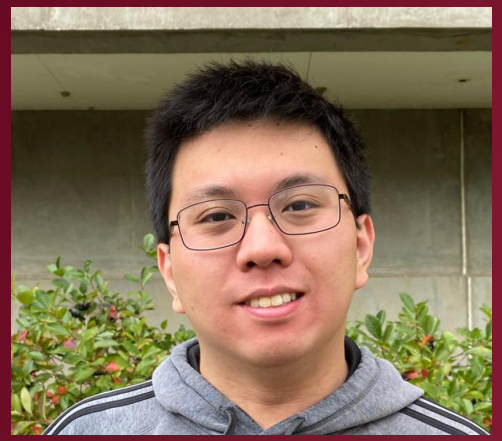
# MIDI Mentor

Neil Guan
Prepsa Ghimire
Megan Mileski
Paulina Vu

University of Massachusetts Amherst
BE REVOLUTIONARY™

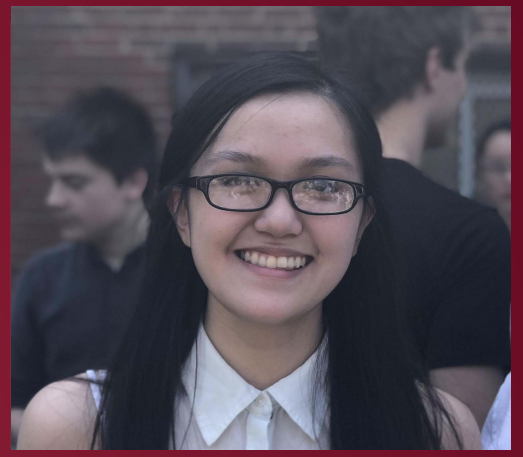
# Team 11



Neil Guan
Electrical Engineer

Megan Mileski
Electrical Engineer

Prepsa Ghimire
Electrical Engineer



Paulina Vu
Computer Engineer

Professor Anderson
Advisor

University of Massachusetts Amherst

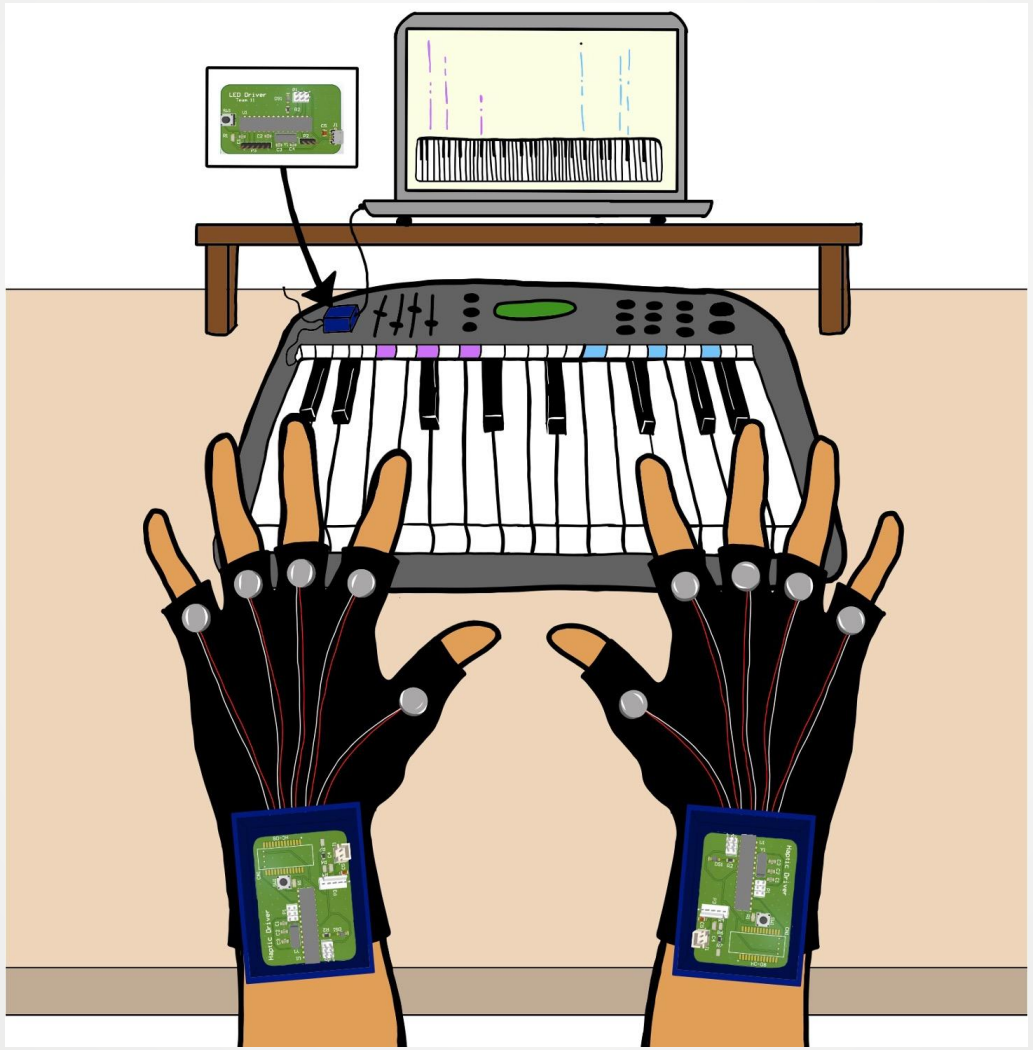
# Problem Statement

Piano is notoriously difficult to learn for several reasons:

- Sheet music does not show the correct fingerings corresponding to which keys to press.
- Sheet music also requires a large amount of memorization and music theory background.
- A great deal of time, effort, and patience is required

Especially for beginners, piano can be a daunting instrument to learn. The mental processing between knowing which keys to press and in what order requires intuition. The memorization, music theory knowledge, and time commitment to learn piano can be discouraging to those wanting to learn the instrument.

