LEDzeppelin

SDP22 Team 25 | MDR

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

Senior Design 2022 – Team 25

Stephen Thimothe
Computer Engineering

Sebastian Harder
Computer Engineering

Gavin Baril
Electrical Engineering

Michael Forbes
Electrical Engineering

Advisor:
Professor Daniel Holcomb

Team Responsibilities

- Michael Forbes
  - Hardware / signal processing methods / team coordinator
- Gavin Baril
  - Hardware / embedded design
- Stephen Thimothe
  - Programming embedded components
- Sebastian Harder
  - Software / GUI program design

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

- 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.
- The LEDs respond to a variety of signal parameters, including:
  - amplitude
  - frequency

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

System Specifications

- LEDs are assigned to each string on each fret, at least down to the body of the guitar.
- The system is portable.
  - It does not require more than one person to setup.
  - The entire system (not including the guitar) is no more than 5 lbs.
- There are LED configurations that users can control and assign.
  - Example: Preset #3 causes the LEDs to respond to low-frequency audio and change colors depending on the frequency.
  - Configuration settings will include color pickers, mode selection (between frequency response, amplitude response, etc.), and more.
  - One of the preset functions will be a built-in tuner.

System Specifications

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.

Configuration Specifications

Using the GUI, the user can implement the following effects:

- Changing colors that have specific hex codes.
- Audio visualizer based on volume of signal.
- Control any part of the LED array based on chosen signal elements.
- Stacking multiple functions in a single configuration.
  - For example: a configuration that scales the neck of the guitar with volume while simultaneously changing LED hue by frequency.

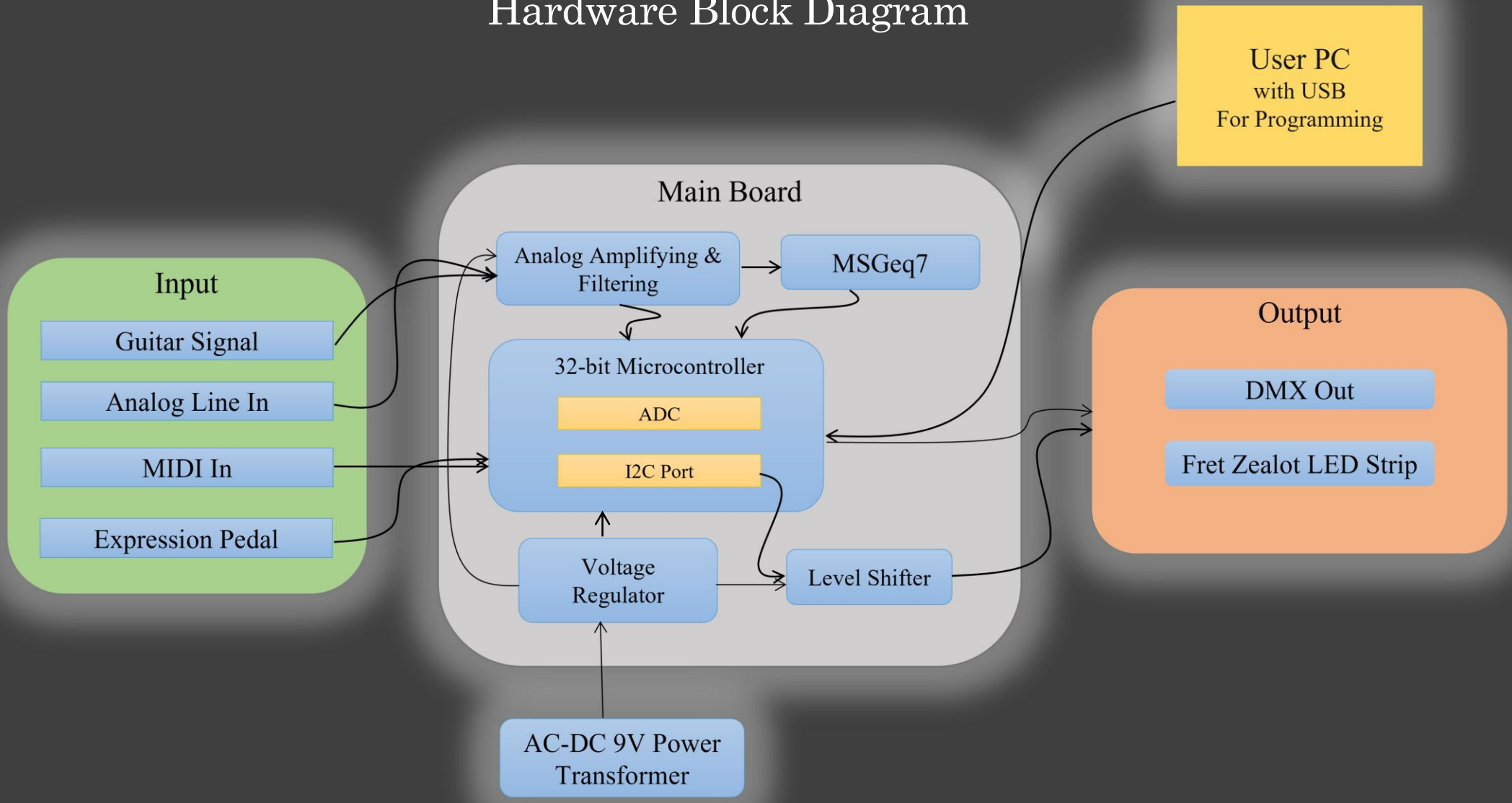
Test Plan

- 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.
- We will test the color accuracy using a spectrometer.
- The refresh rate of the fret board will be tested with an accurate camera and a known shutter speed.
- Frequency detection accuracy can be tested with an oscilloscope for chords.
  - Open string tuning should be at a known fundamental frequency.

