

Guitar Buddy

Team 15: Austin Born and Christopher Horn

ECE 445, Fall 2018



Presentation Overview

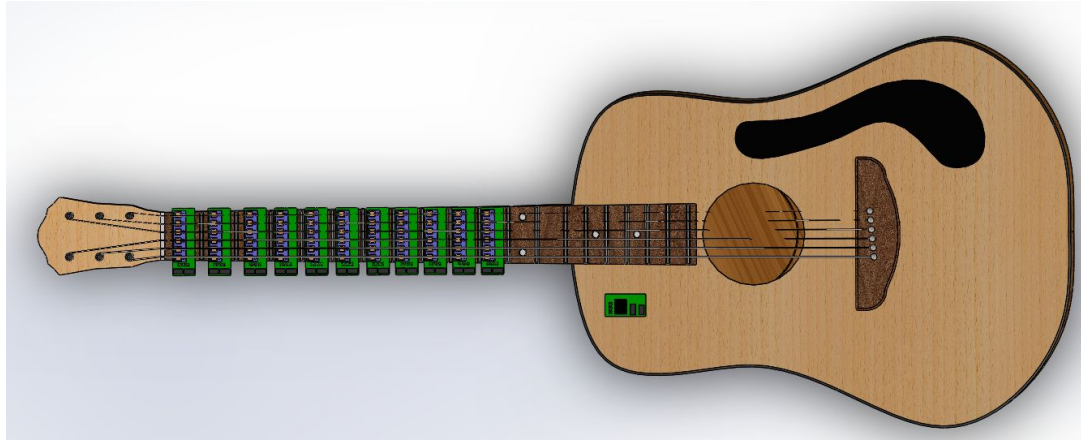
- | | | | | | |
|---|---|---------------------|----|---|--------------------------|
| 3 | - | Project Origins | 11 | - | Software Module |
| 4 | - | Project Objective | 13 | - | LED Output Module |
| 5 | - | Guitar Buddy System | 14 | - | Power Module |
| 7 | - | System Overview | 16 | - | Sensing Module |
| 8 | - | Block Diagram | 18 | - | Successes and Challenges |
| 9 | - | Control Module | 19 | - | Future Work |

Project Origins

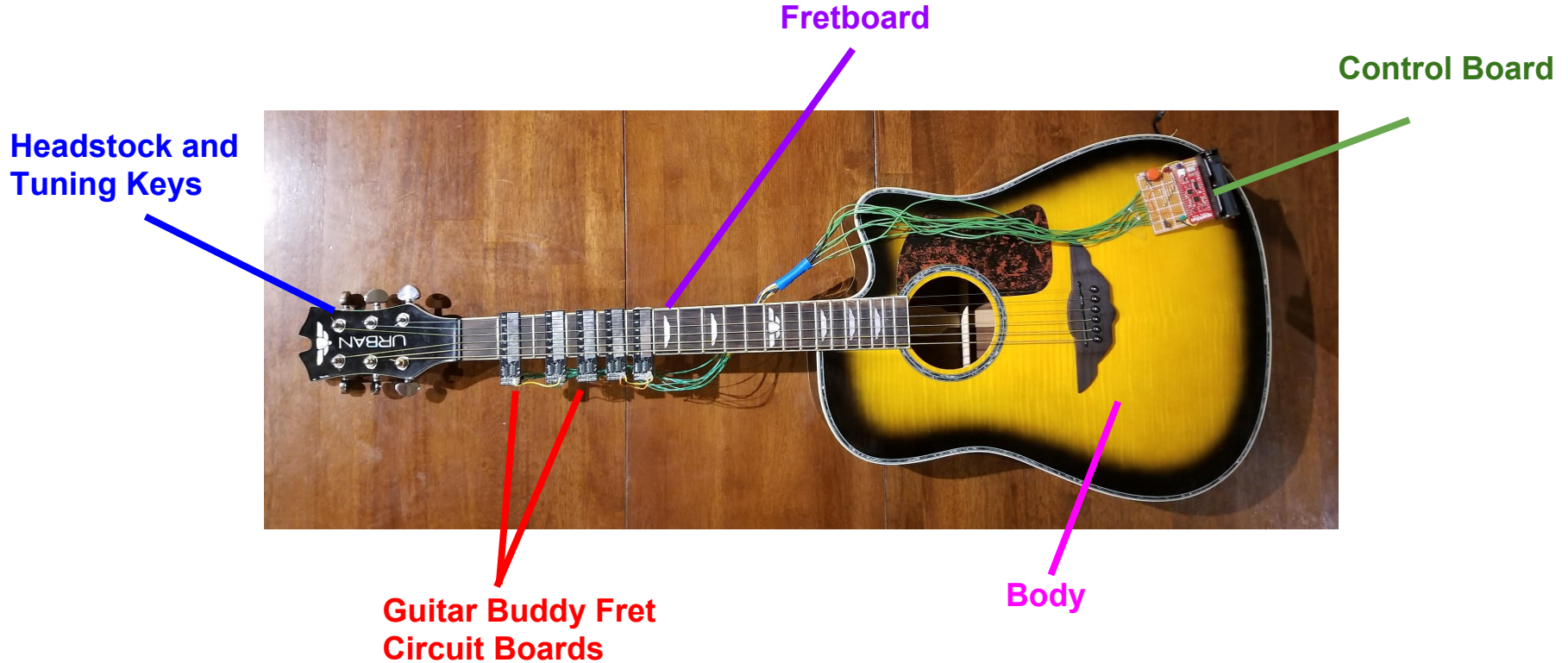
- Music lessons - \$30-50/hr.
- Guitar provides little feedback for self-learners
- Desire for visual cues for notes

