Sound Effect Devices for Musicians

DESIGN DOCUMENT

Team Number: Dec1712
Client: Professors Geiger and Chen
Adviser: Professor Geiger

Team Members/Roles: Jake Asmus/Team Leader
Joseph Brown/Team Communicator
Daniel Peterjohn/Team Webmaster
Jiangning Xiong/Team Key Concept Holder

Team Email: dec1712@iastate.edu
Team Website: http://dec1712.sd.ece.iastate.edu/
Revision: 4/21/17 Version 2

Contents

1 Introduction 2

1.1 Project statement 2

1.2 Purpose 2

1.3 Goals 2

2 Deliverables 3

3 Design 4

3.1 System specifications 5

3.1.1 Non-functional 8

3.1.2 Functional 9

3.1.3 Standards 9

3.2 PROPOSED DESIGN/METHOD 9

3.3 DESIGN ANALYSIS 9

4 Testing/Development 11

4.1 INTERFACE specifications 11

4.2 Hardware/software 11

4.2 Process 11

5 Results 13

6 Conclusions 14

7 References 15

8 Appendix 16

1 Introduction

1.1 Project statement

We plan to make a multiple pedal interface and combine them into one functioning pedal. With the ability to control our pedal with an easier interface with greater versatility for users.

1.2 Purpose

Our main purpose for this project is to provide musicians with a new pedal that can accomplish many of the sounds achieved in various pedals, and to give them a better interface to control their pedals. 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. 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. With our pedal mat interface, we could use foot switches to adjust each setting on a given pedal that's compatible with our design. This mat would free up musicians to adjust their pedals to their heart's content while playing their instruments uninhibited.

The pedal we are designing is unique itself by having the necessary interface that allows it to be controlled by the pedal mat. Using a microcontroller to communicate the adjustments to the rest of the pedal, making adjustments live seamlessly. Another feature unique to this pedal is the splitting of the input signal and manipulation to the split 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 signal.

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 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 any more. With our pedal mat, the performer can quickly adjust any setting on the pedal hands-free!!!

1.3 Goals

The goals we would like to accomplish are having performers in a more relaxed setting with all their pedals at a single location, free state of mind from hazardous wires, and programmable interface 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.

2 Deliverables

At the completion of this project we are hoping to deliver a pedal with multiple functions, and a pedal mat as an easy-to-use user interface.

- Pedal
    - The pedal will be able to have various functions built in such as:
        - octave up and down adjustment
        - delay
        - reverb
        - distortion
        - bit crusher
        - and other functions normally done with multiple pedals.
    - The pedal will have three IO channels
        - one input channel
        - two output channels
            - output of the original signal
            - output of the manipulated signal.
    - The pedal uses digital features on it to control the fine details of a signal such as:
        - volume
        - time delay
        - gain
        - or another part of the signal.
    - This pedal should be roughly the same size and shape as an individual guitar pedal and be run off of a 9V voltage supply.
    - With a digital control for our pedal we will be able to adjust the features using an independent pedal mat.
- Mat
    - The pedal mat should be able to control almost all of the features on our guitar pedal remotely.
    - The pedal mat should have a voltage supply of 9V with a 1A draw.

The mat should be easy to adjust, and easy to understand what is on/off depending on the LED display/seven segment display for the pedal.

