Carnegie Mellon University

Team A2: SuperFret

Owen Ball, Ashwin Godura, Tushaar Jain

Use Case Requirements

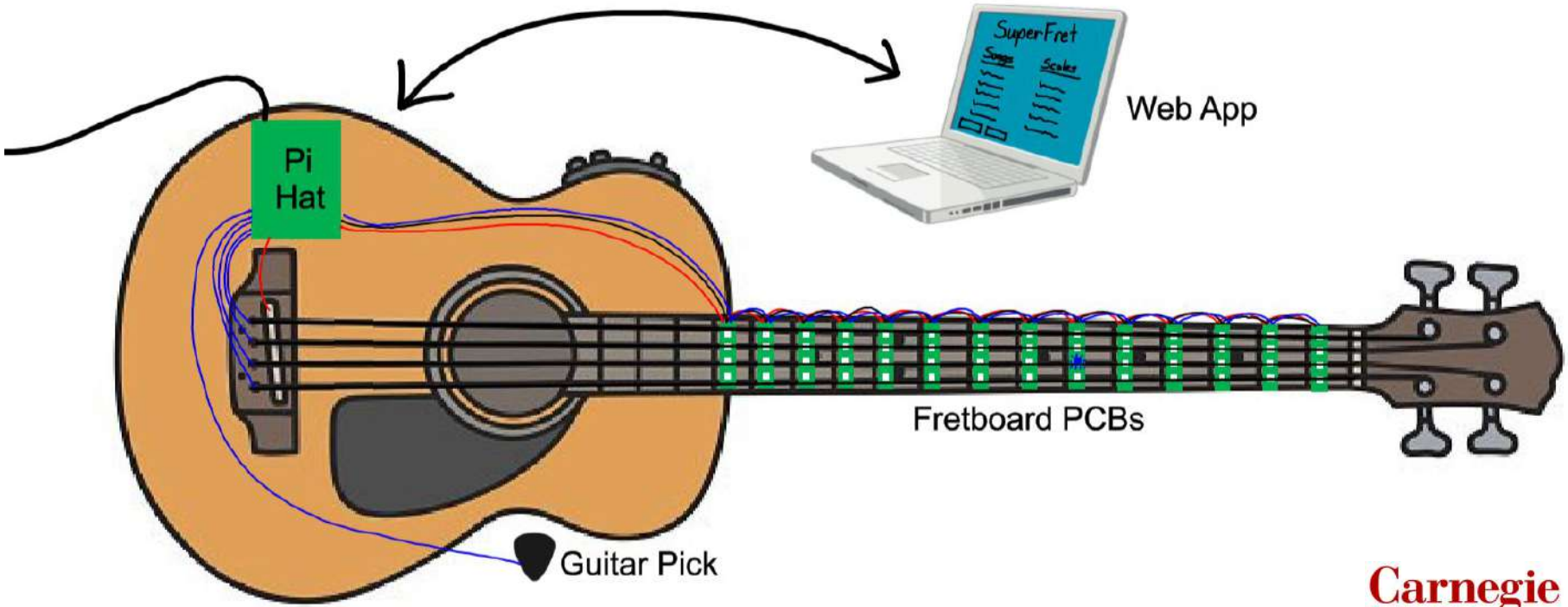
Functional Requirements

- ‘Training’ and ‘Performance’ modes
- ≥14 frets (~2.5 octaves)
- Support 1GB of user’s MIDI files
- Audible Metronome for tempo

