Sound Effect Devices for Musicians

FINAL PROJECT REPORT

Team Number: Dec1712

Client / Advisor:
Dr. Randall Geiger

Team Members:
Jake Asmsus
Joseph Brown
Daniel Peterjohn
Jiangning Xiong

Team Email:
dec1712@iastate.edu

Revised: 12/6/17, Version 1

Contents

1 Introduction 2
1.1 Project statement 2
1.2 Purpose 2
1.3 Goals 2
2 Deliverables 3
3 Design 3
3.1 Related Work/Projects 6
3.2 Proposed System Block diagram 7
3.3 Assessment of Proposed methods 8
3.4 Validation 9
4 Project Requirements/Specifications 10
4.1 Functional 10
4.2 Non-functional 11
4.3 Standards 11
5 Challenges 11
6 Overview 12
6.1 First Semester 12
6.2 Second Semester 13
7 Conclusions 14
8 References 15
Appendix I - Operation Manual 16
Appendix II - Alternate Versions 23

1 Introduction

1.1 Project statement

To create a single pedal board that houses several effects digitally, and allows the user to adjust any feature of the pedals by stepping on them. A series of digital effects and filters can be rearranged and "re-wired" in any imaginable combination to create even more diverse sounds.

1.2 Purpose

Our main purpose for this project is to provide musicians with a pedal board that can accomplish many of the sounds achieved in various pedals, and to give them a better interface to control their sounds. Today's musicians collect several pedals and try to manipulate sound in their own unique way with the limitations of the pedals they buy, but we want to give them more freedom in their sound selection as well controlling them live.

Currently there is no user interface that allows a musician to adjust the individual settings of a pedal in a live setting easily. Normally a musician is focused on the music and does not have the time to adjust the settings manually, usually by hand, which is crazy! Why would we make a musician use their hands as the only method to adjust the settings on a pedal? These sound effect devices are mostly used by musicians for changing the sound of their guitar, bass, or keyboard. All of these instruments require the use of almost both hands at all times to operate, so there is no way to adjust a setting on a pedal in a live performance easily. With our pedalboard interface, we could use foot switches to adjust each setting on a given sound effect that's compatible with our design. This pedal board would free up musicians to adjust their sound to their heart's content while playing their instruments uninhibited.

Using a microcontroller, we communicate the adjustments to the specific sound effect, making adjustments live, seamlessly. Another feature unique to this pedal board is the splitting of the input signal, giving the musician two outputs to utilize. One output could go straight to an amp, while sending the other signal through a series of other pedals to create a unique sound synchronized with the other signals. Several other benefits could be used for this feature, creating more flexibility in a musician's sound.

We are going away with the days of old, where a musician would have to adjust their pedal before a performance and deal with the settings throughout the whole show. Sometimes a musician might kick a pedal's settings on accident or realize too late that his settings don't sound right, well not anymore. With our pedal mat, the performer can quickly adjust any setting on the pedal hands-free!

1.3 Goals

The goals we have are making performers in a more relaxed setting with all their pedals inside the pedalboard, covered, at a single location, free state of mind from hazardous wires, and re-programmable effects and filters for the user to control and customize. In the end, we would like to influence the pedal industry to convert to this standard such that the pedals may be controlled with the pedal itself or a universal interface to be used by all musicians.

2 Deliverables

Our deliverable will be a pedal board that has the functionality of all user's pedals into a single, 2-by-5 button layout interface with the help of a microcontroller. There will be an "on" or "off" state; To aid the user, we will have an LED display to notify the user what state they are in so they can prepare for the next sequence of buttons for another sound effect. The pedal itself will be taking in one input from the user's instrument and then splitting this signal, where one half will be manipulated and the other half will pass unaltered. This allows the user multiple combinations for a wide range of sound adjustments and phenomenon.

3 Design

Below is a summary of the Infinity Board Design.

1. Physical Layout
    a. 2-by-5 button/cell layout
    b. A LCD to provide info on the pedalboard layout
        i. Four buttons to navigate LCD menu
    c. Size: 0.91x0.30 (m)^2
    d. The top platform is inclined by 30 degrees
    e. Button cells
        i. Each cell will have an RGB LED
        ii. Each column (Top cell) will have a two digit 7-segment-display
2. Circuitry
    a. Shift Registers
        i. Serial In - Parallel Out (SIPO)
        ii. Parallel In - Serial Out (PISO)
    b. Pull-up and pull-down networks
    c. Five Two-digit 7-segment-display modules pre-made
3. Controller
    a. Teensy 3.6 microcontroller
    b. Teensy Audio Adapter

The Infinity Board will have three main components: LCD Paging to navigate the pedal mapping, "pedal board" for the effects (side right of the LCD screen in Figure 3), and an audio adaptor to create the audio signals (see Figure 1 for the audio adapter) all controlled/processed by the Teensy 3.6 (see Figure 2). The Infinity Board will have a similar layout in Figure 3.