# Our Solution: MIDI Mentor



## Solution Explanation

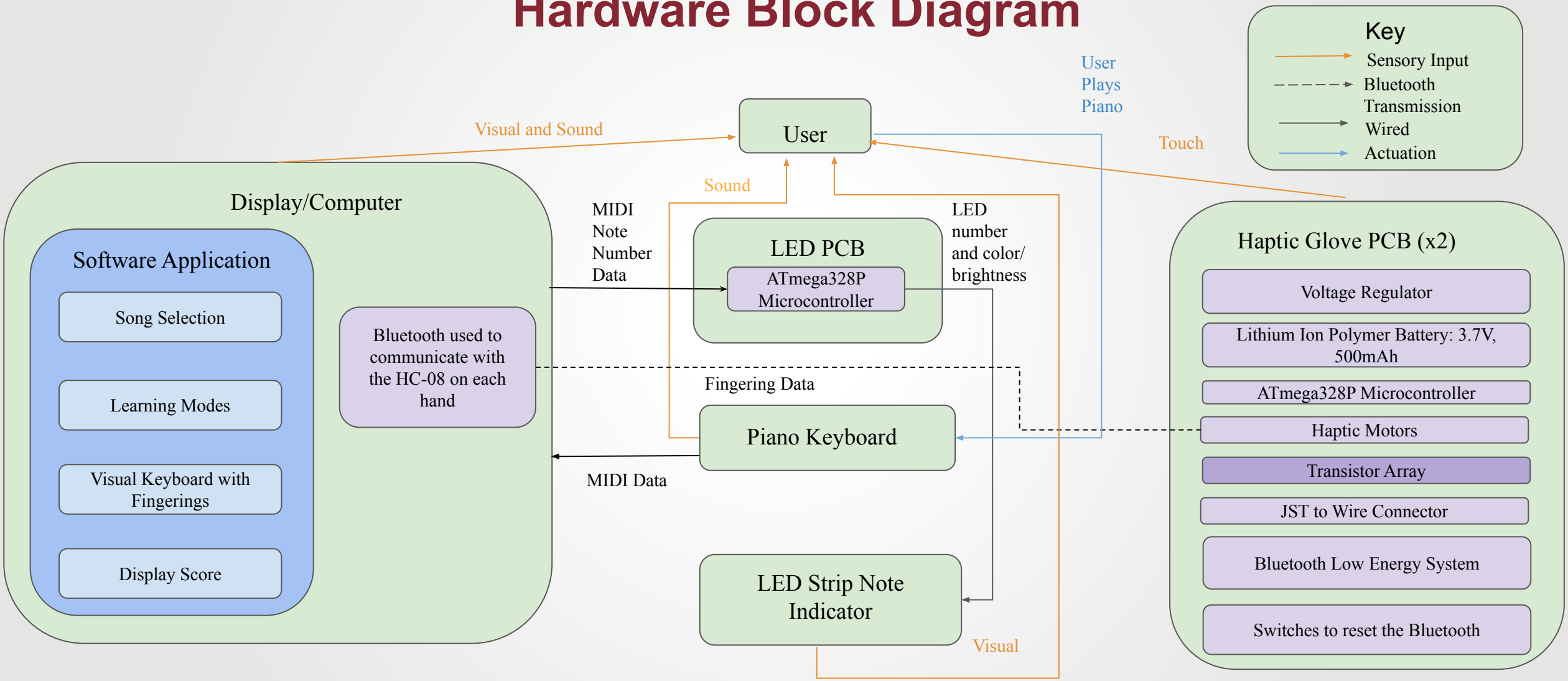
**MIDI Mentor** addresses issues that come with learning how to the piano by:

- Teaching the pianist to play the piano without sheet music
- Providing the pianist feedback on his/her playing
- Helping the user learn while not actively playing the piano
- Engaging multiple of the pianists' senses in real time whilst playing a song--namely touch, vision and sound for active learning

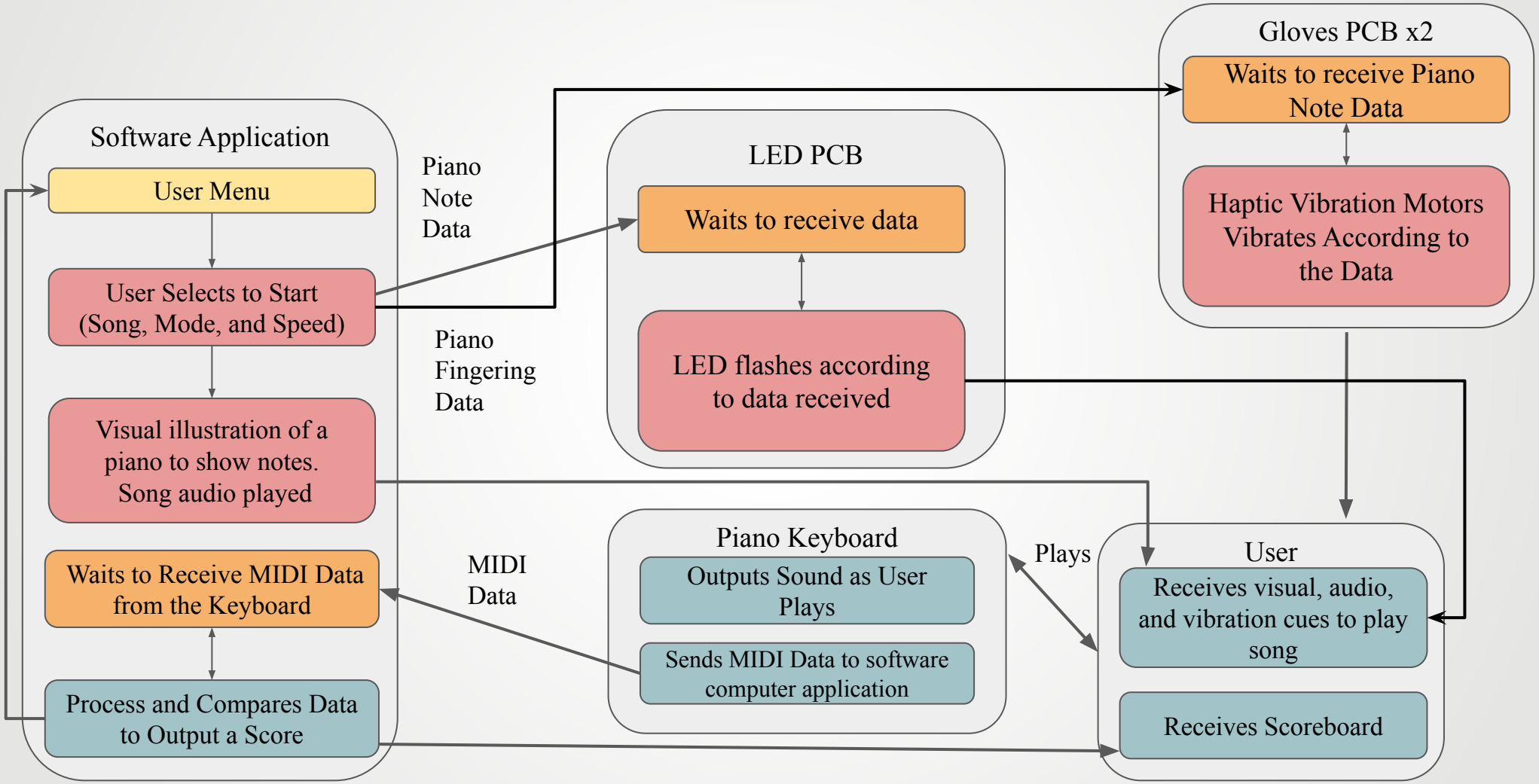
# System Specifications + Verification Plan

| Specification | Verification Plan |
|---|---|
| The gloves are secure and doesn't impose restrictions | Inspection: Gather user data on comfort and mobility |
| The system are replicable for piano key sizes | Inspection: Alterations of the code should allow system to work with any keyboard size. |
| The system provides haptic feedback within a delay of 100 ms. | Analysis: Delay recorded through oscilloscope. Multiple samples taken to compute an average delay for haptics. |
| The system provides visual feedback within a delay of 100 ms. | Analysis: Delay recorded through oscilloscope.. Multiple samples taken to compute an average delay for visuals. |
| The system provides auditory assistance within a delay of 100 ms. | Analysis: Delayed recorded through program. Multiple samples taken to compute an average delay for system. |
| The overall system has a delay within 100 ms. | Analysis: Delay recorded through oscilloscope. Multiple samples taken to compute an average delay for system. |
| The gloves have a range of 5m. | Testing: User tests for maximum range |
| The gloves have a battery life of 1 hour. | Testing: User tests for maximum battery life |