User Experience

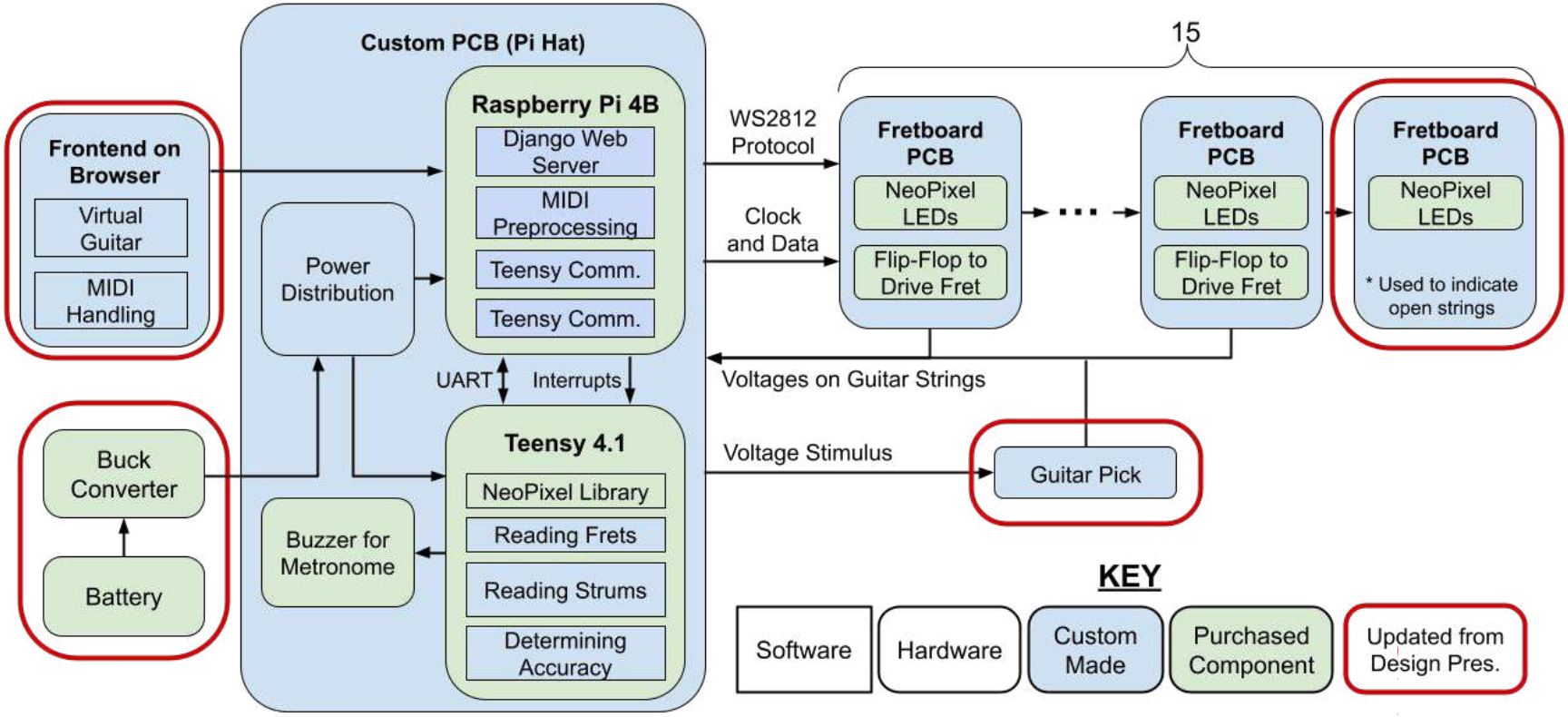
- Intuitive web app (<5 minutes to get acclimated)
- Electronics don’t interfere
- LEDs are visible and effective
- System is responsive