3 Design

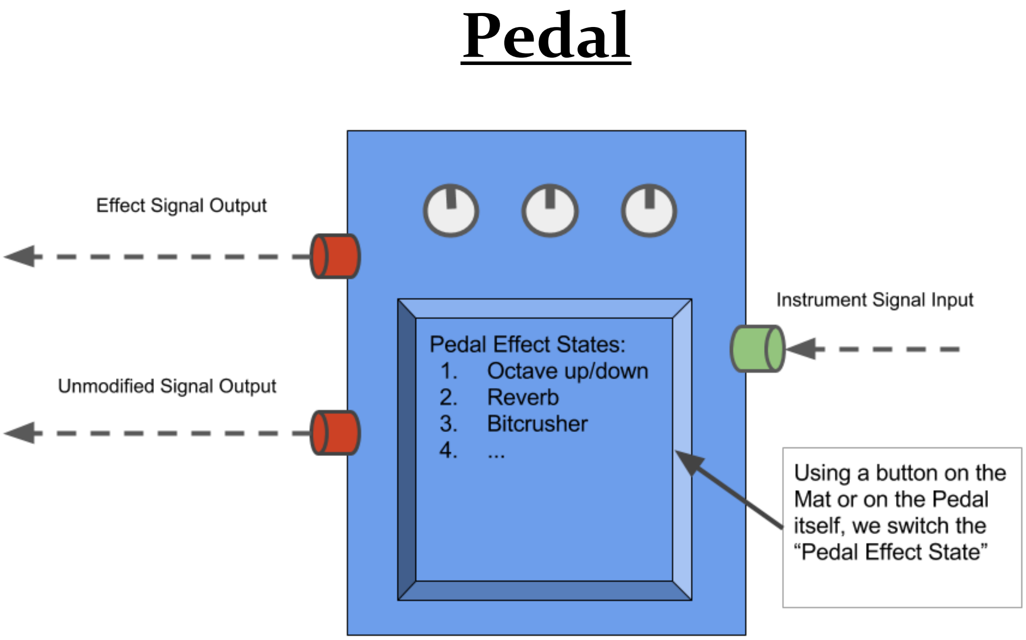
We started with the problem of taking an input signal to create two output signals, where one of the signals was the original signal sent in. With two outputs, we could give the user of our pedal the option of sending the signals through separate amplifier systems or pedal rigs. To approach this issue, we decided that we could adjust the signals for either an octave up, by doubling the input frequency, or more the signal an octave down, by halving the input frequency. Originally, we thought the best way to do this would be through an analog circuit. So, after some research Jake Asmus found a circuit to base our design off of (1), he built the circuit and tested it using PSPICE. The output signal seemed to be the octave up signal we had been looking for, but the wave looked to be changing in amplitude over time. After more research, we concluded that halving the frequency from the analog circuit might not be quite possible during our time for the senior project. So, we turned our attention to a digital means for manipulating the signal.

The group then considered that using a microcontroller with an ADC input and a DAC output might be the best way to manipulate the circuit. Shortly after pursuing this idea we found out that our microcontroller, the Teensy 3.6, had some extensive capabilities (2). The Teensy 3.6 could read sample audio input at sampling rate of 44.1kHz at 16 bits depth (3), then manipulate the digital samples with the Teensy's Audio Library effects (4). We can then start coding our own custom effect to output a value of either an octave up or an octave down. At the time we were contemplating our main problem, another arose. We wanted to know of an interface we could use with the device to allow adjustments to be made to our pedal during a live performance.

When a musician is in a live performance setting they usually have their hands occupied with playing their instrument. This makes adjusting their pedals impossible on the fly, which could be very useful. Joseph Brown, our group musician/performer, has had a tough experience with trying to adjust pedals before at live shows with his band. Whether it was bumped while traveling or kicked accidentally by himself, the pedal settings were all kinds of messed up. He wouldn't have time to adjust all the knobs while playing the song and would have to forgo the experience with the pedal altogether. This brought about the interface of the pedal, a 2x6 section mat with multiple buttons and functionality to control every feature remotely on our pedal. The musician's hands may be preoccupied, but their feet are used to turn pedals on and off frequently throughout the performance. So, the pedal mat allows the user to adjust the knobs and even control the stomp buttons remotely.

To solve this unique problem, we decided a pedal mat layout would be most useful in the size similar to a pedal board. To avoid stepping on both sides at once the back half of the mat is to be raised to allow the user to easily adjust each setting with the push of a foot. The momentary push buttons seemed to be the best use for multiple functionality, where a circuit could latch from an impulse signal, be held down to increment multiple times, or be pressed once to increment/decrement a knob setting on the pedal. Once the button is pressed the microcontroller will then read the input with a given pedal layout and communicate the adjustment back to the pedal wirelessly. So far, the initial test for this logic circuit has been implemented by Jake Asmus using JK Flip-Flops to create the latched state. After some further testing the JK Flip-Flops were ruled out as they would require two voltage sources and complicate the inner workings of the mat. Daniel Peterjohn then suggested that we could store and change the values using shift registers. The benefit of using output shift registers is that it allows the microcontroller to control the 24 LEDs with only three IO pins being used on the microcontroller. In addition to this, the input shift registers also allow the microcontroller to read 16 button or more inputs using only four IO pins on the microcontroller. See the general overview of the project flow in figure 1 of the appendix.

