SuperFret

Team A2: Owen Ball, Tushaar Jain, Ashwin Godura
18-500 Capstone Design, Fall 2023
Electrical and Computer Engineering Department
Carnegie Mellon University

Product Pitch

Beginner guitar players have trouble associating fingering positions in images or guitar tabs with physical locations on the fretboard. SuperFret addresses this by guiding the user with lights. Our product has 2 modes. In “Training” mode, SuperFret waits for the user to strum the correct note. In “Performance” mode, the system flashes LEDs according to a song file and the user tries to keep up. In both modes, SuperFret will record the user’s performance and displays feedback in the form of statistics on the intuitive web app.

SuperFret provides **4 hours of fun on a charge!** With SuperFret’s **sub-2 millisecond LED latency and 99% strum and finger placement sensing**, you can play with confidence and trust the system will keep up, even as your skills rapidly progress.

System Architecture

The SuperFret system has 3 main subsystems:

1. **Frontend UI**: Constantly requests updates from the Raspberry Pi (RPi) over HTTP for low latency synchronization
2. **Raspberry Pi 4B**: Orchestrates communication between front-end user commands and back-end user feedback and maintains consistent system state
3. **Teensy 4.1**: Executes songs on the LEDs and analyzes the user’s playing. Reports playing data back to the RPi

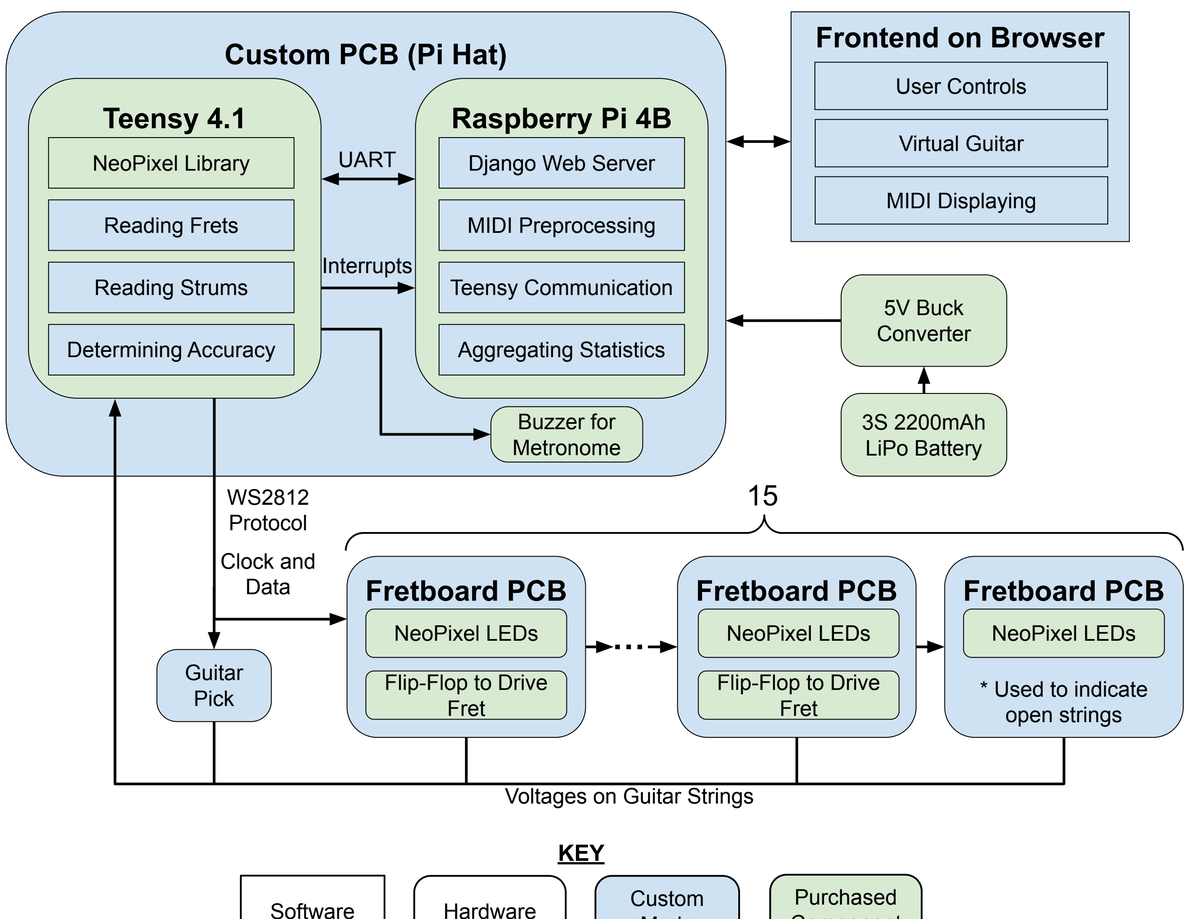

Figure 1: System Block Diagram

Conclusions & Additional Information

Project documentation and weekly updates

https://course.ece.cmu.edu/~ece500/projects/f23-teama2/

SuperFret surpassed all expectations we set and has proven to be an enjoyable and effective way to learn guitar. Throughout the project, we faced numerous challenges including timing violations induced by signal propagation time and synchronizing the web app with user playing. We were able to overcome these challenges through collaborative debugging and integration testing. In the future, this project could be expanded on by using finger placement and strum detection to create MIDI files and sheet music as the user plays.

