LEDzeppelin

SDP22 Team 25 | CDR

Michael Forbes, Gavin Baril, Stephen Thimothe, & Sebastian Harder

Team 25 – Senior Design Project 2022

Stephen Thimothe

Computer Engineering

Sebastian Harder

Computer Engineering

Gavin Baril

Electrical Engineering

Michael Forbes

Electrical Engineering

Advisor:
Professor Daniel Holcomb

Team Responsibilities

Stephen Thimothe

Embedded Software Lead

Component Programming

Sebastian Harder

Software Dev Lead

GUI Programming & Design

Gavin Baril

Hardware Lead

Component Design

Michael Forbes

Signal Processing Lead

Team Coordinator

The Idea

- To create a lighting system to be used as a live performance aid for solo guitarists.
- The lighting is on the guitar and responds to the live audio signal created by the guitarist.

LEDzeppelin is a performance element.

1. The signal will be retrieved from the wired audio signal of the guitar.
2. LEDzeppelin will be primarily a performance element for solo guitarists and small bands.

Problem Statement

Independent guitarists are often looking for ways to make their sets more interactive and unique. However, as artists at small gigs, their options are limited by finances, space, and portability as well as limited in flexibility and customizability.

LEDzeppelin will be piece of equipment that solo artists or small bands can use to create a customizable lighting experience for live performances. The product will be portable, scalable, and unintrusive as to not interfere with the artists' playing style.

System Specifications & Testing Plan

- The system has no more than 20ms latency.*
- An advanced knowledge of programming is not necessary to operate LEDzeppelin.
  - A DMX controller allows for easy adjustments mid-performance.
  - The system is compatible with any standard electric guitar.
  - Impermanent, and easy to install.
    - FretZealot installation video
- The LEDs respond to a variety of signal parameters, including:
  - amplitude
  - frequency
- The refresh rate of the fret board will be tested with an accurate camera and a known shutter speed.
- Focus groups will be used in two ways:
  - To test audience reactions to both individual configurations and an entire performance.
  - To test guitarists' experience using the system for the first time.
- Signal control testing:
  - Amplitude detection accuracy will be tested by strumming the guitar at various volumes.
  - Frequency detection accuracy can be tested with an oscilloscope for chords.
  - Open string tuning should be at a known fundamental frequency.

** For this project, latency is defined as the time between the creation of the signal (when the guitar is strummed), and the visual response of the LEDs.*

Notes on Latency

The latency specifications for the system are based on the distance of the audience and the musician from the sound source, and the average musician's threshold for such latency (about 12-20ms).

For example, a musician standing 1 meter from the amplifier experiences ~3ms of latency.

The audience, however, in a small venue is likely ~4-5 meters away.

Also, note that while individual note detection results in significant latency, a tuning function has room for this.

System Specifications & Testing Plan (continued)

✓	• LEDs are assigned to each string on each fret, at least down to the body of the guitar.	• Tested by inspection. | *Satisfied*

✓	• The system is portable.
	  • No more than one person is required to set it up.
	  • The entire system (not including the guitar) is no more than 5 lbs.
		• Portability testing:
		  • Total system weight will be measured.
		  • Setup/breakdown time will be recorded.

✗	• There are LED configurations that users can control and assign.
	  • One of the preset functions will be a built-in tuner.
	  • Other configuration settings for performing will include color pickers, mode selection, and more.
	  • Example: Preset #3 sets the LEDs to change colors depending on the frequency of audio.
	  • More details on next slide.
		• Tested by inspection.
		  • Verify tuner is accurately narrowing suggestions visually toward a consistent note or frequency.
		  • Test configurations by inspection, among other methods. More details on next slide.