Project Objectives

- Display notes/chords in near-real time
- Bluetooth wireless communication
- No (little) interference during normal use



Guitar Buddy System



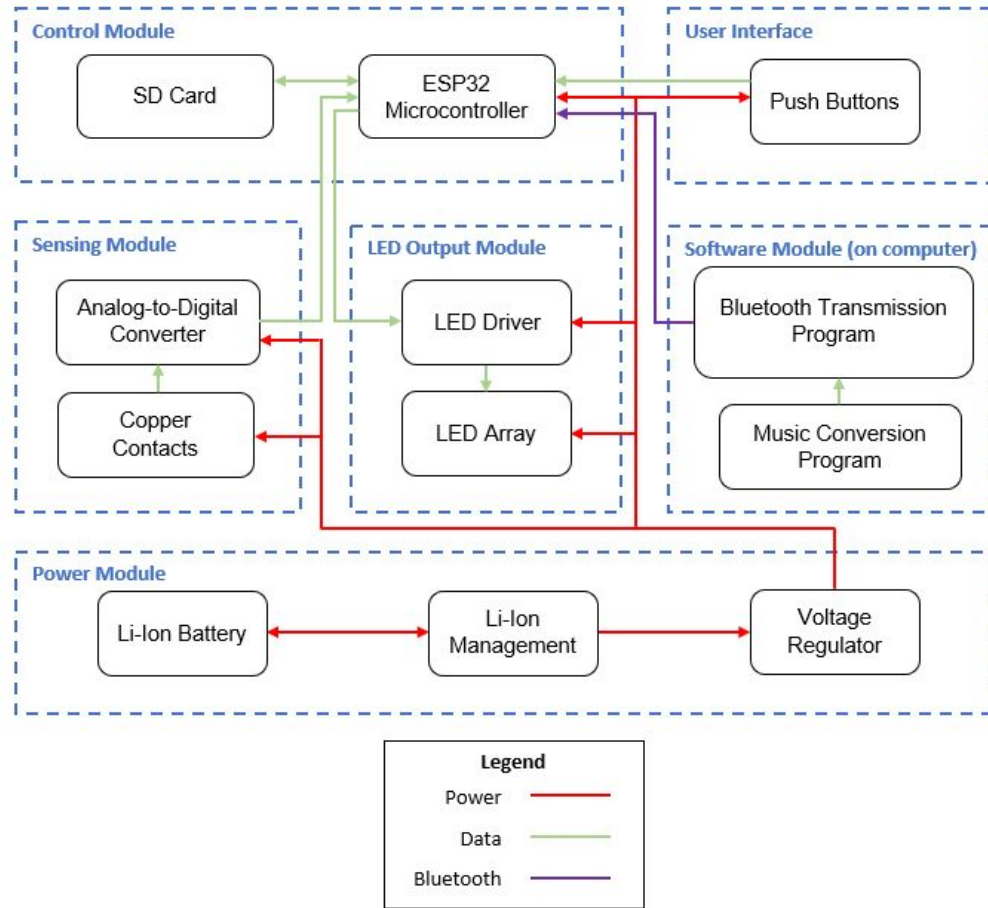
Guitar Buddy System (cont.)