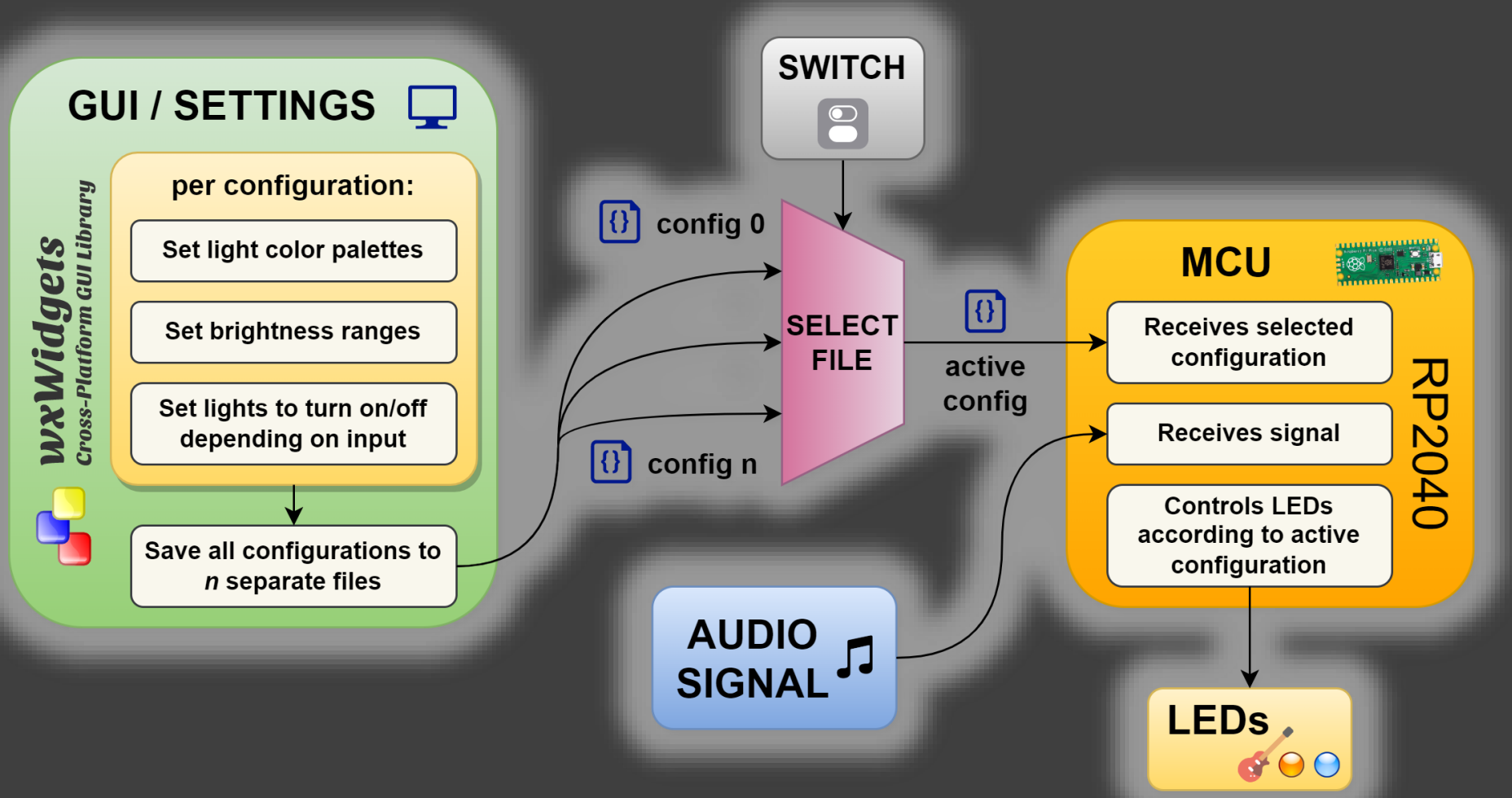
Hardware and Software Components

- Hardware
  - Raspberry Pi Pico (RP2040)
  - Fret Zealot LED strip
  - 2N7000 BJT for level shifter
  - LED and Barrel jack splitters.
- Software
  - wxWidgets (GUI) & C++ programming
  - Raspberry Pi Pico C libraries

Hardware Block Diagram



Software Block Diagram



The following were our proposed deliverables for MDR:

- The ability to retrieve signal parameters
  - Demonstration of amplitude dependent mode.
- Full control of the fretboard LEDs
- Preliminary GUI adjusts aspects of the LED output
- The ability to switch between preset ‘modes’

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));
```

The following were our proposed deliverables for MDR:

- ☑ The ability to retrieve signal parameters
  - Demonstration of amplitude dependent mode.
- ☑ Full control of the fretboard LEDs
- ☑ Preliminary GUI adjusts aspects of the LED output
- ☑ The ability to switch between preset ‘modes’

Budget

Item	Quantity	Price
Guitar	1	$0, Already owned
Fret Zealot Strip	1	$100
PCBs	5	$50
Hardware (MCUs, equalizer)	~	$60
9V Power Supply Unit	1	$0
Total	~	$210

Proposed CDR Deliverables

- 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.

The path to CDR: a Gantt chart

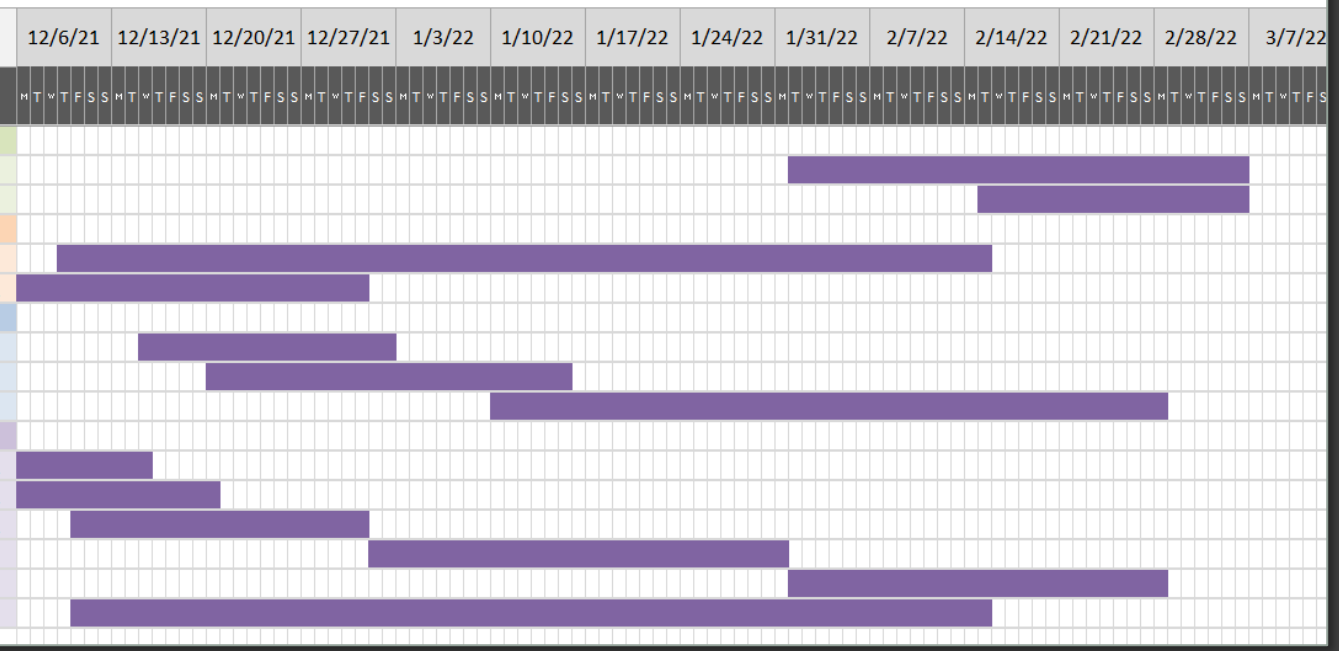
LEDzeppelin

Team 25

Sebastian Harder, Gavin Baril,
Stephen Thimothe, Michael Forbes

Start date: Mon, 12/6/2021

TASK	ASSIGNED TO	PROGRESS	START	END
Administrative Tasks				
Order custom PCB	All	0%	2/1/22	3/6/22
CDR presentation preparation	All	0%	2/15/22	3/6/22
Hardware				
Design custom PCB	Gavin/Michael	0%	12/9/21	2/15/22
Optimize hardware performance	Gavin	0%	12/6/21	12/31/21
Signal Processing				
Experiment w/MSGEQ7	Stephen	0%	12/15/21	1/2/22
Frequency-dependent configurations	Michael/Stephen	0%	12/20/21	1/15/22
Tuning function	Michael/Stephen	0%	1/10/22	2/28/22
Software				
Make GUI executable outside of IDE	Sebastian	0%	12/6/21	12/15/21
Output GUI config files directly to MCU	Sebastian	0%	12/6/21	12/20/21
Add LED brightness/amplitude settings	Sebastian	0%	12/10/21	12/31/21
Add frequency settings	Sebastian/Stephen	0%	1/1/22	1/31/22
Tuner settings	Sebastian/Stephen	0%	2/1/22	2/28/22
Improve & expand GUI design elements	Sebastian	0%	12/10/21	2/15/22



Questions?