software*

For our system the main hardware components we used include the Atmega328p Microcontrollers, Haptic Motors, HC-08 Bluetooth Surface Mount Devices, LED strip, and other components including resistors and capacitors. Furthermore, our software includes a GUI, written in python, that allows the user to choose a song and their choice of learning style. Additionally, our software also includes C++ code for the Atmega328p Microcontrollers to control the haptic motors as well as the LED strip. The HC-08 receives and sends data to Atmega328p then the data is processed using C++ code in order to turn on the Haptic motors accordingly. Additionally, the other circuit receives data from the FTDI to Serial converter that the Atmega328p Microcontroller processes and sends data to the LED strip.

C. *Custom Hardware*

This system required the creation of two custom PCBs for the led subsystem and the haptic motors subsystem. The led PCB was designed as a baseline for both of our custom PCBs. It was designed to be minimalistic with the functionality of receiving data via a serial connection and powering the corresponding led on a led strip. The ATmega of the system is configurable by SPI and we included an led to indicate if the chip was configured successfully.

Using this baseline, the haptic motors PCB was designed with added functionality of being powered by a lithium polymer battery and includes a power regulator in order to obtain a specified current for our haptic motors. The system is also configured through SPI. Our haptic PCB also contains the HC-08 bluetooth module in order to communicate with our computer wirelessly and is configurable by USART.

D. *Prototype Functionality*

For this prototype, we are able to achieve complete functionality for our diagrams. For the led systems, the custom led PCB was able to communicate with the computer via the serial connection. After configuring the Atmega using SPI, we are able to send the midi note data to the system and obtain an output on the ledstrip as shown in our hardware block diagram (see figure 2). For our haptic motors subsystem, our haptic PCB is also demonstrated to be working properly as intended. After configuring both the atmega and the hc-08 bluetooth module, our program was able to connect with the hc-08 and send data to the system via Bluetooth. This data is converted to signals for the corresponding fingers that vibrate and we obtain an output through the vibrations of the corresponding motors.

E. *Prototype Performance*

Our final system has been able to meet all our requirements and specifications we initially set (see figure 5). Our system is portable due to the fact that our battery life is at least 2.5 hours minimum on continuous play, it is also secure and doesn't hinder the player while they are playing, and it's transportable due to its enclosure that was 3D printed. Furthermore, our system also meets all of its functionality requirements including: having auditory assistance from the laptop (which plays the song in the background), sensory input that the haptic motors on the users fingers, as well as visual assistance from the GUI. To add further, the whole system together with all of the other outputs to the user has a delay less than 100 ms and works synchronously (see figure 6) . The system is also versatile with the user being able to change the piano size in the python code.

Requirement	Specification	Value	System Result
Portable	Battery life	at least 1 hour	2.5 hours
	Convenience	freely bend joints, while secure	comfort: 9.1/10 mobility : 8.7/10
	Range	at least 5m for gloves	8-9 m for gloves
Functionality	Auditory assist	delay within 100ms	avg. delay = 7ms
	Sensory assist	delay within 100ms	avg. delay = 49ms
	Visual assist	delay within 100ms	avg. delay = 34ms
	Overall system	delay within 100ms	avg. delay = 49ms
	Versatile	replicable for different keys	replicable for 88 keys

Figure 5: System Specifications Met

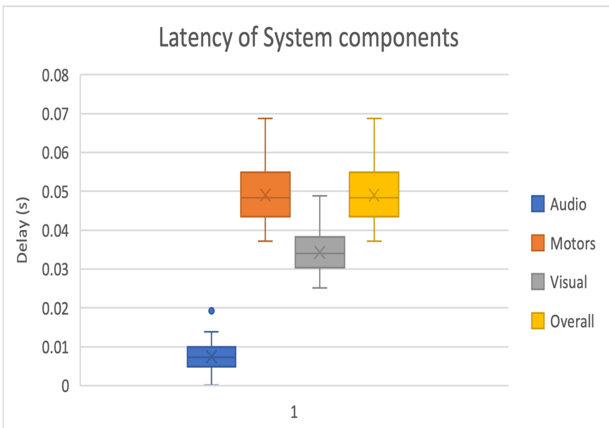

Figure 6: Box and Whisker plot for latency of sub-systems.