Solution Approach



Carnegie Mellon University

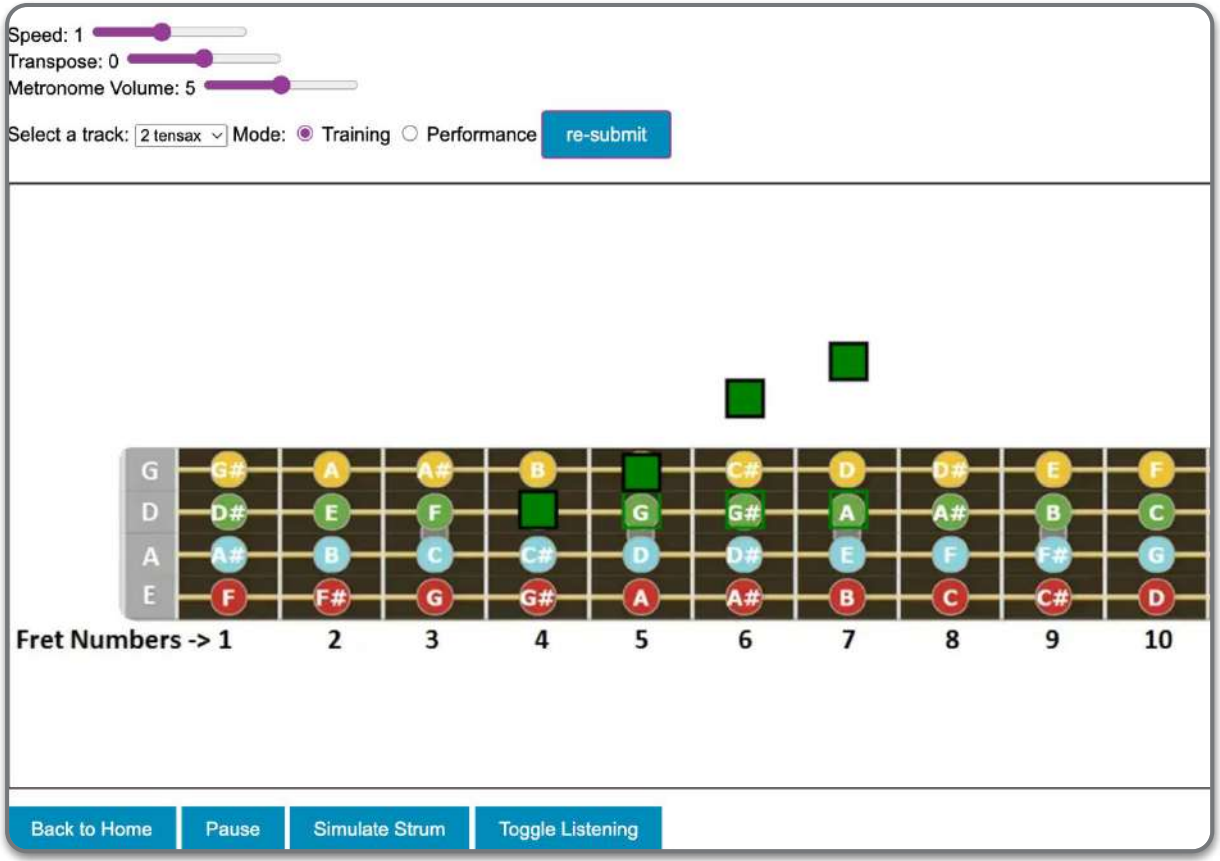
Overall Block Diagram



Public Demonstration Solution

Public Demonstration Solution

- Customize song options
- Fill in the gaps
- Listen to the actual file
- View playing statistics



Testing, Verification, and Validation

Latency	Hardware: Use oscilloscope to measure delay between stimuli, such as time from strumming to LEDs being updated Webapp: Measure one-way latency using high frame-rate video
Accuracy	Strums: Play 200 notes at various BPMs on each string and record % correct Finger Placement: Place a finger on each combination of string and fret position, verify correct LED underneath is illuminated LEDs: Load various MIDI files on guitar and verify that the proper notes are illuminated
User Experience	Have users evaluate categories on scales from 1 to 10 to create a quantitative metric Webapp: Intuitive interface, easy to read statistics, intuitive uploading of songs, etc Hardware: Comfort, effectiveness of LEDs, volume and pitch of metronome

Design 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 ½ brightness	<4.5A	**0.96A**

Use Case Testing

Website:

- How intuitive is the interface? - 8.5
- How responsive is the website? - 10
- How aesthetic is the website? - 7

Guitar:

- How intrusive are our modifications? - 10
- How visible are the LEDs? - 10
- How Responsive is the guitar? - 9

How cohesive is the entire experience? - 10

Meaningful Comments:

- Some note placement timing can be tricky
- Power adapter is bulky

Engineering Tradeoffs

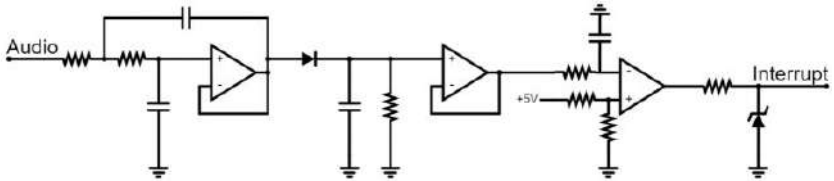
MIDI Parsing

	Parsing on RPi and Teensy	Only parsing on the RPi
Benefits	- Teensy gets *musical* information - Decouple Teensy & RPi development	- Simpler Teensy software - Easier to maintain
Drawbacks	- Complex Teensy software - Code duplication	- Teensy doesn't get *musical* information - Couples Teensy & RPi development

Engineering Tradeoffs

Audio/Piezoelectric vs Pick-Based Strum Detection

	Audio/Piezoelectric	Electrode on Pick
Benefits	- Non-intrusive - Can play without pick	- Extremely fast response time (~1.5ms) - Immune to external noise and vibrations - Allows detection of which string strummed
Drawbacks	- Sensitive to external noise - Slow response time (~50ms) - Complex to calibrate - Relatively inaccurate (~90%)	- Requires playing with a custom guitar pick - More intrusive to user



Tested solution for audio-based detection

Pick with metal electrode

Carnegie Mellon University

Project Management