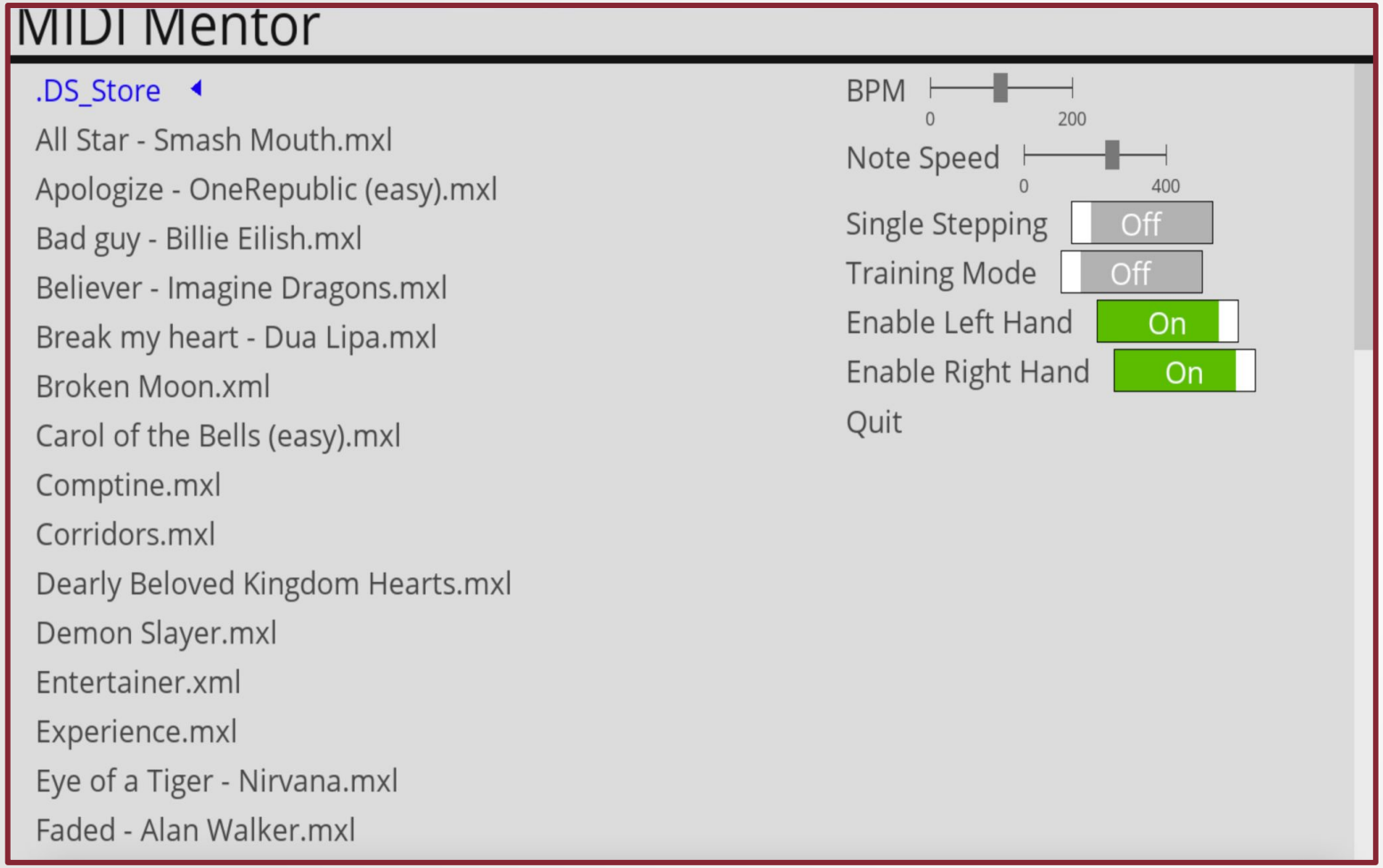
# Hardware Block Diagram



# Software Block Diagram



# Display Menu




Selection menu

- Allows users to control song, speed and mode
- Sends Fingering Data to Haptics via Bluetooth
- Sends MIDI note number data to Led Strip via FTDI Converter Cable
- Left column contains list of song available to play
- Right column contains controls to mode and speed

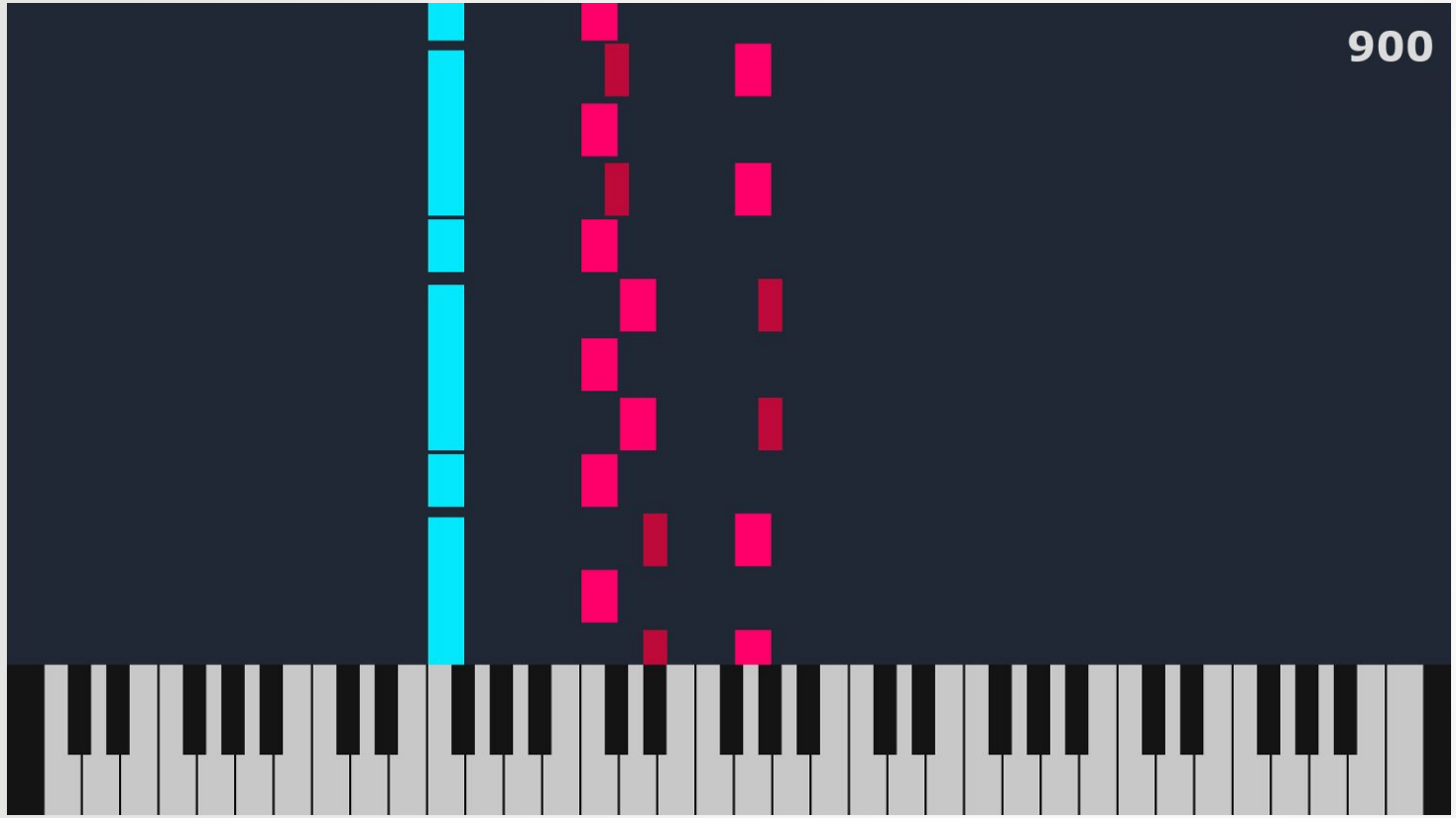
Speed

- Beats Per Minute (Tempo)
- Note Speed

Mode

- Single Stepping ( Play by note )
- Training mode ( Play by measure )