IV. Conclusion

Our final prototype (see Figure 7 and Figure 8) is a demonstration that the concept that pianists can be self taught without requiring to have learned sheet music. The haptic piano instructional gloves are able to provide active and passive [10] forms of learning the piano and can be way less intimidating because it allows the player to see their process with the scores. They are able to learn whenever they want and wherever. The GUI allows flexibility in what songs they want to learn as well as stretches of the song length they want to learn. The cost of learning how to play the piano is also heavily reduced.

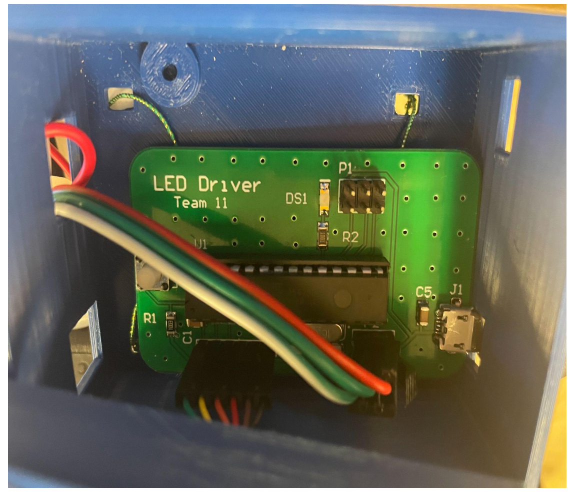

Figure 7: PCBA for the LED PCB.

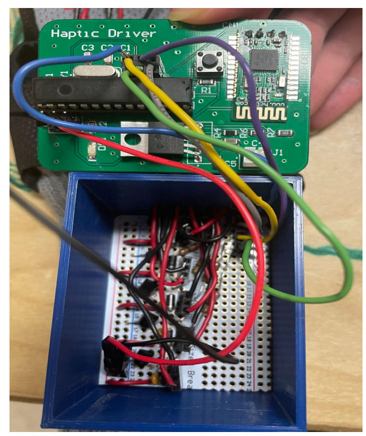

Figure 8: PCBA for the Haptic Glove PCBs.

Acknowledgment

We would like to thank everyone who has helped us with our project, especially everyone at M5. Everyone there has guided us and helped us better our project. Additionally, we would also like to thank our Evaluators, Professor Noh and Professor Soules, for their guidance and feedback. Finally, we would like to thank Professor Anderson for guiding us through this project, and for always being positive and encouraging. We are very appreciative of his time and patiences.

References

[1] A. R. Hoffman, “Compelling Questions about Music, Education, and Socioeconomic Status,” *Music Educators Journal*. [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/0027432113494414. [Accessed: 08-Apr-2022].

[2] I. Belloso, “The hidden cost(s) of piano lessons,” *PIANO BELLOSO*, 08-Sep-2018. [Online]. Available: https://pianobelloso.com/blog/2018/9/4/the-hidden-costs-of-piano-lessons. [Accessed: 08-Apr-2022].

[3] M. & A. m : “Tips on how to become a better piano player,” *The Vault at Music & Arts*, 04-Aug-2021. [Online]. Available: https://thevault.musicarts.com/tips-on-how-to-become-a-better-piano-player/. [Accessed: 08-Apr-2022].

[4] E. Nakamura, Y. Saito, and K. Yoshii, “Statistical Learning and Estimation of Piano Fingering,” *arxiv.org*, 01-Jan-2020. [Online]. Available: https://arxiv.org/pdf/1904.10237.pdf. [Accessed: 08-Apr-2022].

[5] “Pygame Front Page,” *Pygame Front Page - pygame v2.1.1 documentation*. [Online]. Available: https://www.pygame.org/docs/. [Accessed: 10-Jan-2022].