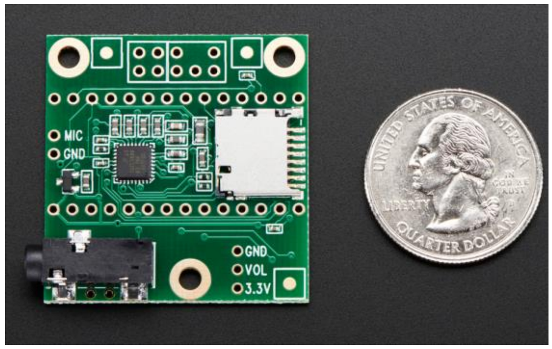


Figure 1: *Audio Adapter*

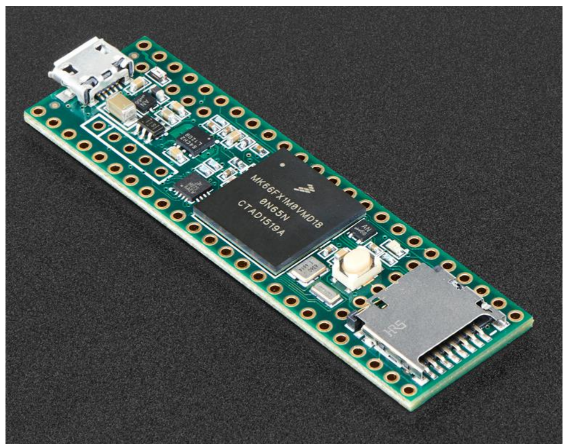


Figure 2: *Teensy 3.6*

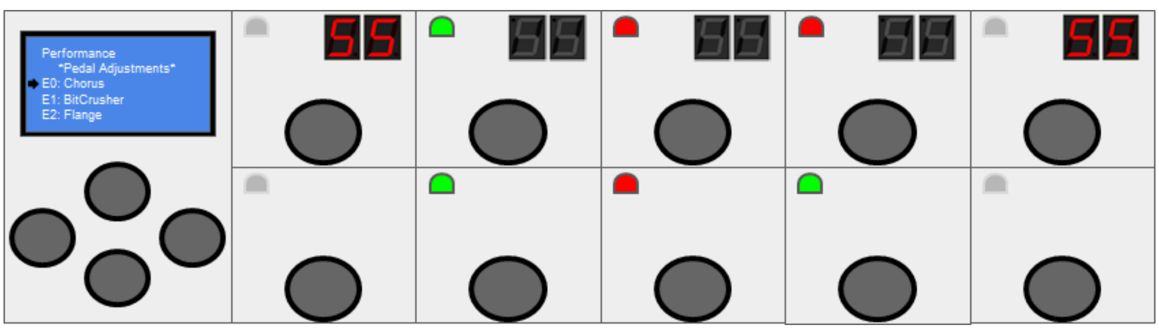


Figure 3: *Board Layout Diagram*

Every square/rectangle is noted as a cell. When two squares line up vertically, it's defined as a column and the rows are designated with top and bottom row accordingly; The software is written with these definitions. The cell on the far right is for the LCD screen to navigate the Infinity Boards' layout and the 4 momentary push buttons are used as a D-Pad to navigate the LCD. The other cells are used to map pedal effects onto the Infinity Board and can change settings by the press of a button to the corresponding cell. When the 7-segment display are in use, it takes the whole column, top button to increment and bottom to decrement. Looking at the overall design layout, you can see that there are 2 7-segment displays (for top row), 1 RGB LED (Blue floating), and one momentary push button for each cell. Each 7-segment display needs 8 pins, 2 pins for the LEDs and one for the momentary push button. Since the Infinity Board will be taking in all of these input/output data pins and sending data to/from the Teensy, there would be: 20 LED pins, 14 pins for momentary push buttons, and 80 pins for the 7-segment display. First of all, the Teensy 3.6 does not come close to the number of pins, plus, the program would repeat numerous blocks of code, not as optimized, and mistakes can take place. To get around this dilemma, we used shift registers that would cut the amount of data pins to 10. Each one of these data pins list above will be stored in a shift register and the Teensy will shift in/out the bits where necessary to process. To handle the Momentary push buttons (Data Driver) and LED Driver, we will be using multiple 8-bit registers to store the values for each input/output. See Figure 4 for the PCB layout.

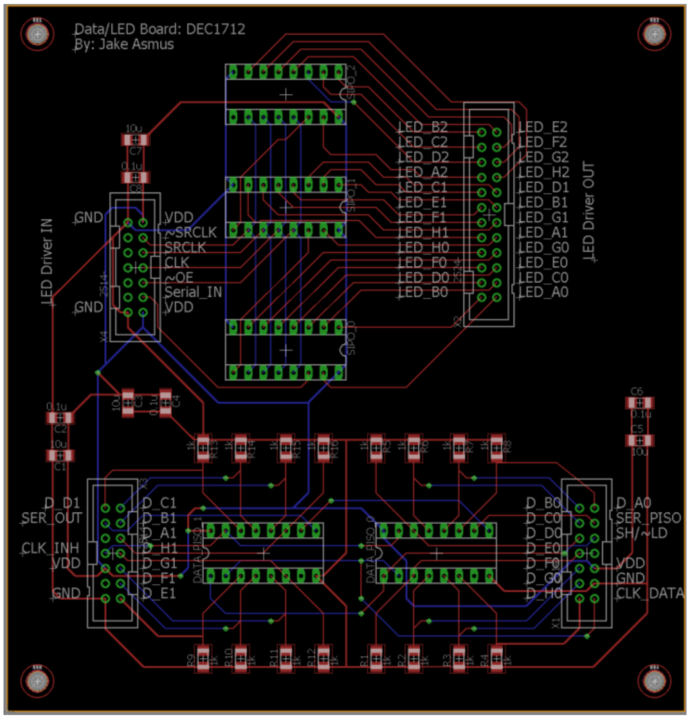


Figure 4: *PCB Layout*

The PCB needed 5 8-bit registers to store the data for the LED driver and the Data driver. To emphasize again, this lightens the load for the Teensy and hands off the load to the shift registers to simply handle high/low signals. For the 7-segment driver, the PCB was initially going to handle the 7-segment driver; however, we came across something similar to our LED and Data driver implementation already on a PCB. See Figure 5 below.