# Display




Display

- Projects the current played notes on to keyboard
  - Includes next sets of notes
  - Length of note matches with delay
  - Left hand = Blue
  - Right hand = Pink
- Process MIDI data received and outputs score for user
  - Displayed on top right and updates whenever user plays correct note

# LED Subsystem

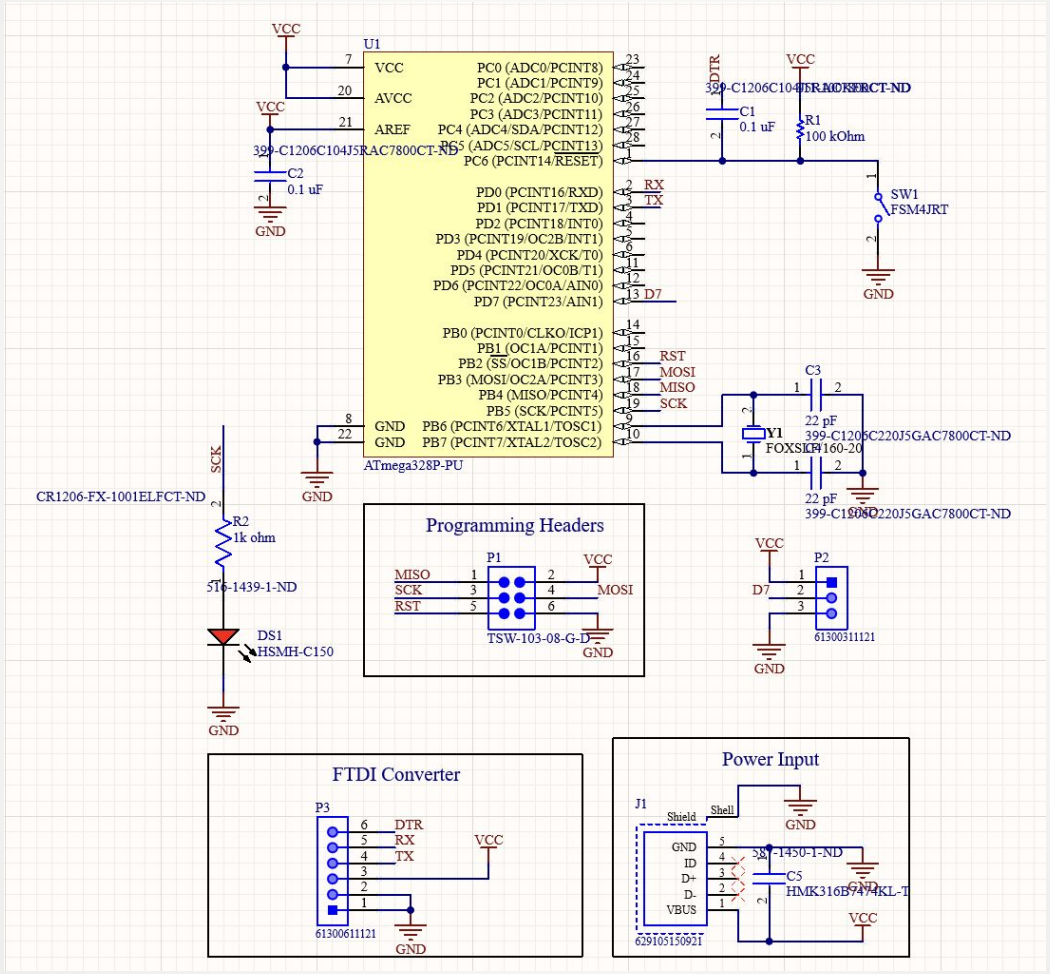

LED PCB

- Uses Atmega328p Microcontroller
- USB to Serial Cable
- Functions:
  - Receives converted Note Numbers (36 - 96) (from pitch (A0 - C8) values) from python via serial communication
  - C++ code receives ASCII value then turns on LED number accordingly