Configuration Specifications & Testing Plan

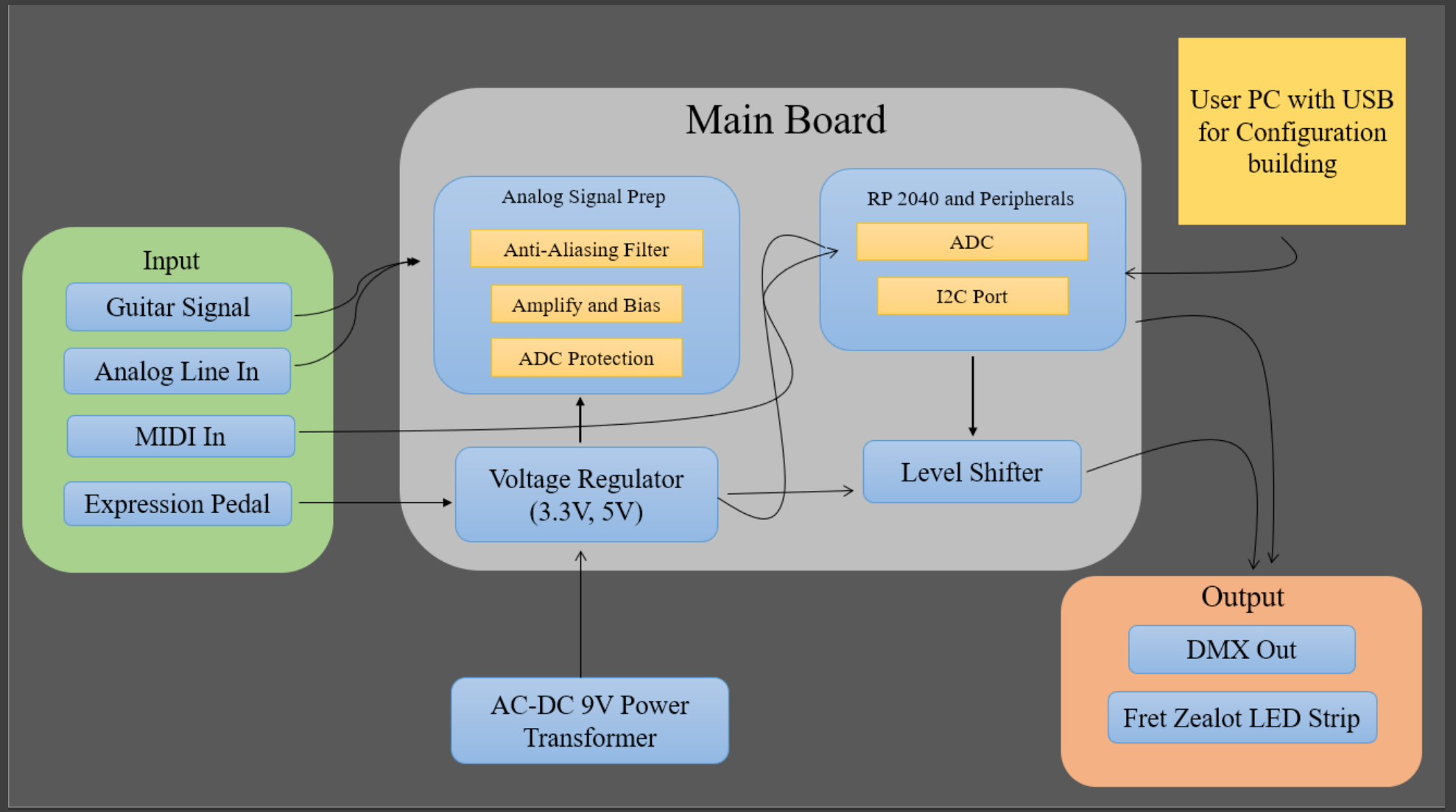
Using the GUI, the user can set the following effects to be implemented:

- ✓ Change LED colors, with the option of setting specific hex color codes.
  - We will test the color accuracy using a spectrometer.
- ✗ Audio visualizer based on volume of playing / amplitude of signal.
  - Test by inspection, verifying that LED output changes properly for different volumes played.
- ✗ Audio visualizer based on note played / frequency of signal.
  - Test by inspection, verifying that LED output is consistent for a specific note strummed.
- ✗ Control any part (all or some) of the LED array with chosen signal elements.
  - Test by inspection, checking that GUI settings for turning off sections of the LEDs work.
- ✗ Stacking multiple functions in a single configuration.
  - Example: a configuration that scales the neck of the guitar with volume while simultaneously changing LED hue by frequency.
  - Test by inspection, combining the applicable aforementioned tests for the various stacked outputs.