	Strings						
Frets		E2	A2	D3	G3	B3	E4
	1	F2	B ^b 2	E ^b 3	A ^b 3	C4	F4
	2	G ^b 2	B2	E3	A3	D ^b 4	G ^b 4
	3	G2	C3	F3	B ^b 3	D4	G4
	4	A ^b 2	D ^b 3	G ^b 3	B3	E ^b 4	A ^b 4
	5	A2	D3	G3	C4	E4	A4
	6	B ^b 2	E ^b 3	A ^b 3	D ^b 4	F4	B ^b 4
	7	B2	E3	A3	D4	G ^b 4	B4
	8	C3	F3	B ^b 3	E ^b 4	G4	C5
	9	D ^b 3	G ^b 3	B3	E4	A ^b 4	D ^b 5
	10	D3	G3	C4	F4	A4	D5

System Overview

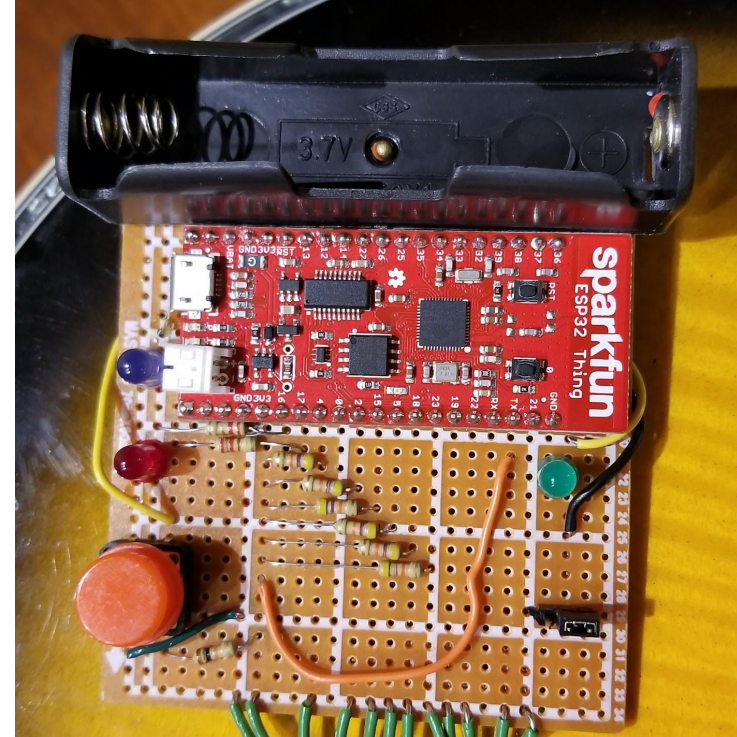
- Control module
 - Generate control signals and manage data
- Software module
 - Extract notes from MIDI files
- LED module
 - Serial interface to drive LEDs
- Power module
 - Provide safe, portable power
- Sensing module
 - Track user's playing and identify errors