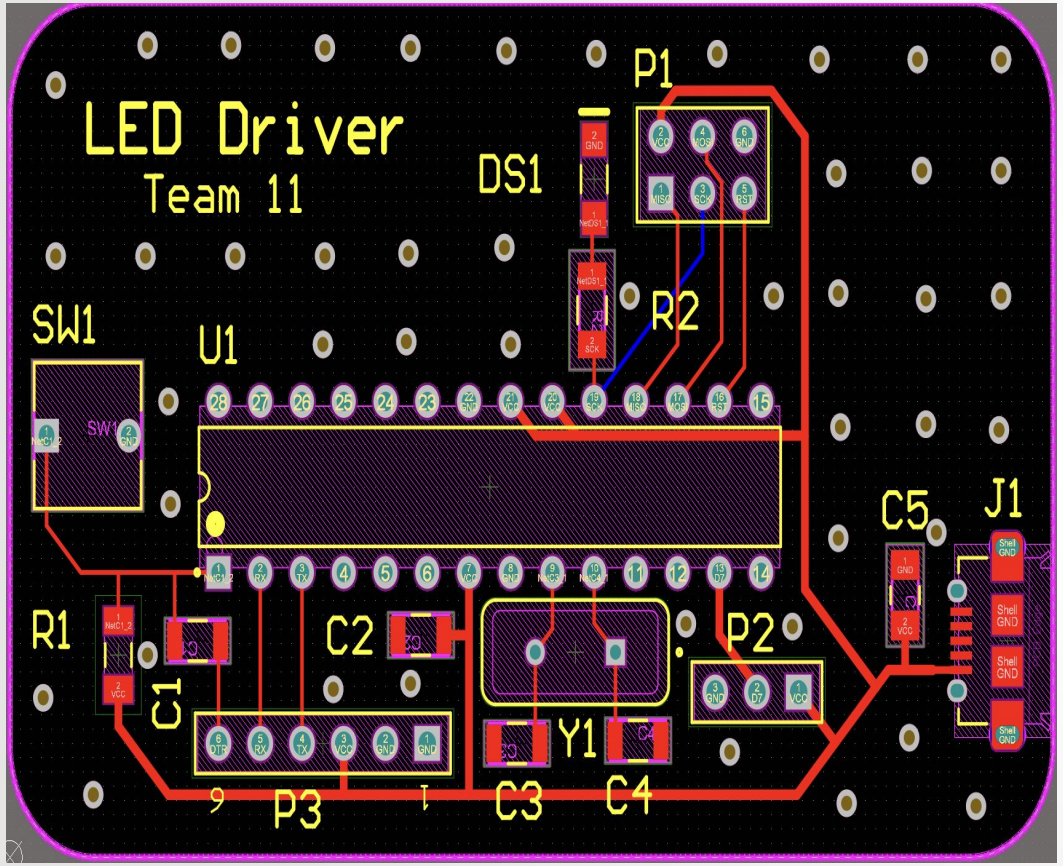
LED Strip

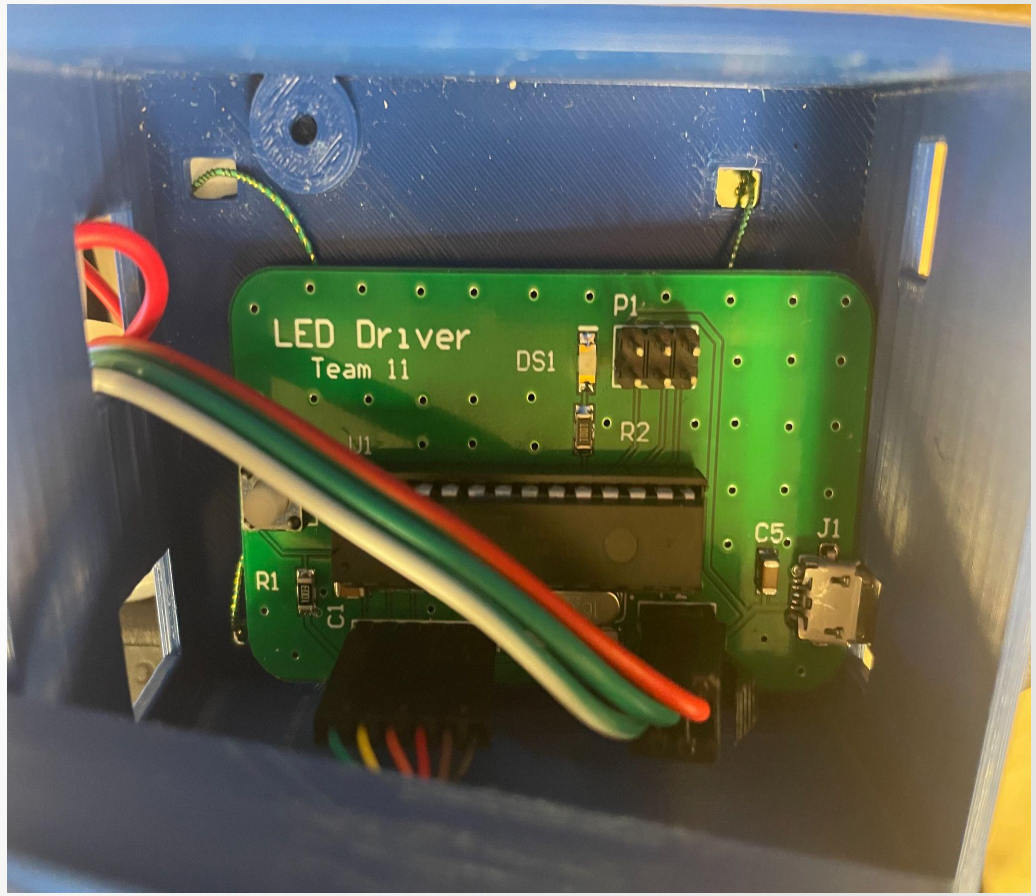
- Lights up to indicate which notes to press
  - Left hand = Blue
  - Right hand = Pink