System Description

Software

➢Web App
- Allows the user to upload and select MIDI files
- Displays upcoming notes visually on a guitar

➢MIDI Parsing
- Converts MIDI notes into string-fret tuple

Firmware

➢Teensy
- Receives song and settings from web app
- Directs LEDs to light up along with song

Hardware

➢Fretboard PCBs
- Display target note using NeoPixel LEDs
- Use D-flip-flops to drive each fret high sequentially

➢Pi-Hat PCB
- Connects the Teensy and RPi with GPIOs and UART
- Breaks out Teensy I/O to control hardware

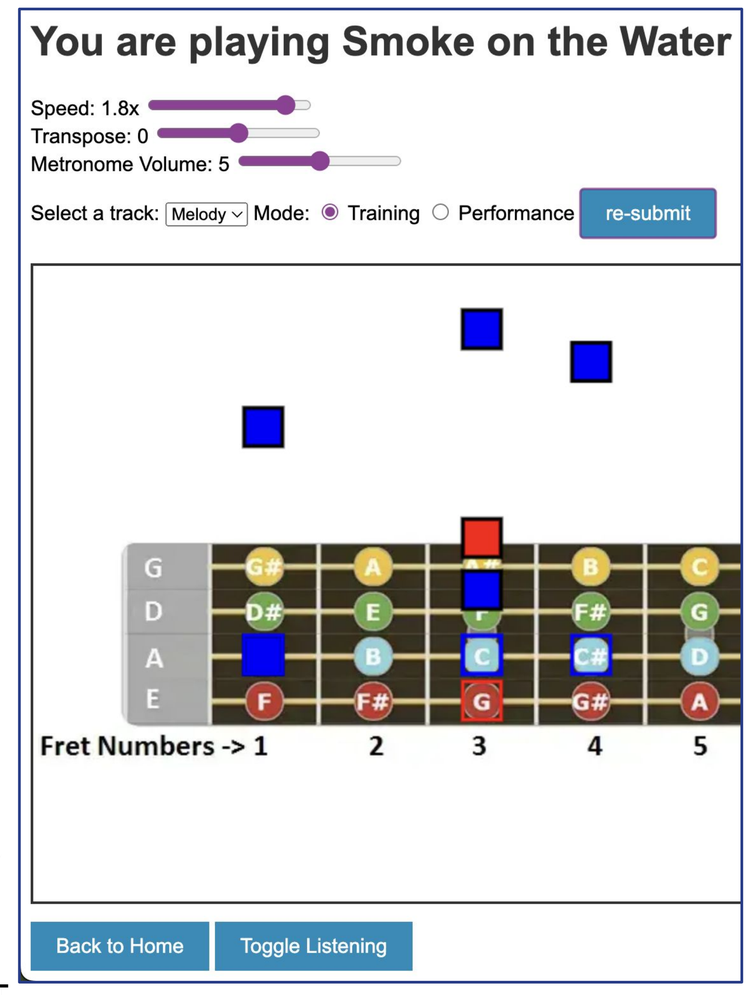

Figure 2: Virtual Guitar on Web App

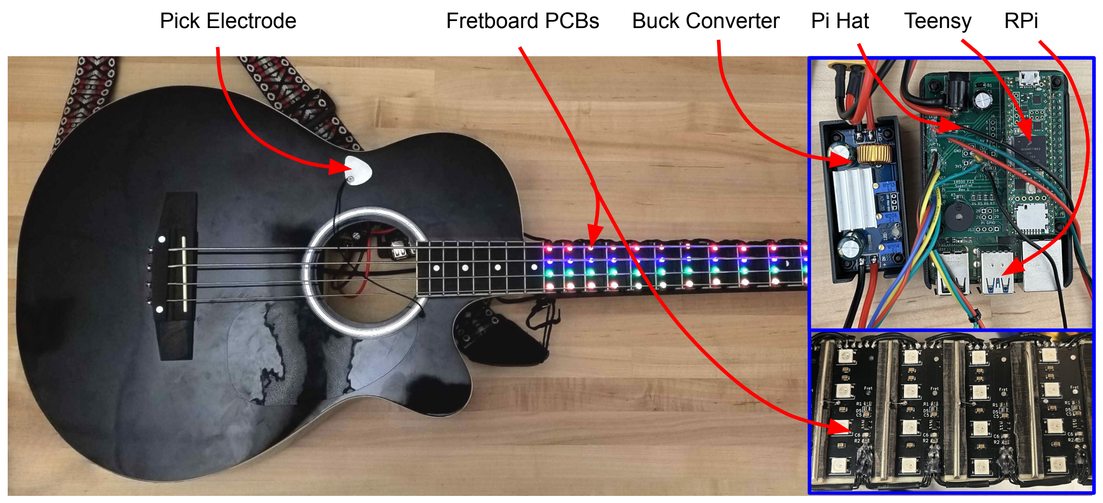

Figure 3: Full Guitar Assembly

System Evaluation

To ensure the system provides a positive user experience, numerous use-case and design requirements were developed. Using lab-bench ammeters and oscilloscopes, along with various accuracy testing procedures, we were able to verify that the system met all our target requirements.

Metric	Target	Actual
MIDI to fretboard LED conversion accuracy	100%	**100%**
Finger placement detection accuracy	≥99%	**100%**
Strums per minute supported	≥200	**300**
Strum detection accuracy	≥99%	**99%**
Latency from strum to LEDs updating in response	≤50ms	**1.85ms**
Latency from strum to web app updating in response	≤250ms	**215ms**
Average current through body possible	≤1mA	**5.37μA**
Total system current with all LEDs at 50% brightness	<4.5A	**0.96A**

Figure 4: Test Results Table

Electrical & Computer ENGINEERING
Carnegie Mellon