[6] “Python dictionary,” *GeeksforGeeks*, 31-Oct-2021. [Online]. Available: https://www.geeksforgeeks.org/python-dictionary/. [Accessed: 20-Jan-2022].

[7] Evening, Aleksander. “Piano LED Visualizer.” *YouTube*, YouTube, 9 Apr. 2019, https://www.youtube.com/watch?v=IZgYViHcXdM. https://github.com/thegreatkwanghyeon/autofingering

[8] “HC-08 Bluetooth UART Communication Module V3.1 User Manual,” *HC01*. [Online]. Available: https://www.professorcad.co.uk/Bluetooth/HC-08A%20version%20english%20datasheet.pdf. [Accessed: 20-Feb-2022].

[9] G. C. Blog, “The difference between classic Bluetooth and Bluetooth Low Energy,” *The Difference Between Classic Bluetooth and Bluetooth Low Energy*. [Online]. Available: https://blog.nordicsemi.com/getconnected/the-difference-between-classic-bluetooth-and-bluetooth-low-energy. [Accessed: 12-Feb-2022].

[10] C. Seim, T. Estes and T. Starner, "Towards Passive Haptic Learning of piano songs," *2015 IEEE World Haptics Conference (WHC)*, 2015, pp. 445-450, doi: 10.1109/WHC.2015.7177752.

Appendix

A. *Design Alternatives*

We had to make a lot of design alterations before we initially started working towards the Critical Design Review. This is due to us mainly using breakout boards and modules for the Midway Design Review. We initially were looking at many different microcontrollers to pair with the NRF51822 Low Energy that is part of the module we used for MDR before we settled on the Atmega328p. However due to firmware concerns, we also couldn’t use the ultra-low power SoC (NRF51822 Low Energy). After a long time debating and evaluating our resources, we decided to go with the microprocessor on the Arduino breakout board we had used for MDR, which is the Atmega328p. Even though the Atmega328p is older, we weren’t concerned with the clock speed because we were able to create a fully functioning prototype for MDR. Furthermore, due to the fact that we weren’t able to use the NRF51822 Low Energy, we had to find an alternative which we did by looking at past SDP projects. HC-08 SMD is a capsulated device that is able to connect to the computer and communicate with the Atmega328p. It’s fairly small and there was a fairly large amount of resources available to help us connect to the device and debug it.

B. *Technical Standards*

The state of the art for piano fingering estimation uses statistical learning methods based on hidden Markov models which outperforms other models such as those using deep neural networks (Nakamura et al.) as of 2020. We used an implementation of the automatic fingering generation by GitHub user thegreatkwanghyeong to automatically generate the fingerings of each song.

C. *Testing Methods*

We incorporated two sets of experiments in order to confirm that our system specifications were met and that our system was working as intended. The first set of experiments are designed for each of our system specifications.

For the system battery life, we had made a couple of trials in which we would continuously use our system after a full charge cycle. During this time, we would record the amount of time it would take for the battery of the haptic motors to run out. After conducting this experiment multiple times, we were able to get an average battery life that was more than 1 hour.

For the comfort aspect of our specifications, we had intended to get a small group of people with varying hand sizes in order to test if the gloves were comfortable and did not pose any hindrance. We would ask the participants to rate both how comfortable they felt with the gloves and rate the difficulty of bending their joints. These participants would also be asked whether they have any suggestions for our glove design. Based on this feedback, we would make any adjustments and repeat the process until we had a high satisfaction rating on their comfortability and a low rating for any irritation or hindrance posed by the gloves.

For the system range requirement, we would test the system through making repeated connections from the bluetooth to the computer at different ranges. We would start at 1m away from the computer and play a song to ensure that no packets were dropped. If the song is played successfully, we would increase the distance by 1m and repeat the process until we reached 5m. At the end of our experiment, we were able to have a connection between the computer and bluetooth by more than 5m away and the connection was stable as we didn't lose packets for some of the most difficult songs.