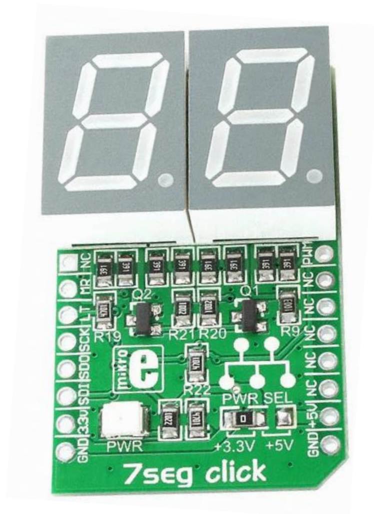


Figure 5: *Dual 7-Segment Display Module*

This premade PCB has the SIPO shift registers on them and only need to connect the data lines from the Teensy to send information. We need 5 of these 7-segment drivers, one for each column. Due to this finding, this makes our PCB smaller because we don't need to integrate 5 7-segment drivers on the PCB. This was a huge find and made our lives easier.

With the Data, LED, and 7-segment drivers on a PCB with shift registers, we can represent all digital logic for the Infinity Board with minimal of 10 data pins instead of 114.

3.1 Related Work/Projects

In the second semester of class we were asked to focus more on other projects that have been done by others or currently being worked on in the world. To compare and contrast our differences and see if there were features that could be shared or improved upon with our project's findings. The main features of our project that can benefit current guitar pedals or sound effect devices would be the digital re-wiring and digital sound effects. The re-wiring feature of our pedal board is unique in the sense that we can change the pedal layout/order by clicking a few buttons. This allows the user to be more efficient and have an easier time changing their pedals around. The gateway to new and creative sounds relies on the changing of pedal order. As the effects pass through a pedal the output signal becomes the input signal for the next pedal, so by changing the order you can change the input signal being received by the next pedal in line. There are some products that try to achieve this effect, but use physical wiring still. An example of this is the **Patchulator 8000** [4], which allows users to change their pedal order by plugging them into this device in a different order. Their device would greatly benefit from the ability to adjust pedals' order wirelessly.

We also discovered similarities with other multi-sound effect pedals for guitar, showing that you can have several effects in one pedal board. Many of these multi-effect boards do not offer quick and easy re-wiring of the effects, which could help increase their sound opportunities. There are a few pedals that allow the user to change the order of the various effects, but not very many seem to have the ability to change the settings of a pedal with just using the buttons on the board, especially not in a live setting.

3.2 Proposed System Block diagram

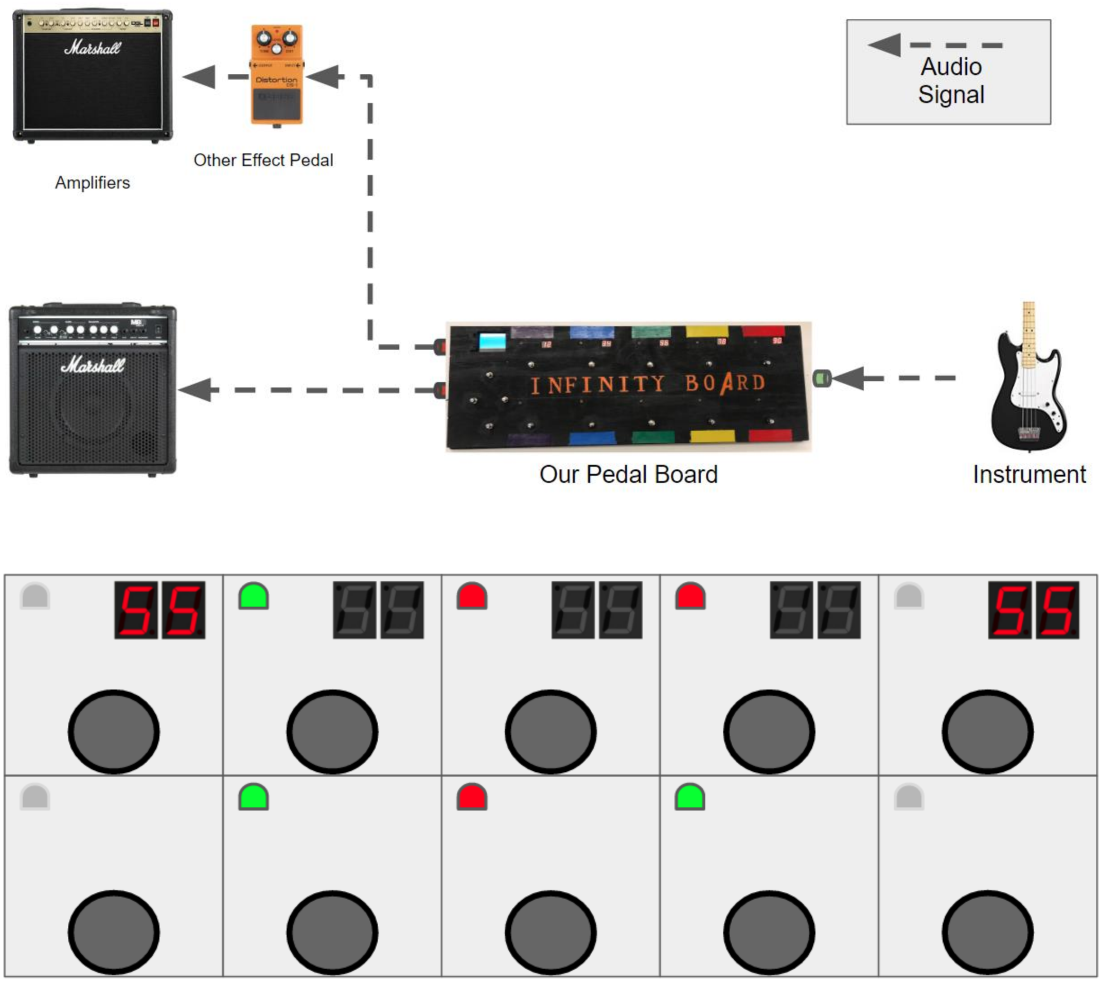


Figure 6: *System Block Diagram for the Pedal and Pedal Board Layout/Flow of Information*