Block Diagram

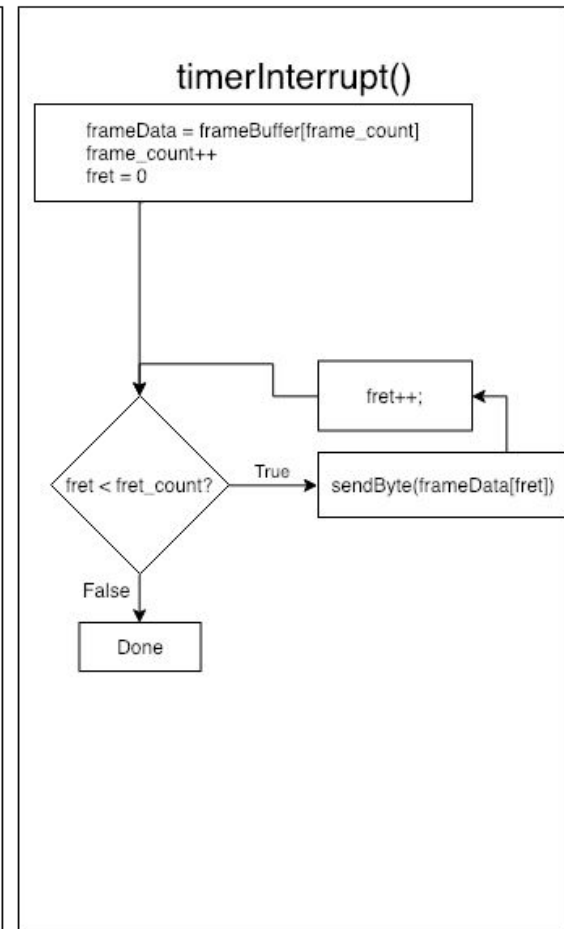
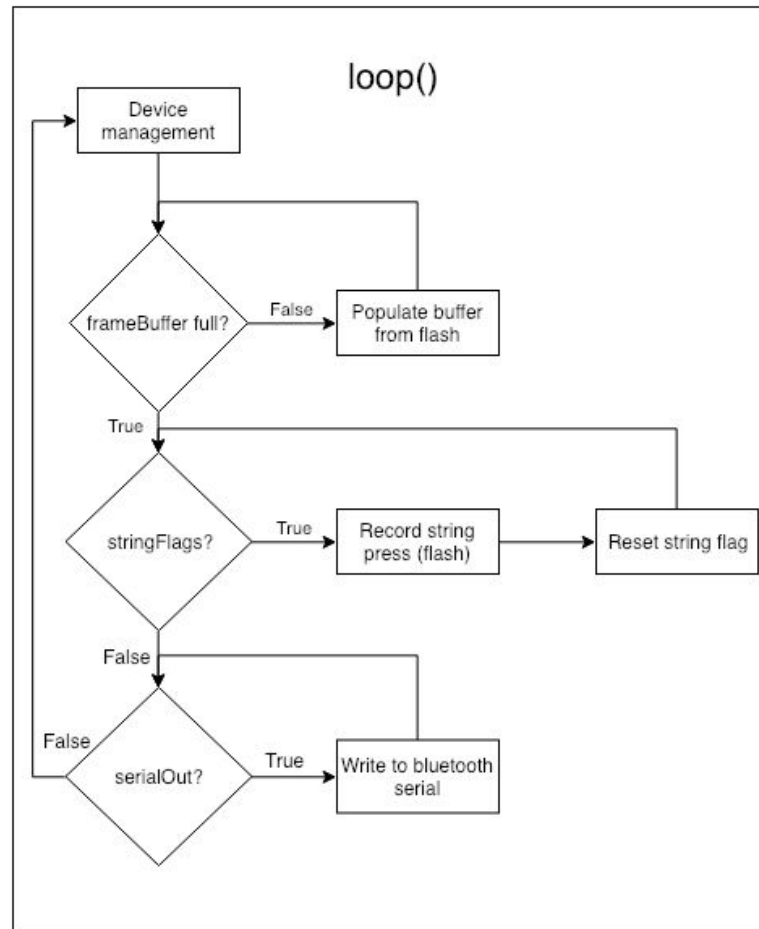


Control Module

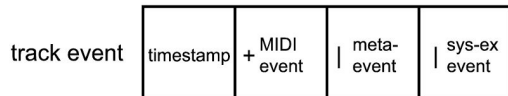
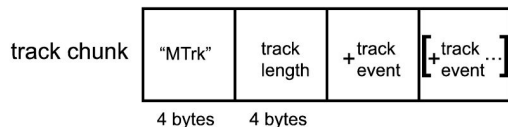
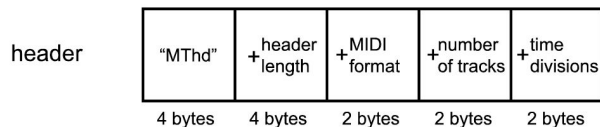
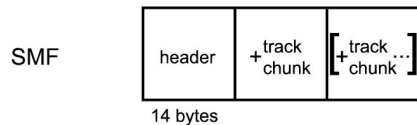
- Control LEDs
- Manage reading string data bus
- Bluetooth communication
- Save song and user data
- Interface with user



Control Software Flowchart



Software module (Parsing MIDI data)



KEY:
This diagram uses regular expression notation.
+ means "and"
| means "or"
[+ x ...] indicates 0 or more repetitions of x.