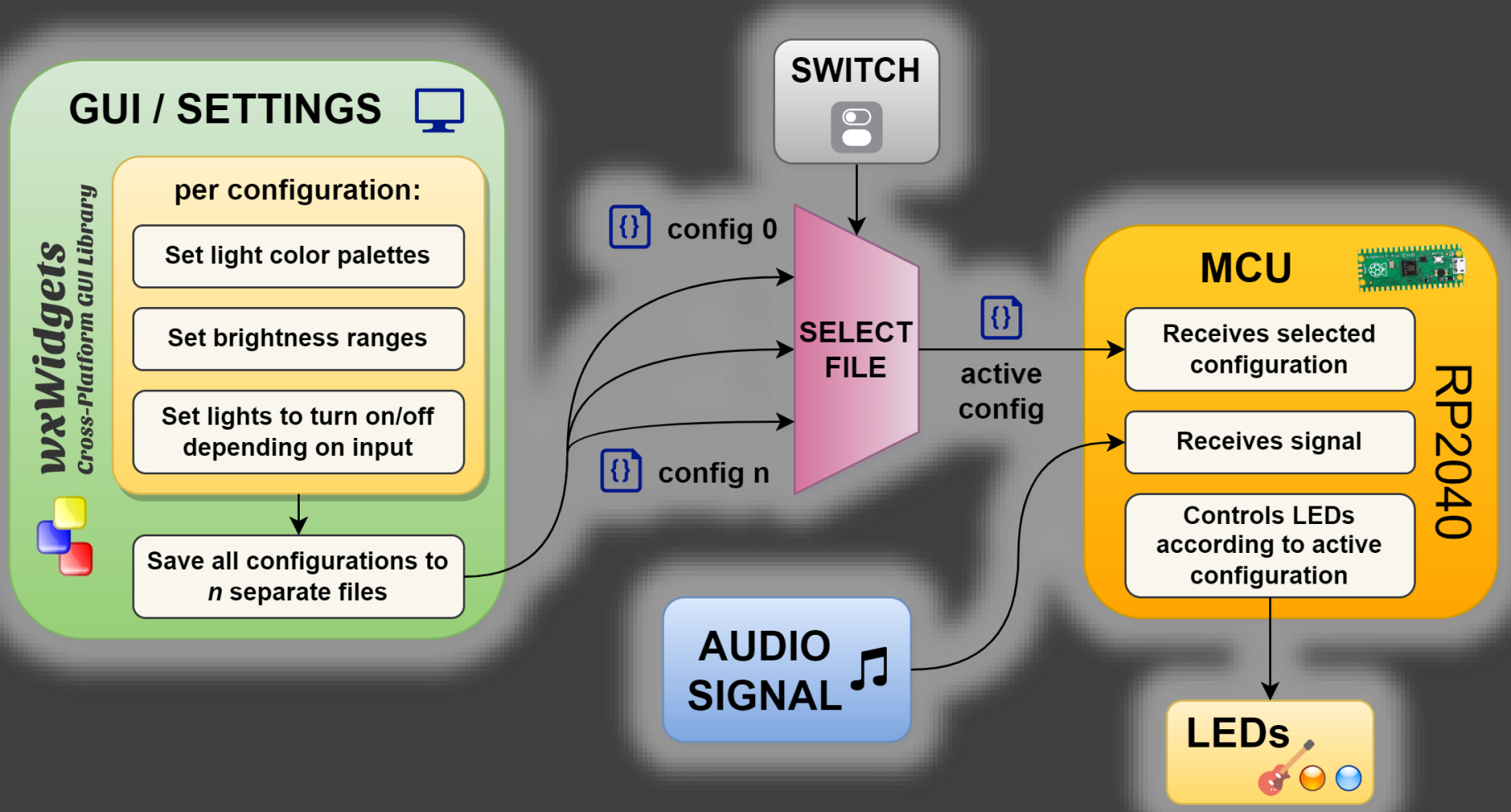
Hardware and Software Components

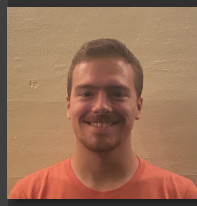
- Hardware
  - Raspberry Pi Pico (RP2040)
  - Fret Zealot LED strip
  - 2N7000 BJT for level shifter
  - LED and Barrel jack splitters.
- Software
  - wxWidgets (GUI framework)
  - C++ programming in Visual Studio, Visual Studio Code

Hardware Block Diagram



Software Block Diagram





The following were our proposed deliverables for CDR:

- ✗ The system is fully integrated on a printed circuit board.