From the diagram above (Figure 6), you can see the general flow for our pedal and pedal mat designs. We want to take the input from the guitar/bass/keyboard/whatever instrument they're using and use that input to manipulate and split the signal. The pedal then has the option of sending the input signal and the manipulated signal to other pedals, or sending their signal straight to an amplifier. The main function for our pedal is to split the input signal to manipulate another signal and synchronize it with the original input signal. The pedal mat feature would interface to our pedal, and from there it will be able to control multiple features of a wide variety. Ultimately the goal of the pedal mat is to have a section to control every feature on a pedal that would normally be adjusted by hand. This allows the user to have the freedom with their hands, and the luxury of adjusting their sound live.

3.3 Assessment of Proposed methods

Pedal effects in the old days function mostly in the analog domain. However, the world is switching to a more digital world and now, pedal effects are appearing in the digital domain. Within the Teensy 3.6 microcontroller family, we can take analog pedal effects and convert the effect into the digital domain. The Teensy microcontroller family has many shield adaptors we can integrate to make these pedal effects digital. For our design and purpose, the audio adaptor is the perfect fit for our project.

The [PJRC] has a great GUI interface for chaining pedal effects. You can sequence your full effect with multiple pedal effects and once satisfied, you can export the netlist code and compile onto the Teensy. With the correct hardware setup, you can affect the input signal and hear the effect output.

When compiling your code onto the Teensy, there are “knob value” setting you can specify to get a specific sound to your effect. We want to change these “knob value” on the fly without recompiling. This is where our pedalboard comes into play.

See the Figure 7 below for an example of a pedal effect pipeline. As you can see, there are three effects in series with the input on the left and output on the right.

Figure 7: *Effect Pipeline Block Diagram*

These three pedals have many “knob value” settings that can be altered to change the sound effect. The pedalboard will be used to change these digital effects. You change the volume gain, sampling resolution, number of bits, voices etc. All of these “knob value” settings are on the corresponding analog pedal effects. For the analog pedals, you simply turn the knob to the corresponding level with your hand. For the digital pedals in the Infinity Board, you have to specify in code what corresponding level you want. When in the digital domain, the possibilities on how to change these “knob” effects are end list. For the sake of this project, we decided to keep the tradition “foot stomps” to alter the effect. Now, instead of using your hand to change, you can now use your feet to change the effect and have your hands on your instrument and playing. We can now change the effects on the fly! With an analog pedal, good luck with playing your instrument and bending down to change your knob by hand. This can change how music is produced on live stage. But it can get better.

When using analog pedals, your serial sequence is wired together in a specific arrangement. Let’s say you want to change that arrangement. How would you do it? Well, you would have to re-wire the pedals in the corresponding order of your choosing using your hands etc. What if you can change the effects with our pedalboard by pressing a couple of buttons with your feet and have your hands on your instrument? You would be able to change your effect sequence on the fly! This, again, could change how music is produced on live stage.

As you have read, you can see that we are changing on how live music is produced and making it easily for the user to switch everything by foot instead of hand. Why not keep your hands on your instrument and switch the effect or sequence on the fly? It would change the musician's way of performing. But it can get better. With analog pedal effects, you purchase one pedal and you get that one pedal. In the digital domain, all you have to do is copy some lines of code and have a plan to pipeline the effect where it's desire. This could be cost effective for many individuals to have a duplicate effect instead of purchasing another. Everyone likes to save money.

For all of this to come together, we will need a fast microcontroller (Teensy 3.6) to complete digital signal processing, logic to change effect "knob values", effect sequence, and produce a clean signal out. Since our pedalboard is 2-by-5, we will have a total of 10 buttons to edit the "knob values" and on/off switches and have an LCD to show which page is mapped onto the Infinity Board. The pedal effect sequence can be changed with a "D-Pad" by selecting the corresponding pedal sequence and swap out the effect with another using the LCD and D-Pad. LED's will display the states of the pedals effect as a whole (on or off) and 7-segment displays the state of the "knob value" for that same pedal.

3.4 Validation

To verify our pedalboard pipeline effects, we will have multiple sound variation tests: Rearrange effects, turn off pedal effects, change pedal effect settings, and listen for the differences. We can hard code the effects and compare the audio from the pedalboard.

Verifying the audio coming out of the Infinity Board will be more of a challenge. The main area to focus is the sound differences. To the nature of effect pedals, if you have an arrangement and make small alterations to the arrangement, one can hear the differences and possible hear a "new" signal. The next part for audio verification is to hear how a pedal effect changes when you alter the "knob values", volume, and on/off states.

Verifying hardware associated with the Infinity Board, we will have to verify that the PCB is designed correctly. To test the PCB, we have three testing programs that's compiled onto the Teensy and test with either serial or LED's. There's a program for the LED driver, 7-segment driver, and for the data driver. The 7-segments and LED driver will simply be verified with visuals of the LED's and compared to the program and make sure it matches. For the Data driver, we will use the Teensy serial terminal and have appropriate serial statements within the program to let us know which button is pressed/active. Once all tests past, we know that the PCB works as designed and can be used for the Infinity Board.

4 Project Requirements/Specifications

4.1 Functional

The functional requirements for this pedal design are to output a transformed and original signal. We want the pedal board to easily be controlled by the user with the use of their foot instead of hands. The pedal board will have a LCD, 7-segment displays and LED's to read the state of the current effect and four buttons for menu navigation and 10 buttons for effect fine tuning.