- Track and time info in header
- Musical notes in track events
 - Timestamp
 - Status byte
 - 0x8X: note off
 - 0x9X: note on
 - Velocity

Software module (Recording LED data)

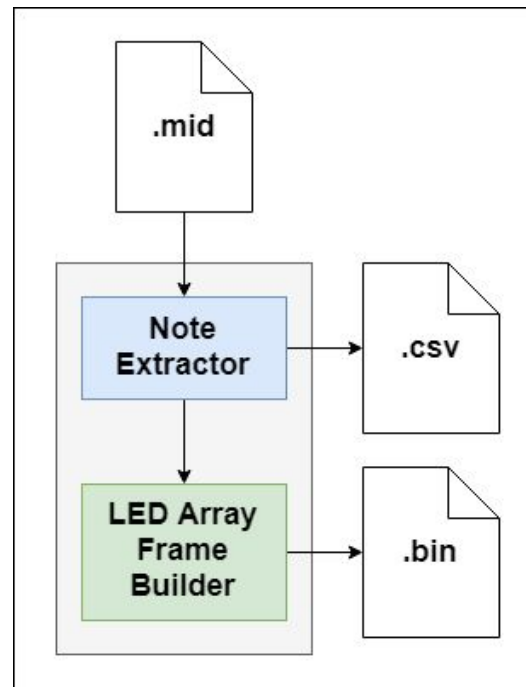
- Takes in MIDI file, outputs CSV and binary files
- Binary file then sent wirelessly to controller

$$\text{Binary File Size (bytes)} = t \times r \times f$$

t = song duration (s)

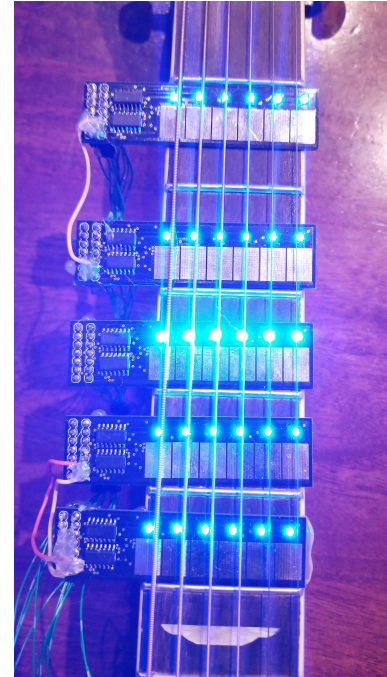
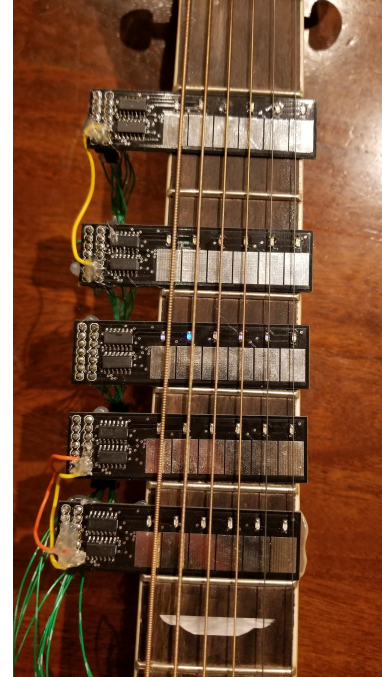
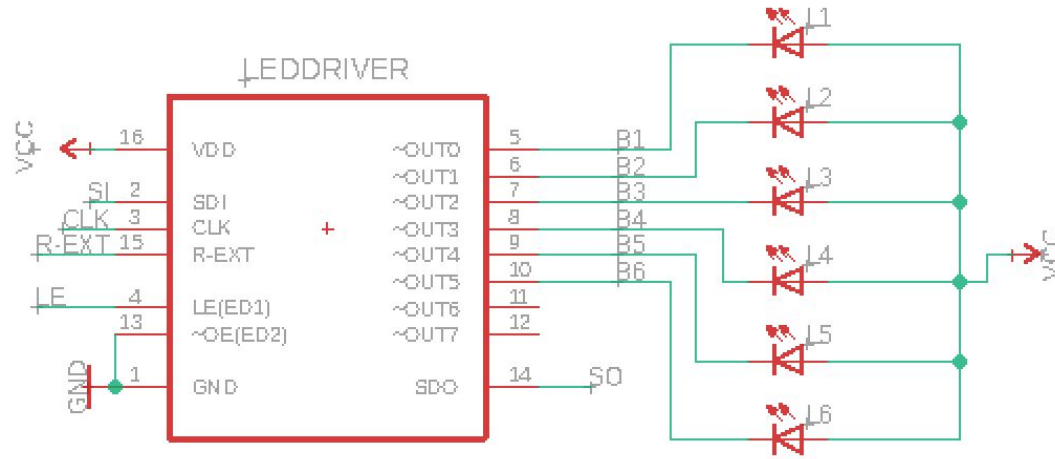
r = refresh rate (Hz)