For our functionality specification, we would test the system to measure the individual delay through an oscilloscope. We would place the oscilloscope in the correct position and analyze the signals created by the haptic motors and the led strip. Using this data, we can compare with our expected delay between notes and calculate the delay average. This experiment was designed for our visual and sensory delay. For the auditory delay, we could calculate it through a program that would output the difference in real time of the notes being played. This output would give us an estimate of the auditory delay for a song. After multiple calculations of this delay for multiple songs, we would be able to see whether or not our system has met the specifications. For our overall system, we would analyze the delay for each individual delay and note whether the system overall has a delay more than 100ms. Since each system runs in real time, all of our systems are synchronized with the computer which indicates that individual delay does not impact each other. We noted that none of our systems has a delay more than 100ms which was analyzed to be true for the entire system.

For the robustness of our system, we ensured that our design would be replicable to other piano board sizes by making sure our GUI was possible when adjusting for 88 keys.

For our second set of experiments, we will be testing whether our system is beneficial in helping a user learn the piano. We'll have two groups of participants test our system and will give us some qualitative feedback on whether it was beneficial or not. The first group of participants will include users that have little to no experience with the piano and the second group would be those who have more experience or demonstrate mastery of the piano. We would ask our first group a set of questions relating to the ease of use for the system, how effective they perceived it to be, and whether it has changed their stance on learning piano. In addition to these qualitative questions, we would also open up to any suggestions or modifications that they would like to see added. For our second group, we would ask a different set of questions related to whether our system had provided basic structure for a pianist to learn, is this approach efficient, and whether the fingering generation is accurate according to their experience.

D. *Project Expenditures*

Category	Cost
Prototype (Using Modules)	$171.07
Prototype (On Breadboard)	$103.50
PCBs	$54.49
PCB BOMs	$110.29
Total	$439.35

Our project was under budget and created with the intention of reducing waste.

E. *Project Management*

In our team, each individual member is given an important lead position and auxiliary roles. Prepsa Ghimire is our PCB lead and plays a crucial role in development of the PCB design and specializes in the led subsystem. Neil Guan is the lead in our software development and plays the crucial role of integrating all the subsystems. Megan Milesky is our team coordinator and specializes in development of our GUI, glove construction and bluetooth integration. Paulina Vu is in charge of development of scoring code and specializes in working with the haptic PCB.

Our team has great team spirit in which we all work together in the development of all of each component regardless of whether we specialize in that area. Our communication is very good as we typically keep each other informed on all the updates and progress made on the system.

F. *Beyond the Classroom*

1) Prepsa: After creating and completing this project, I've learned how to approach and work in collaborative spaces as an engineer would in the workplace. Additionally, I have also further developed my hard skills including PCB design as well as expanded my knowledge of embedded software design. Not only did I get to learn a new PCB design software, Altium, but I also got a refresher on embedded systems and got to expand my knowledge of the C++ and python programming languages. Overall, I have not only developed new skills, but I also have grown to appreciate the process of product development.

2) Neil: I learned how to approach a greenfield software development project. Software I used for my undergraduate thesis taught me how code should be organized, and the project helped me understand better software practices on a small scale. I've also learned how to collaborate with other people under pressure.

3) Megan: Over the span of this year in senior design project, I was able to refine my C and python coding skills. I also gained experience using Altium. I found the pygame website to be very useful for the GUI design. I also found the website "GeeksforGeeks" very helpful especially for the y-position dictionary used for the GUI. Eventually, I would like to have a professional job that is more software based in the future.

4) Paulina: In order to complete this project, I've had to learn a new application called Altium. In addition to this, I've developed my coding skills for python and c. Stack overflow has been a great resource for understanding possible sources of error and learning debugging methods. I think that having three other teammates also helped a lot since they provide new perspectives on different strategies that could be implemented that are more effective or efficient. I think this experience is extremely valuable as it give me an insight into the typical work structure for engineers.