- ✗ Demonstration of intuitive guitar tuning configuration.

-  Demonstration of frequency dependent performance configurations.

-  The GUI fully manipulates necessary configuration parameters.

Frequency Detection (currently)

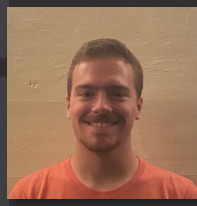
- Using KissFFT library.
- Sampling at 5KHz, 5000 point FFT (1Hz frequency resolution.
- Some issue with double frequencies when detecting strings.
  - Plan to use Harmonic Addition to detect string fundamentals.

Open Guitar String Fundamentals

String	Frequency	Scientific pitch notation
1 (E)	329.63 Hz	E_4
2 (B)	246.94 Hz	B_3
3 (G)	196.00 Hz	G_3
4 (D)	146.83 Hz	D_3
5 (A)	110.00 Hz	A_2
6 (E)	82.41 Hz	E_2

GUI Demo

```
30      bSizer_main->Add(choice_configs, 0, wxALIGN_CENTER|wxALL, 15);
31
32      // COLOR PICKER //
33
34      gSizer_colors = new wxGridSizer(0, 4, 0, 0);
35
36      button_color1 = new wxButton(this, wxID_ANY, "Red", wxDefaultPosition, wxDefaultSize, 0);
37      button_color1->SetBackgroundColour(wxColour(255, 0, 0));
38      button_color1->SetToolTip("#FF0000");
39      gSizer_colors->Add(button_color1, 0, wxALL, 5);
40
41      button_color2 = new wxButton(this, wxID_ANY, "Orange", wxDefaultPosition, wxDefaultSize, 0);
42      button_color2->SetBackgroundColour(wxColour(255, 165, 0));
43      button_color2->SetToolTip("#FFA500");
44      gSizer_colors->Add(button_color2, 0, wxALL, 5);
45
46      button_color3 = new wxButton(this, wxID_ANY, "Yellow", wxDefaultPosition, wxDefaultSize, 0);
47      button_color3->SetBackgroundColour(wxColour(255, 255, 0));
48      button_color3->SetToolTip("#FFFF00");
49      gSizer_colors->Add(button_color3, 0, wxALL, 5);
50
51      button_color4 = new wxButton(this, wxID_ANY, "Green", wxDefaultPosition, wxDefaultSize, 0);
52      button_color4->SetBackgroundColour(wxColour(0, 255, 0));
53      button_color4->SetToolTip("#00FF00");
54      gSizer_colors->Add(button_color4, 0, wxALL, 5);
55
56      button_color5 = new wxButton(this, wxID_ANY, "Blue", wxDefaultPosition, wxDefaultSize, 0);
57      button_color5->SetBackgroundColour(wxColour(0, 128, 255));
58      button_color5->SetToolTip("#0080FF");
59      gSizer_colors->Add(button_color5, 0, wxALL, 5);
60
61      button_color6 = new wxButton(this, wxID_ANY, "Purple", wxDefaultPosition, wxDefaultSize, 0);
62      button_color6->SetBackgroundColour(wxColour(128, 0, 255));
```



Budget

Item	Quantity	Price
Guitar	1	Already owned
Fret Zealot Strip	1	$100
PCBs	5	$50
Hardware (MCUs, equalizer)	~	$60
9V Power Supply Unit	1	Already owned
Total	~	$210

Proposed FPR Deliverables

- The system is fully integrated on a printed circuit board.
- Demonstration of frequency and amplitude dependent performance and tuning configurations.
- The GUI fully manipulates all implemented configuration parameters.
- System specifications all successfully tested using test plans and criteria is met.