f = # of fret PCBs



LED output module

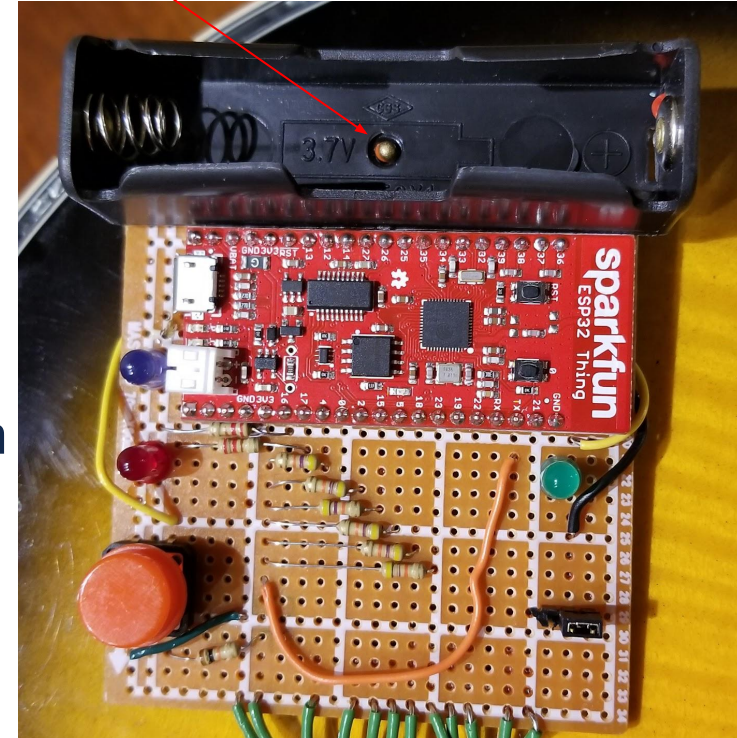
- Constant current shift registers
- SMD LEDs
- Several kHz refresh rate