- Pedal Board
    - Ability to output a signal that models a series of chained effect pedals
    - Raise input/output volume up and down
    - Output an unmodified input signal
    - Reprogrammable 2-by-5 button interface layout
    - LCD, 7-segment display and LEDs to notify the user what state the microcontroller is currently in.

The Infinity Board will have the ability to change the pedal effects serial arrangement, alter each individual pedal effect on the fly by foot, and an LCD screen to show pages' selected layout that's currently on the Infinity Board ready to be altered by the momentary push buttons. See Figure 8 for the LCD layout.

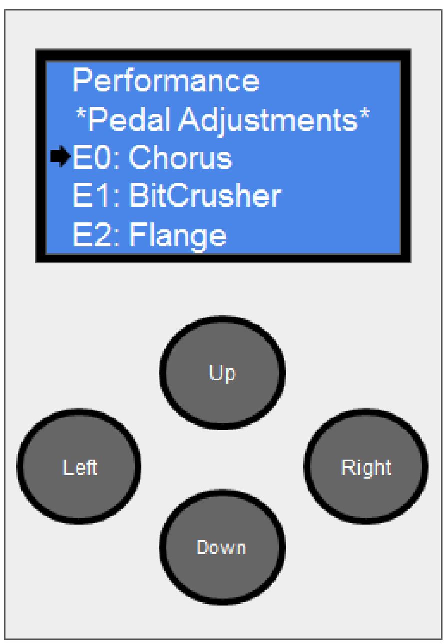


Figure 8: *Menu LCD Screen Layout*

For instance, in Figure 8, "E0: Chorus" page is selected. Something to note is that any page selected under the "Pedal Adjustments" (E0, E1, and E2) will map the Infinity Board to the corresponding pedal effect to be directly adjust by using the corresponding momentary push buttons. The “Performance” page simple has the on/off switches for effects E0, E1, and E2.

4.2 Non-functional

- Run off of a 9-volt power supply
- There must not a significant delay between the input signal going into the effect pedal and the signal being sent out of the effect pedal

4.3 Standards

We wrote our code in Arduino C/C++. Some goals of this style are to optimize for the reader, not the writer and to be consistent with existing code. There are not very many protocols for guitar/instrument pedals, since the industry is very vast and spread amongst a wide variety of professionalism. Some guitar pedal manufacturers are part of a global company and are of a higher level of professionalism, but some users seek out the pedals by hobbyists for their unique design and sound. One of the general standards for guitar pedals is that most use a 9V power supply. We would like to incorporate a standard 9V power supply barrel plug, but the current version uses a standard 5V USB port.

5 Challenges

One major roadblock we faced with programming is some of the constraints of the Teensy Audio library. A particular constraint was the restriction of not being able to rewire audio blocks during runtime. Audio connections in code had to be declared during the compilation time which limited the ways we could dynamically switch effect sequences.

We found a way around this by using mixer audio block objects. These audio block objects allow four channels of digital audio signals to be mixed into one output. This allowed us to create a pipeline of three “Effect Pipes” by shutting off certain inputs on the audio mixers to simulate switching out effects, pictured in Figure 9.

Another challenge was wiring up the PCB to the actual physical board. The cable management was a mess. It took a couple of days, wiring up and testing the board with the programs listed in the Validations header. Away around it was to label everything and write down what we would do for the next step so we could pace ourselves. Once everything works, everything was “nailed” to the floor. It was a frustrating process and it was satisfying when the cable management was complete.