# LED PCB Schematic

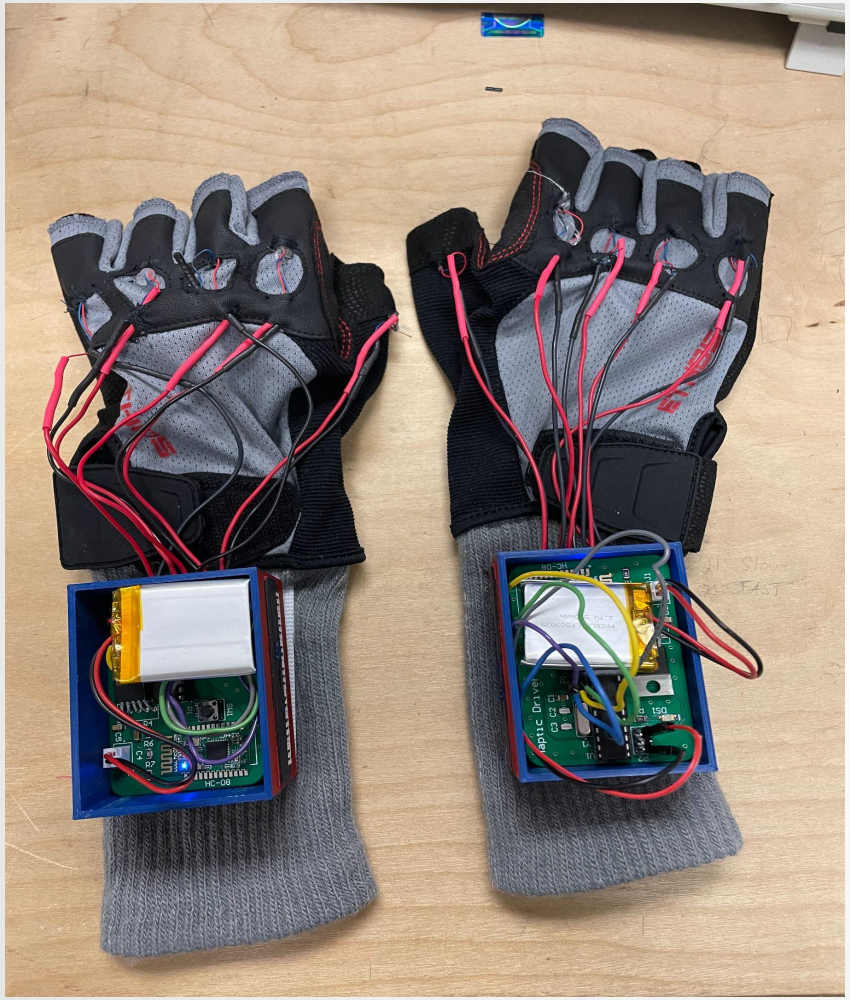


# LED PCB


LED Driver
Team 11
P1
DS1
SW1
U1
R2
C5
J1
R1
C2
P2
C1
Y1
P3
C3
C4


LED Driver
Team 11
P1
DS1
U1
R2
C5
J1
R1
C1

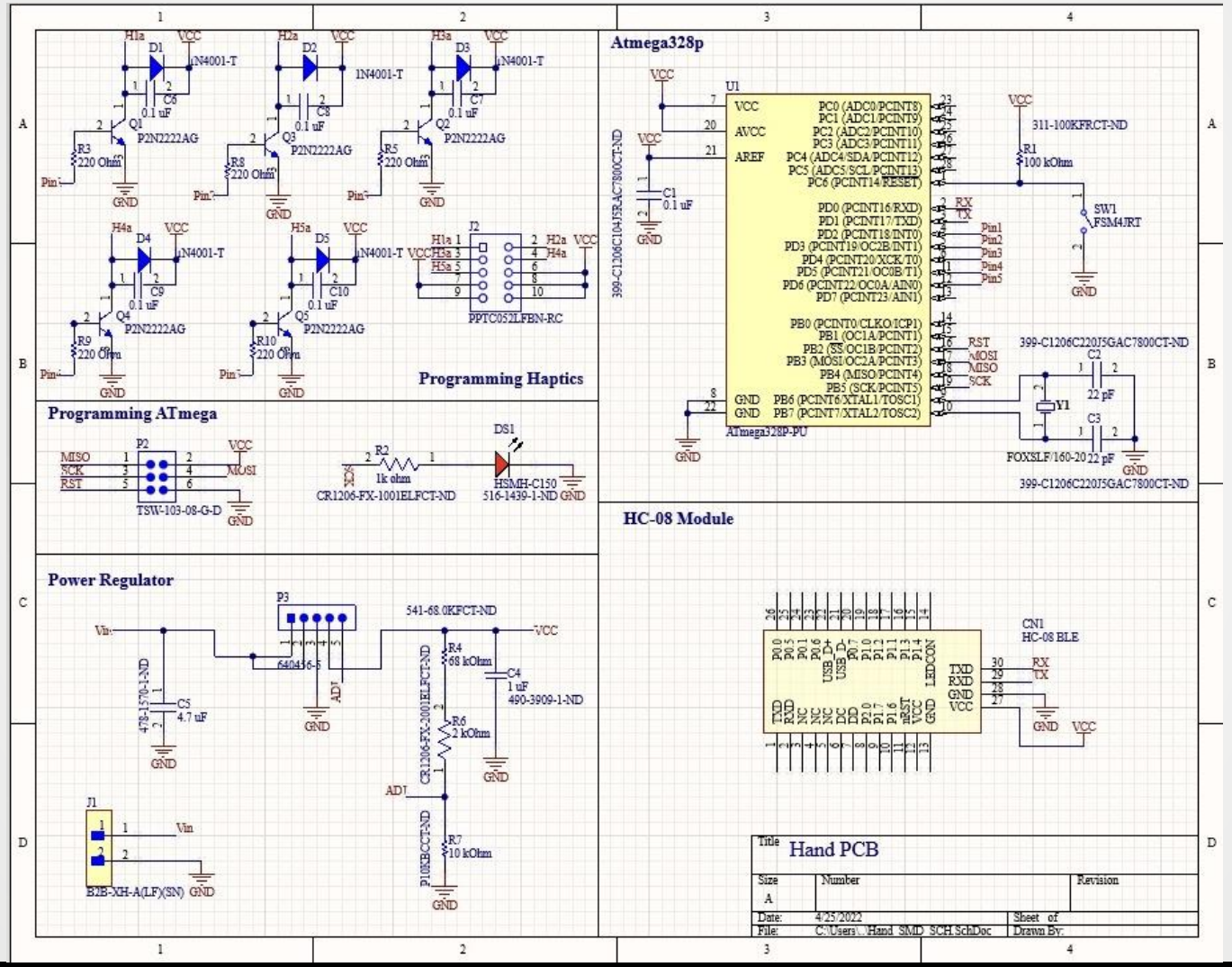
# Haptic Subsystem



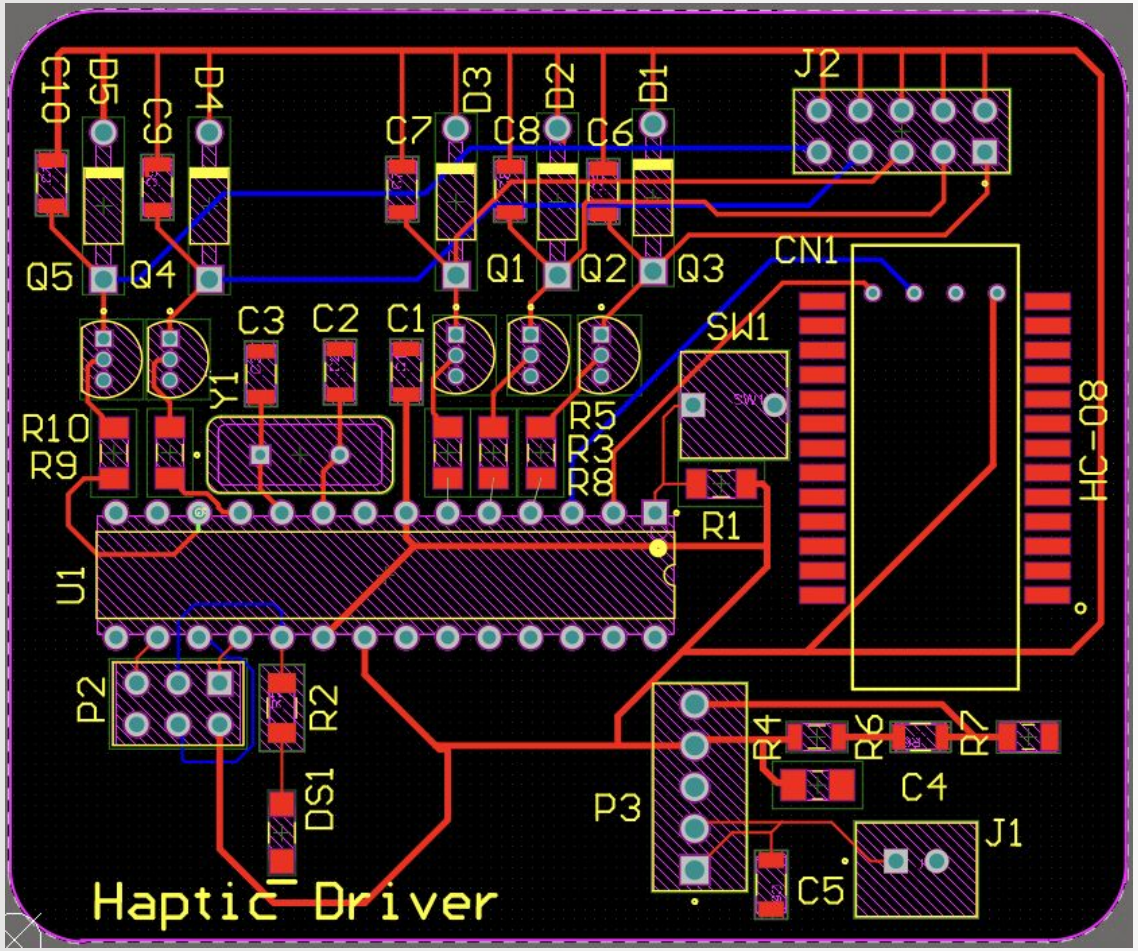
Haptic PCB

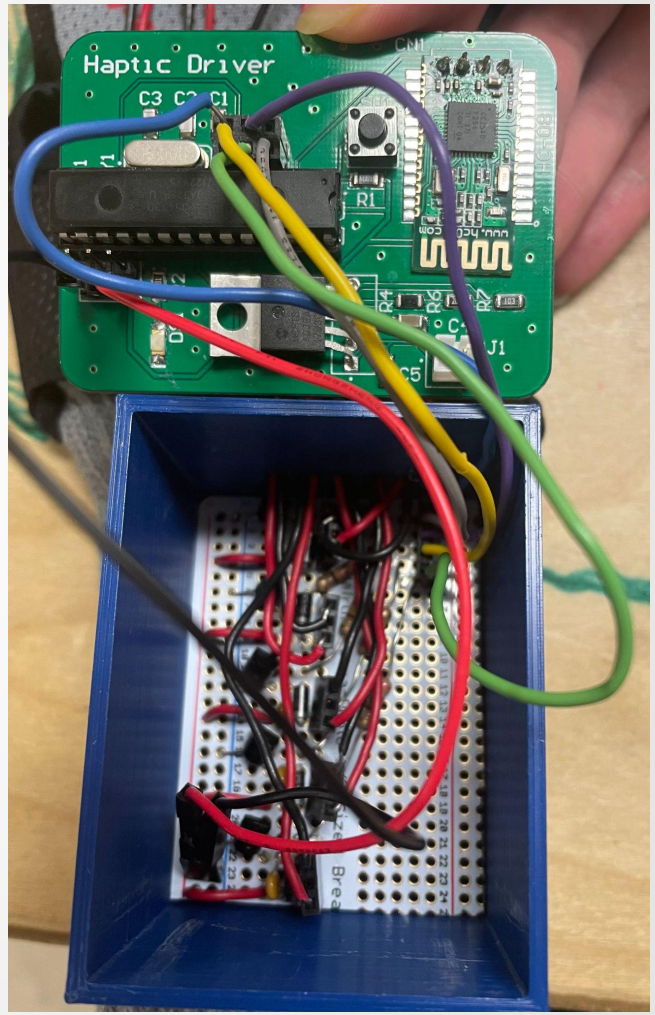
- Atmega328P Microprocessor