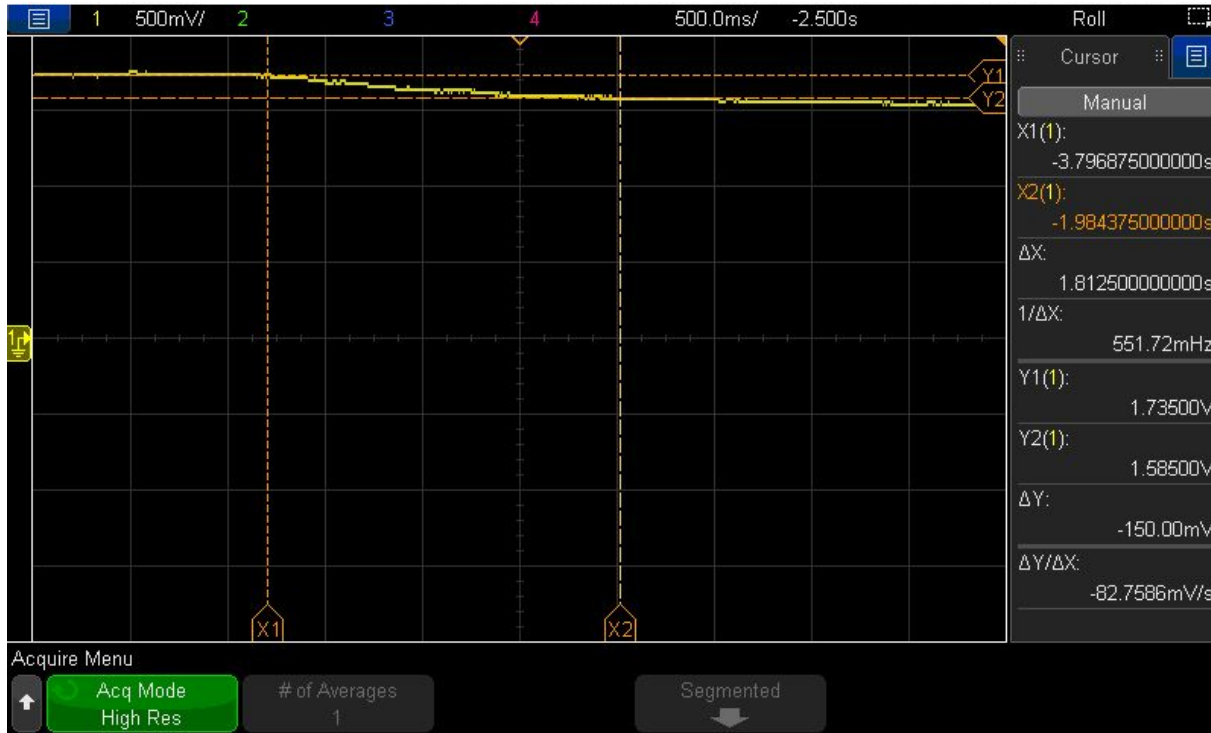
Power Module

- Wired or battery powered
- Removable 18650 battery
 - Provides > 1500mAh capacity
 - Compact and safe
- Overheating protection
 - Protect user in event of malfunction