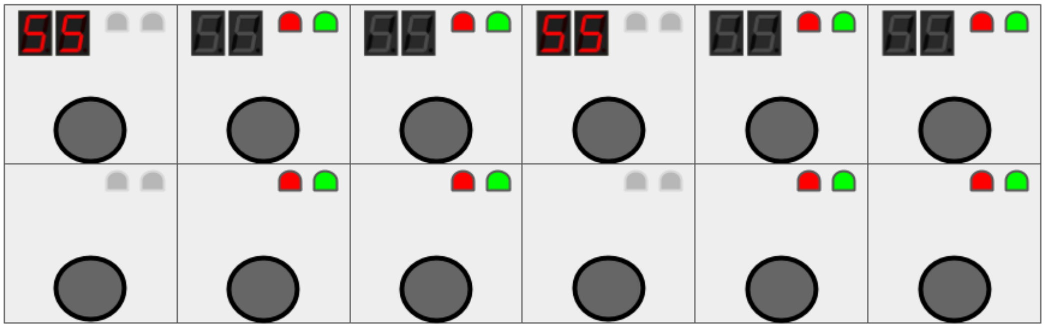
3.1 System specifications

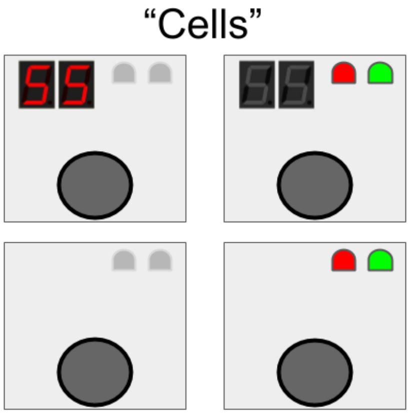


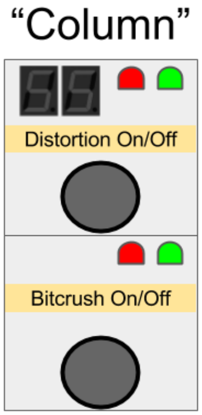
- The pedal will need to run off of a supplied 9V power supply
- Quarter inch instrument cable for input and outputting two quarter inch instrument cables out.
- The input signal from the guitar will have to originally be offset to work with the Teensy microcontroller to fit in the range of 0-3.3V.
- The signal is interpreted by the microcontroller the signal at the first output will be equal to the incoming signal, and the second output will be the manipulated signal.
- The output signals will then be adjusted back to their original offsets to work with the guitar amplifiers or other effects pedals.
- The microcontroller of the pedal will also interpret the signals from the microcontroller in the pedal mat to make adjustments to the settings of the pedal such as:
  - input volume,
  - octave up/down volume,
  - mixed volume,
  - and adjustments made to the output signals.

Mat

Overview of the Mat



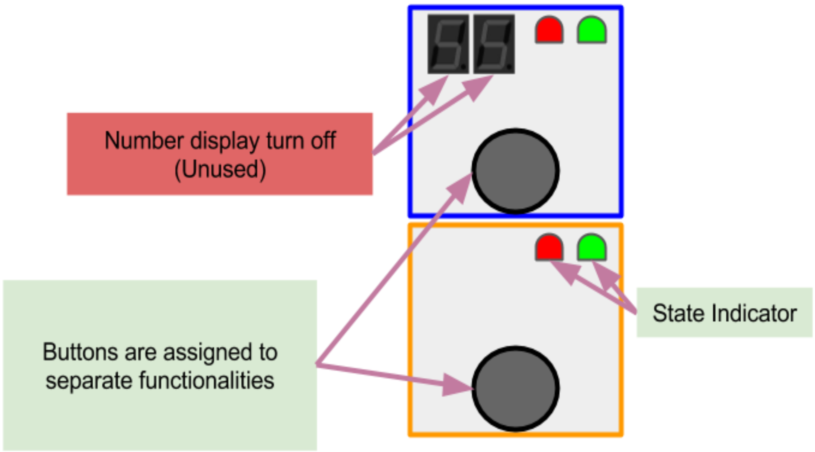




- Two types of Mat Cells
    - Bottom Mat Cell:
        - One momentary push button
        - Red and green LEDs
    - Top Mat Cell:
        - Two Seven segment display
        - One momentary push button
        - Red and Green LEDs