- HC-08 Bluetooth Component
- Haptic Motors
- Functions:
  - ATmega328P receives transmitted data over bluetooth and drives the corresponding GPIO pin(s)

# Hand PCB Schematic

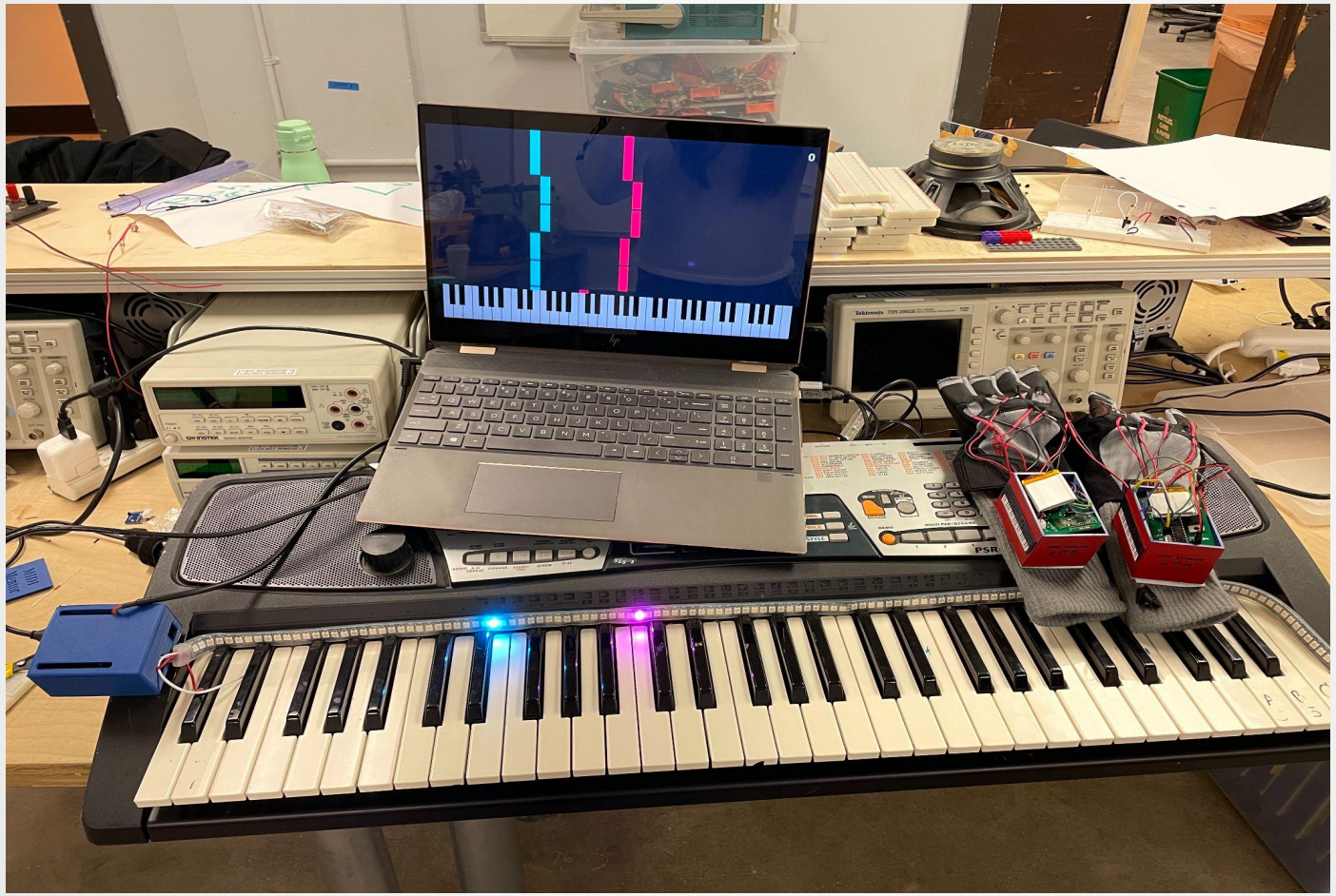


# Hand PCB


D5
D4
D3
D2
D1
C10
C9
C7
C8
C6
J2
Q5
Q4
Q1
Q2
Q3
CN1
C3
C2
C1
SW1
Y1
R10
R9
R5
R3
R8
R1
U1
80-CH
P2
R2
DS1
P3
R4
R6
R7
C4
J1
C5
Haptic Driver