Thermistor



Verifying Thermal Cutoff



$$R_{therm_{RT}} = 10512 \Omega$$

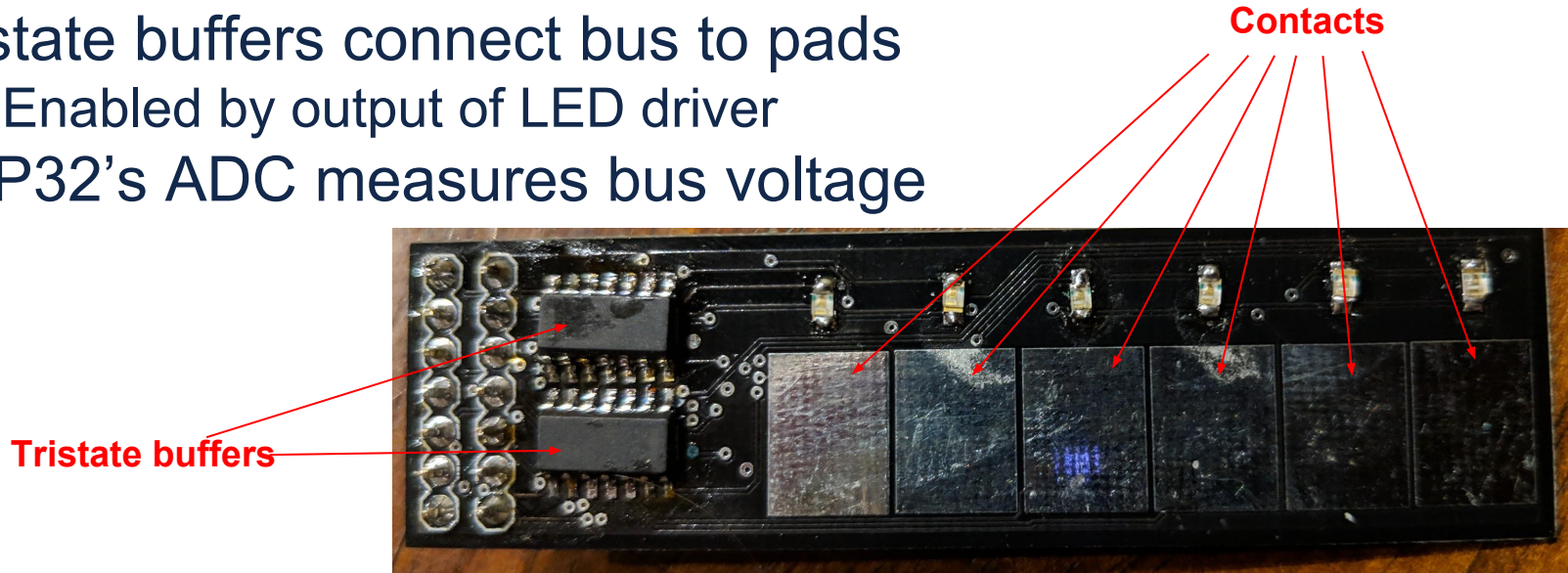
$$R_{cutoff} = 8850 \Omega$$

$$R_{pullup} = 9754 \Omega$$

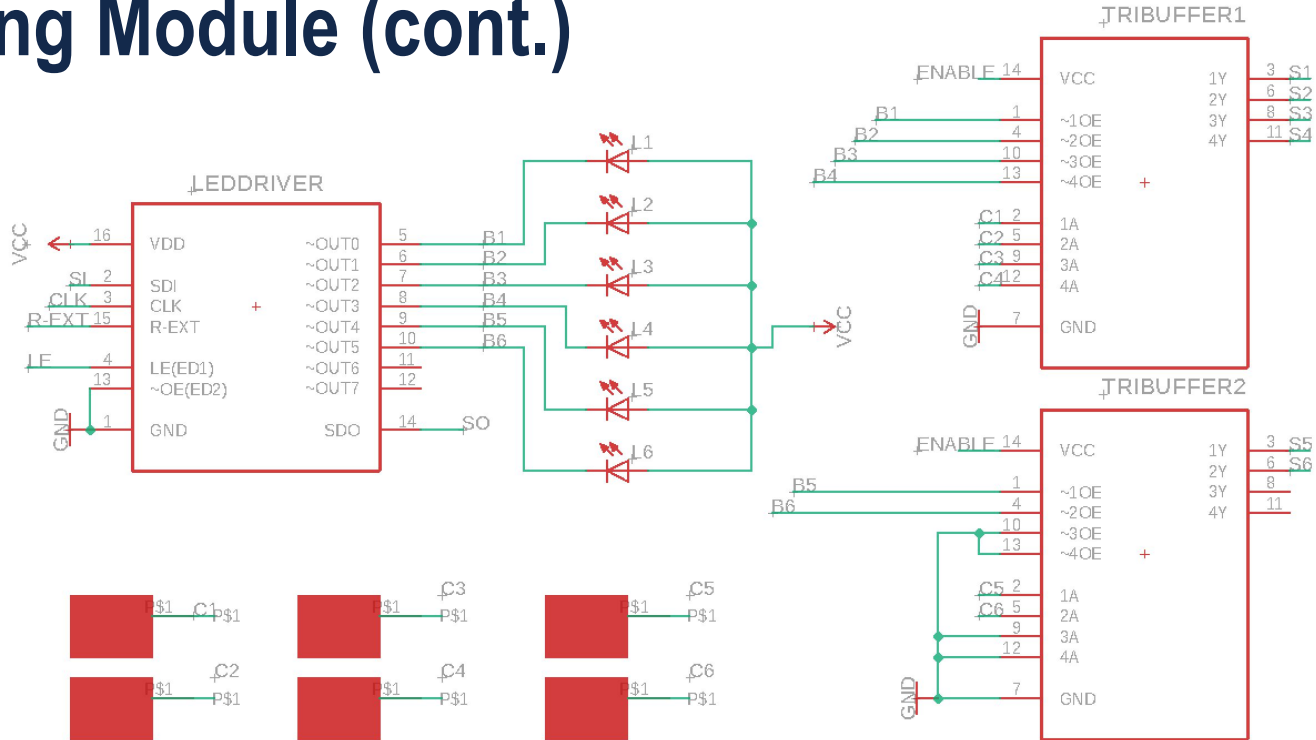
$$V_{cutoff} = \frac{R_{cutoff}}{R_{pullup} + R_{cutoff}} \approx 1.58V$$

Sensing Module

- Guitar strings pulled to GND
- String data bus pulled to 3.3V
- Tristate buffers connect bus to pads
 - Enabled by output of LED driver
- ESP32's ADC measures bus voltage



Sensing Module (cont.)



Successes and Challenges

Successes

- LED array control
- Wireless communications
- Note mapping from MIDI data
- Fitting device onto guitar
- Sensing of strings on individual boards

Challenges

- Saving song data in contiguous memory
- Tristate buffer enable from LED drivers

Future Work

- Representation for 0th fret
- Different teaching modes (practice mode, arcade mode, chord progressions, etc.)
- On-board speaker
- Less invasive fret PCBs and wiring

Questions?