The path to FPR: a Gantt chart

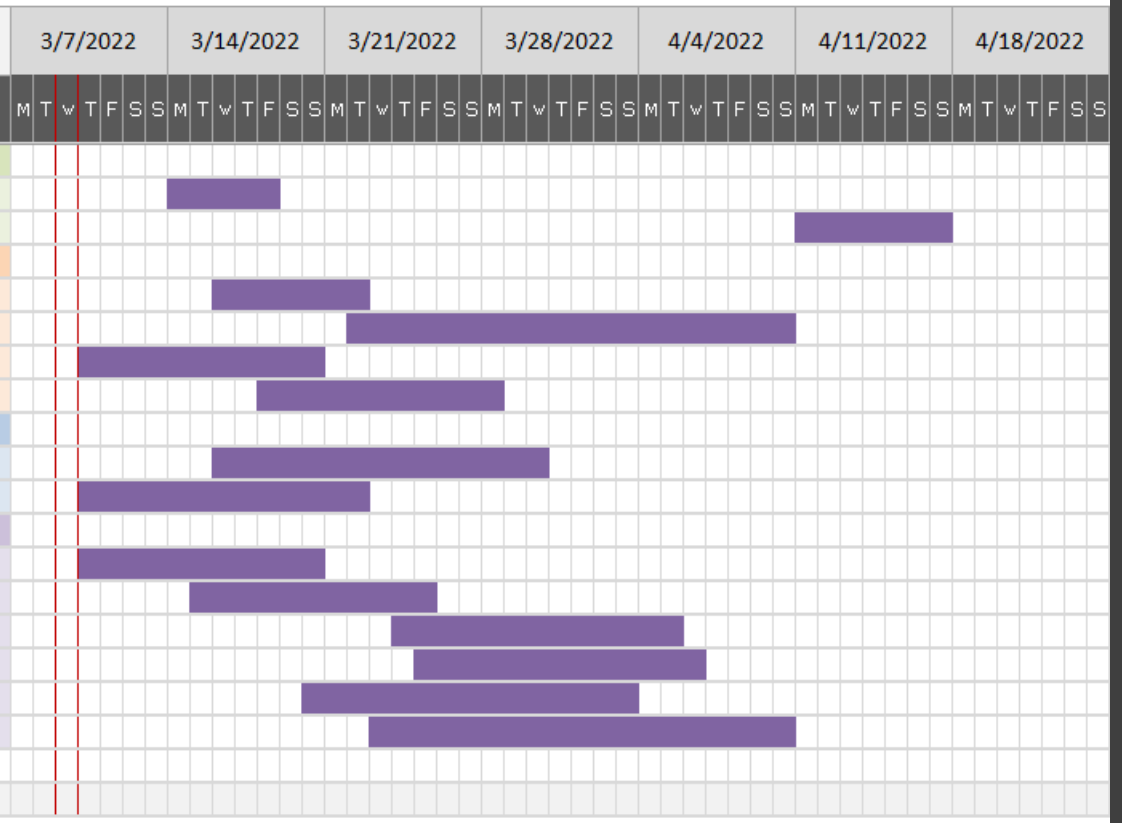
LEDzeppelin

Team 25

Sebastian Harder, Gavin Baril,
Stephen Thimothe, Michael Forbes

Start date: Wed, 3/9/2022

TASK	ASSIGNED TO	PROGRESS	START	END
Administrative Tasks				
Populate custom PCB	All	0%	3/14/22	3/18/22
FPR presentation preparation	All	0%	4/11/22	4/17/22
Hardware				
Populate custom PCB	Gavin/Michael	0%	3/16/22	3/22/22
Get functions operating on PCB	All	0%	3/22/22	4/10/22
Get the LEDs back online	All	0%	3/10/22	3/20/22
Optimize hardware performance	Gavin	0%	3/18/22	3/28/22
Signal Processing				
Frequency-dependent configurations	Michael/Stephen	0%	3/16/22	3/30/22
Tuning function	Michael/Stephen	0%	3/10/22	3/22/22
Software				
Make GUI executable outside of IDE	Sebastian	0%	3/10/22	3/20/22
Output GUI config files directly to MCU	Sebastian	0%	3/15/22	3/25/22
Add LED brightness/amplitude settings	Sebastian	0%	3/24/22	4/5/22
Add frequency settings	Sebastian/Stephen	0%	3/25/22	4/6/22
Tuner settings	Sebastian/Stephen	0%	3/20/22	4/3/22
Improve & expand GUI design elements	Sebastian	0%	3/23/22	4/10/22
Insert new rows ABOVE this one				



Questions?