Haptic Driver

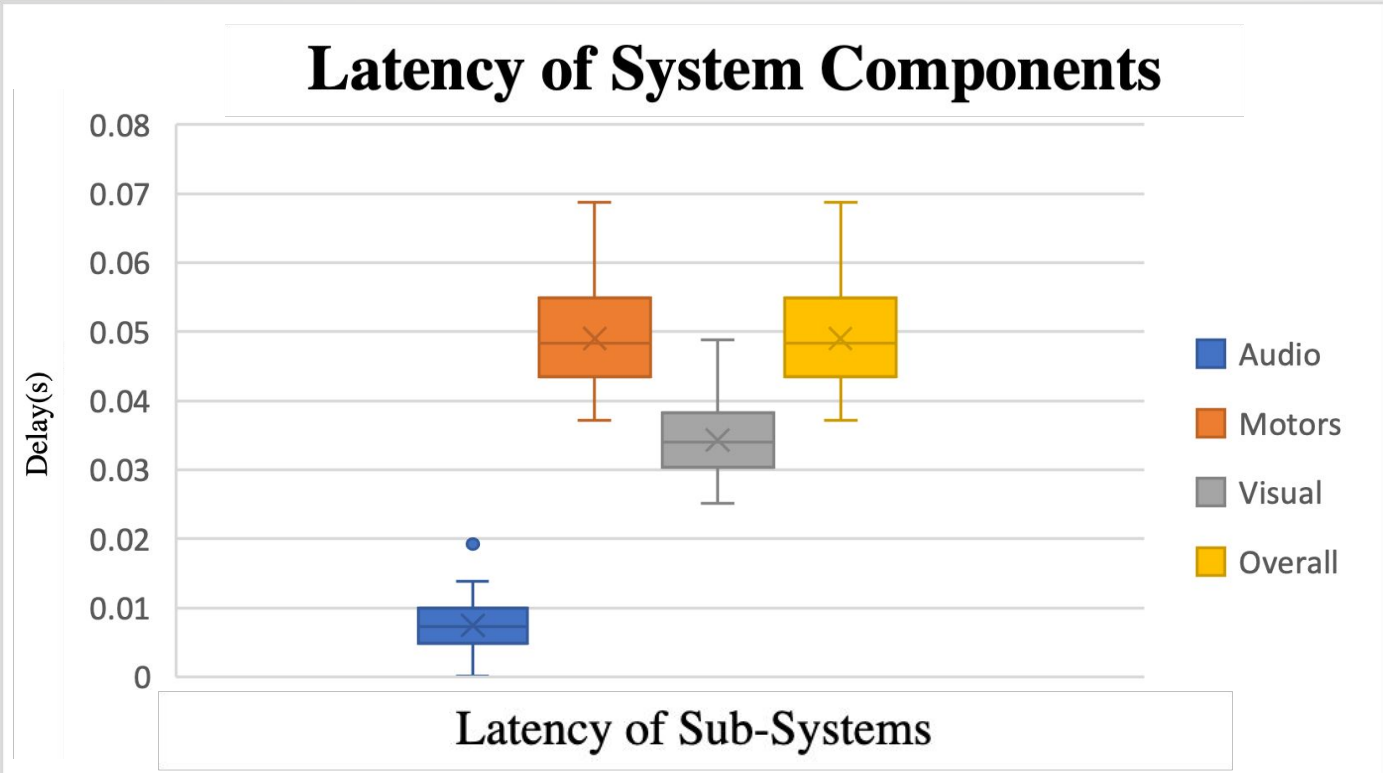
# Final System



# Peer Review

- “Combines those great YouTube videos with a tech-savvy glove!”
- “Snug and sleek glove design”
- “10/10 comfort rating”

# Testing Results




Latency of System

- Average Audio Delay = 7.2 ms, outlier never made system more than 100ms
- Average Motor Delay = 49.0 ms, outlier never made system more than 100ms
- Average Visual Delay = 34.3 ms, outlier never made system more than 100ms
- Overall System Delay = 49.0 ms
  - same as motor delay since it each delay occurred in parallel

# FPR Achievements

| Specification | Result |
|---|---|
| The gloves are secure and doesn't impose restrictions | Comfort Level: 9.1 / 10.0<br>Mobility Rating: 8.7 / 10.0 |
| The system are replicable for piano key sizes | Applicable to apply for 88 keys |
| The system provides haptic feedback within a delay of 100ms. | Average haptic feedback delay: 49.0 ms<br>Standard deviation: 7.5 ms |
| The system provides visual feedback within a delay of 100ms. | Average visual feedback delay: 34.3 ms<br>Standard deviation: 5.4 ms |
| The system provides auditory assistance within a delay of 100 ms. | Average auditory feedback delay: 7.2 ms<br>Standard deviation: 4.1 ms |
| The overall system has a delay within 100 ms. | Average system latency: 49.0 ms<br>Standard deviation: 5.4 ms |
| The gloves have a range of 5m. | Glove range: 8-9m |
| The gloves have a battery life of 1 hour. | Battery life: 2.5 hours |

# Demo of Full System

# Hardware Components

| Hardware | Component | Quantity |
|---|---|---|
| Hand PCBs x2 | Haptic Vibration Actuators | 10 |
| | HC-08 Bluetooth Components | 2 |
| | Voltage Regulator (3.7V to 3.3V) | 2 |
| | Various Resistors<br>(220 Ohms (10), 1k Ohms (2), 100k Ohms (2), 10k Ohms (2), 2k Ohms (2), 68k Ohms (2)) | 20 |
| | Lithium Polymer Batteries | 2 |
| | ATmega328p | 2 |
| | 1N4001 Diode | 10 |
| | P2N2222AG NPN Transistor | 10 |
| | Connectors<br>(JST to Wire Cable, Pin headers) | 3 |
| | Various Capacitors<br>(0.1 uF (12), 22 pF (2), 0.47 uF (2), 1 uF) | 14 |

| Hardware | Component | Quantity |
|---|---|---|
| Hand PCBs x2 | 16 MHz Crystal Oscillator | 2 |
| LED PCB | Micro USB Connector | 1 |
| | LEDs (WS2812B LED) | 3 |
| | Various Capacitors<br>(0.1uF (2), 22 pF (2), 0.47uF (1)) | 5 |
| | Various Resistors<br>(1k Ohm (1), 100k Ohm (1)) | 2 |
| | FTDI Serial Converter Cable | 1 |
| | ATmega328p | `1 |
| | Pin Headers | 3 |
| | 16 MHz Crystal Oscillator | 1 |

# Software Components

| Software | Component | Use |
|---|---|---|
| Display | MusicXML Files (or MIDI) (MuseScore) | Songs ranging from C2 – C6 for the 61 Keys |
| | Python 3<br>- Libraries Used: asyncio, mido, auto fingering, pandas, music21, numpy, serial, pygame, time, bluetooth.uart, bluetooth.message, pygame_menu, random | Used to coordinate multiple functionss, interface with MIDI input, generate fingerings, manipulate data, parse MusicXML/MIDI files, connect to bluetooth, process numbers, and connect to LED strip via serial |
| | Bluetooth (Python) | Used to coordinate multiple functions, interface with MIDI input, generate fingerings, manipulate data, parse MusicXML/MIDI files, connect to bluetooth, process numbers, and connect to LED strip via serial |
| LED Sub-system | Microchip Studio (C++) (LED) | Piano note data sent to the Arduino via pyserial from the python code. Python code sends bytes corresponding to the LED number ranging from 0-61 and the corresponding color (EX: 0b0100001) |
| Haptic Sub-system | Microchip Studio (C++) (Haptic Motors) | Fingering data is sent to the bluetooth component that is then processed by the microcontroller to turn on the corresponding GPIO pins |

# Final FPR System Cost

| Component | Quantity | Cost |
|---|---|---|
| Haptic PCBA | 2 | $35.00 |
| LED PCBA | 1 | $8.00 |
| LED Strip | 1 | $10.00 |
| Gloves | 1 | $30.00 |
| **Total** | | **$83.00** |

# Total FDR Cost

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

# Citations

K. Huang, E. Y. Do and T. Starner, "PianoTouch: A wearable haptic piano instruction system for passive learning of piano skills," *2008 12th IEEE International Symposium on Wearable Computers*, 2008, pp. 41-44, doi: 10.1109/ISWC.2008.4911582.

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.

https://github.com/thegreatkwanghyeon/autofingering

Nakamura, E., Saito, Y., & Yoshii, K. (2020). Statistical learning and estimation of piano fingering. Information Sciences, 517, 68-85.

# Live Demo!

# Questions?