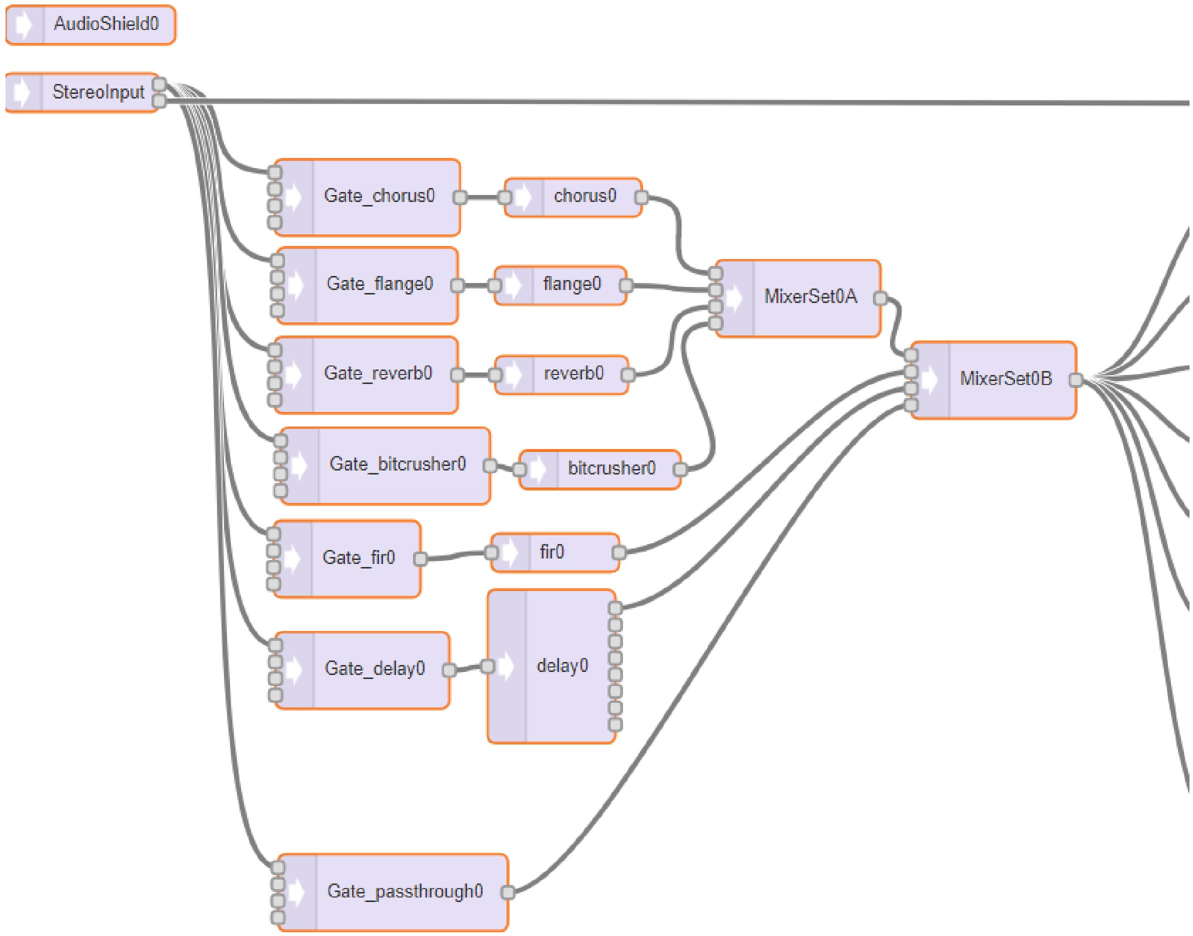


Figure 9: *Portion of Audio System "Effect Pipeline"*

6 Overview

6.1 First Semester

To start the first semester, we are conceptualized some ideas for a pedal. After we brainstormed we then went out and researched some pedals similar to the functionality of our own. Upon finding the circuit diagram for the Green Ringer pedal (2) we built this circuit in PSPICE to test the functionality of the circuit. We found the pedal could achieve a frequency that was double of the input sine wave, but the amplitude could be inconsistent, depending mostly on the amplitude of the input signal. After some more thought we thought we could implement the frequency doubling/halving using digital logic. Once we were settled on the basic idea to make our pedal work, we discussed how we would want to interface with our pedal. Which made us think of the creative way of controlling every feature of the pedal using a pedal mat. We figured that the average musician could not afford to free their hands to make adjustments to the actual pedal while playing so we made every feature controllable with their free feet. Then we discussed how we would layout our pedal, and the general functionality of our pedal mat came into focus. The pedal mat should make small incremental changes depending on how many times someone taps a momentary switch, with higher on the mat increasing the level and lower on the mat lowering the level. The momentary switches would deliver a higher voltage to the microcontroller, which would then interpret the signal and communicate back to the pedal what would need to be adjusted. After the functionality of the mat was figured out, we moved back to making the pedal work. The microcontroller we decided on using was discovered to be powerful enough to measure frequency of input signals and then output an analog signal. Once we worked out the kinks of measuring the frequency we then moved to testing the output values to make sure the frequency was accurately doubled/halved.

6.2 SECOND SEMESTER

This semester we leaned more towards the digital side of the sound effect devices. This was a sharp change to our previous ideas which incorporated more analog changes to the input signal. We kept the idea involving two outputs, one effected signal and one original signal. The concept of a pedal and a pedal mat, quickly became just the pedal board. We condensed the overall concept to function off a Teensy 3.6 microcontroller, with the sound effects coming from a digitally altered input signal. The benefits of digitally altering sound opened several doors for our project, one of which became re-wiring effects digitally. This would allow the user to change their effects around quickly and easily, no more tangled messes or long hours chasing unique sounds. Just a few clicks on the menu buttons and the whole board could be rearranged. The use of the Teensy opened a lot of sound effect options as well, since several programmers online could write libraries of different effects. The possibilities for new sounds grew to new heights, with every additional pedal added to our board.

As we made big strides in the programming of our effects, we improved upon the physical design of the pedal board as well. We added 7-segment displays to each column to show the setting adjustments for each pedal. Included RGB LED’s for each toggle button to make the current state of the pedal board easier to see and understand. Those two installments were in-line for our first semester design as well, but the big change for the pedal board physically in semester two came from the addition of the menu buttons and screen. This would allow the user to adjust the board more efficiently, since they would be able to glance at the LCD screen and see the board’s mode. The menu buttons also added to the efficiency of the pedal selections and adjustments, while still leaving appropriate room for the user to click around with their feet. All of this made possible from the PCB layout that Jake Asmus made.

7 Conclusions

We created a single pedal board that houses several effects digitally, and allows the user to adjust any feature of the pedals by stepping on them. With a series of digital effects and filters that can be rearranged and “re-wired” in any imaginable combination to create even more diverse sounds.

8 References

1. Octave-down fuzz
    a. http://pedalparts.co.uk/docs/BlueFool.pdf
2. Green Ringer Pedal
    a. http://www.generalguitargadgets.com/pdf/ggg_gro_sc.pdf
3. Royal Blood Band: YouTube video of the octave up/down in use.
    a. https://www.youtube.com/watch?v=ere2Mstl8ww&feature=youtu.be
4. Patchulator-8000
    a. http://www.boredbrainmusic.com

Appendix I - Operation Manual

User Manual for The Infinity Board

Table of Contents

Reference Guide to the Pedal Board 1

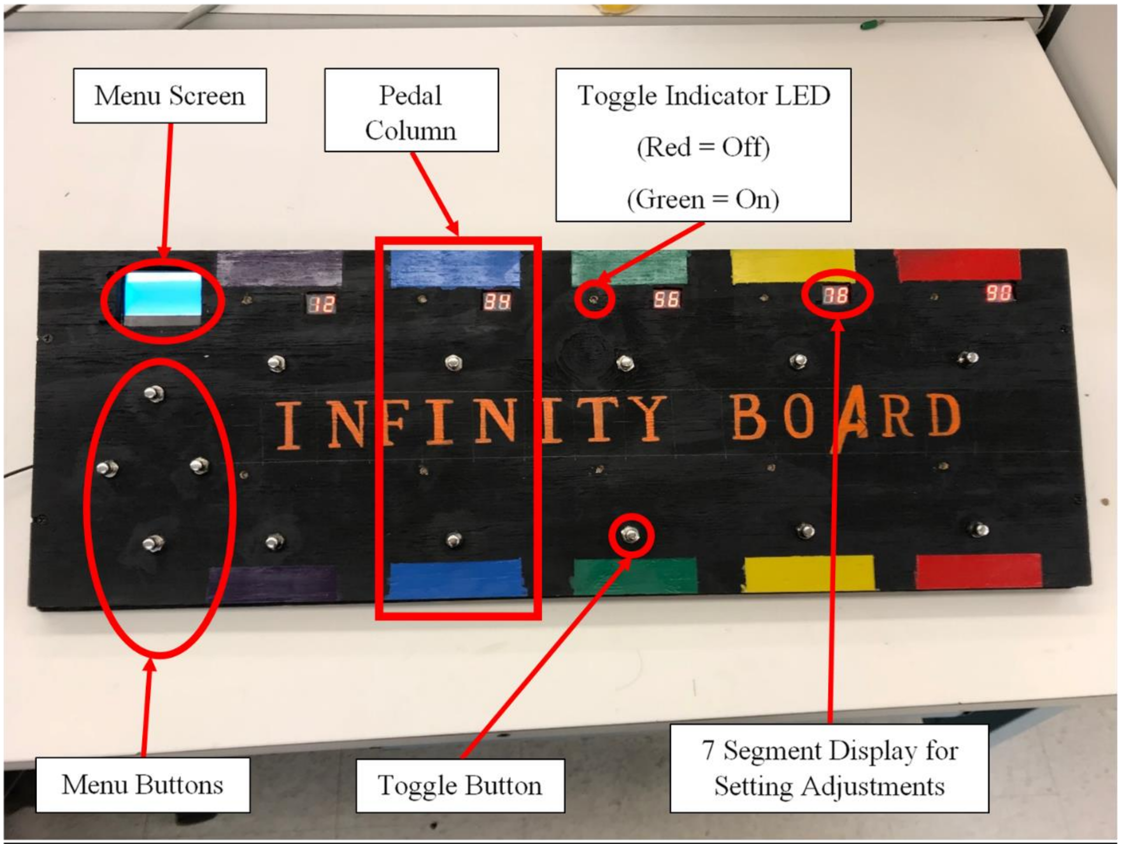
Powering Pedal Board On 2

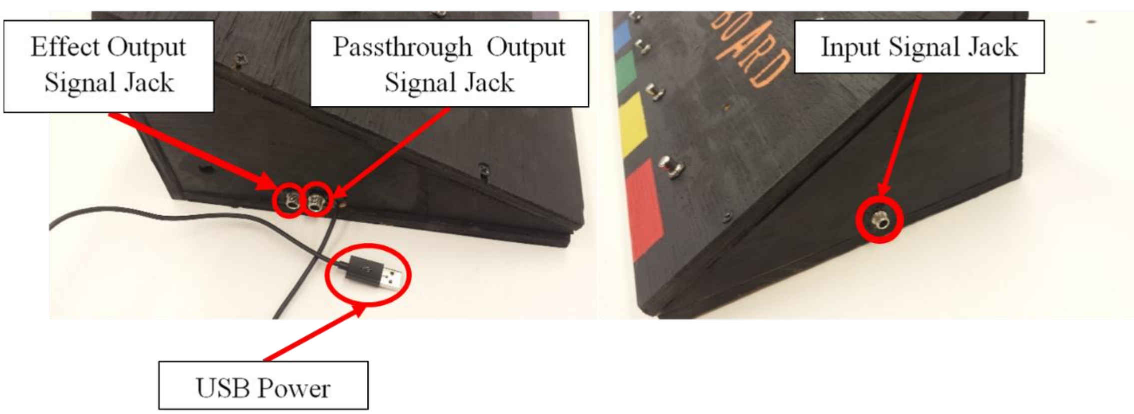
Choosing Your Pedal Layout 3

Adjusting Settings for a Pedal 4

List of Effects 5

Reference Guide to the Pedal Board

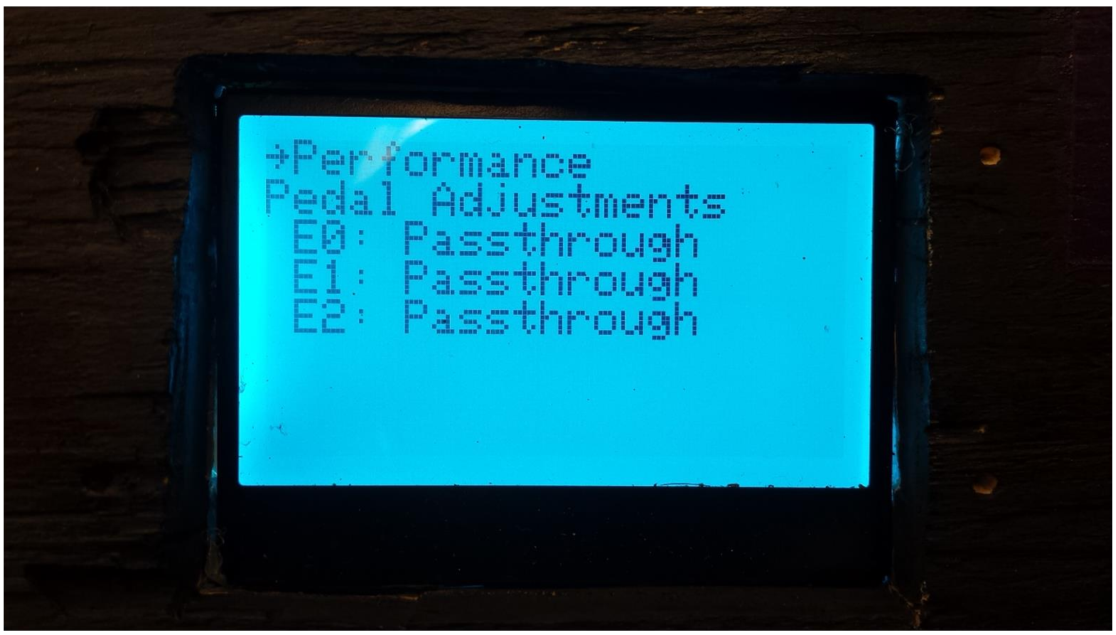




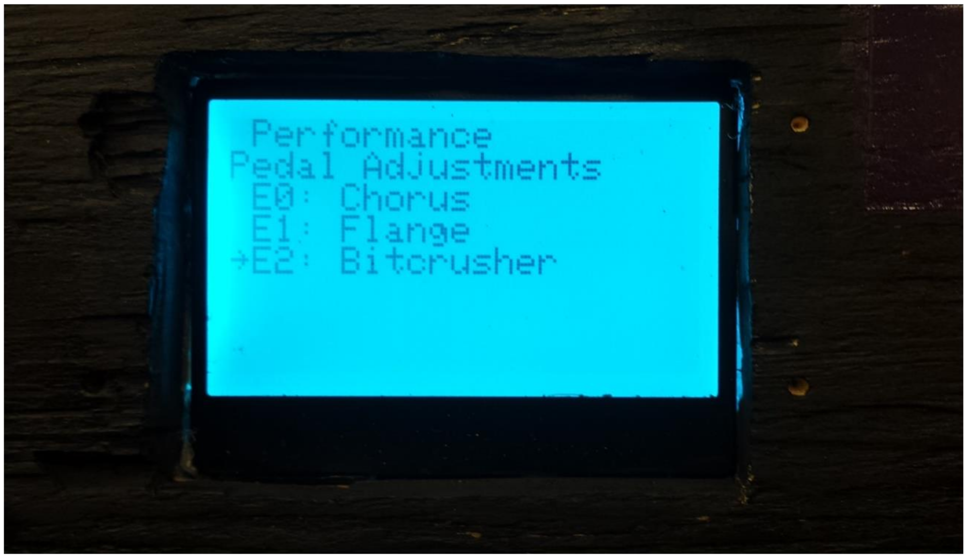
Page 1

Powering Pedal Board On

1) To turn the device on, plug the power cord from the left –side of the board into a USB at 5 Volts.
2) The device will then power up, this can be seen by the menu screen lighting up with a blue background and displaying:



Choosing Your Pedal Layout



1) To choose your pedal layout press down on the menu buttons until the → is on E0, which is the first effect in the sequence.
2) Then to choose your effect you click on the left or right menu buttons until the effect you desire is selected.
3) After your first effect is selected you can add any additional effects by moving the → to the desired effect location using the menu buttons and repeating step 2.
4) When you are done selecting your effect pedals in the order you would like to use them (from right to left) you can move the → back to "Performance" on the menu screen to use the pedal board's performance mode.

Adjusting Settings for a Pedal

1) To adjust the settings for a particular pedal, you must move the → down to the desired pedal location using the menu buttons.
2) This will open up that pedal's settings page, this can be seen by clicking in one of the pedal's columns, which then displays as that setting underneath "Fine Tuning:" on the menu screen.
3) After you are finished adjusting the pedal's settings you can return to "Performance" on the menu screen to use your updated pedal.

List of Effects

1) **Pass-through**
2) **Chorus**
3) **Reverb**
4) **Flange**
5) **Bit Crusher**

Appendix II - Alternate Versions

In this section we have some of the changes or revisions that we have made to the *Infinity Board* since we started the project in Spring 2017.

Pedal Mat Design (Semester I)

1. Physical Layout
    a. 2-by-6 button layout
    b. Size: 0.75x0.35 (m)^2
    c. Back row raised to accommodate for the chance of accidently pressing two buttons
    d. Button Square
        i. Each will have a LED display
        ii. Each will have a two digit 7-segment display
2. Circuitry
    a. T Flip Flops
    b. Multiplexers
    c. Pull-up and pull-down networks
3. Controller
    a. Teensy 3.6 microcontroller
    b. Teensy Audio Adapter

The changes that we made to the physical layout have mostly to do with the ability to use our board more effectively. The 2-by-6 button layout gave way to 2-by-5 button layout to allow more room on the board and make the space necessary for the menu system. The overall size of the pedal board stayed relatively the same, minor adjustments to create a more balanced layout. The reason we did not raise the back row is because we felt the user had enough space to maneuver their feet effectively among the toggle buttons. The button squares translated better into columns to change the settings of the pedals.

Our circuitry ideas changed very drastically after our first semester, mainly due to more thorough research into the area of digital pedals. Daniel Peterjohn demonstrated a larger array of effects through a programmable library that could easily perform on the Teensy 3.6 microcontroller, so we left the analog effects alone. We also had run into a problem trying to manipulate the input signal into lower and higher octaves like we originally intended.