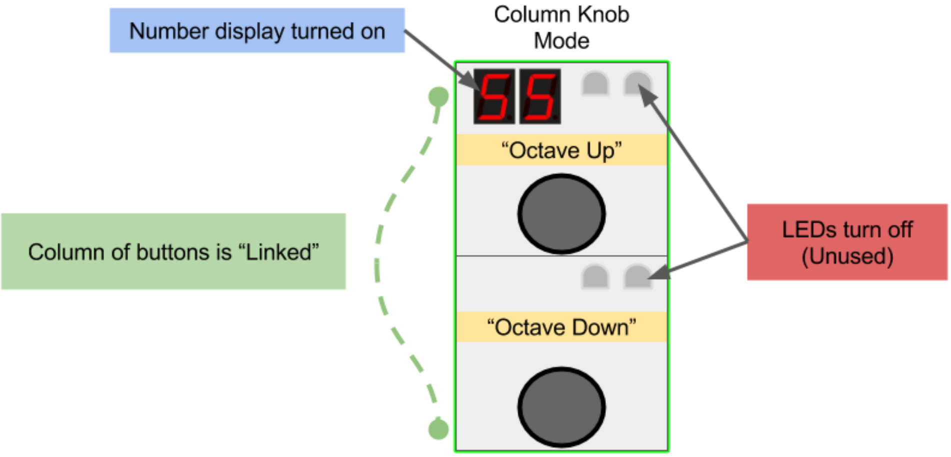
- Mat Column:
    - Contains top and bottom Mat Cells
    - Type types of Column Modes

Column Toggle Mode



- Mat Column will set in the knob state will use input from the top and bottom Mat Cells
    - Input from the two Mat Cells will increment/decrement a value on the pedal by predetermined increments
    - The top Mat Cell in the Mat Column will display the value with 7 segment LED's

Column Knob Mode



- Mat Column will set in the knob state will use input from the top and bottom Mat Cells
  - Input from the two Mat Cells will increment/decrement a value on the pedal by predetermined increments
  - The top Mat Cell in the Mat Column will display the value with 7 segment LED's

Other Features

- The microcontroller of the mat will then communicate the adjustments wirelessly back to our pedal.
- The pedal mat will also run off a 9V power supply and consist of 12 sections in a 2x6 formation.

3.1.1 Non-functional

- Minimal delay in pedal response time between input and manipulated output signal.
  - The purpose is to perform in a live setting, so if our adjustments are made at a delayed rate the effect won't occur at the right moment.
- Minimal delay between original and manipulated output signal
  - if the delay is too big between the signals the frequencies could mix in more destructive ways.
- Balance between the octave signals is set proportionally to each other to allow maximum control for the user to adjust them.
  - Without a normalized output signal between the octave up and octave down signal one might overpower the other frequency and mask the effect.
- The system should also be able to clearly signal the state of the current device and make the changes easily seen by the user.

- This will ensure the maximum efficiency for the performer when they are trying to make adjustments live.

3.1.2 Functional

- The functional requirements of this project will be that the pedal will
    - accept the guitar input
    - offset the voltage so the Teensy microcontroller can read the frequency values
    - output the original signal and manipulated signal,
    - readjust the offset to go into other pedals or the perspective amplifier.
- The pedal will also have to adjust the pedal settings based upon the communication between the microcontroller in the mat vs. the microcontroller in the pedal.
- The pedal mat will have to wire each button press to the perspective logic values, and translate those changes to the pedal.

3.1.3 Standards

Our project is not really constricted by standards of the guitar pedal industry. The industry does not set about certain values for the power supply, size of the pedal, or any particular work method. Although most of the guitar pedals in use have a 9V power supply usually with a barrel connector. We would like to keep the pedal around the size of a single guitar pedal to make it easy to contain on preexisting pedal boards. The output value of our pedal will have to be closely monitored to ensure that we are not harming the instruments or amplifiers hooked up to the pedal.

3.2 PROPOSED DESIGN/METHOD

So far, the team has decided to implement the digital adjustment method for the pedal, allowing us to pursue other digital effects once we confirm our initial effect is working as designed. By creating a digital pedal, we can also adjust the various knobs that control input volume, octave up volume, octave down volume, and mixed volume digitally with communication with the microcontroller. The pedal mat is moving forward with the implementation of the momentary push switches and creating various logic circuits. Using shift registers, the microcontroller can manage many input and output signals with very few pins (5). Using output shift registers, the 24 LEDs are controlled with only three IO pins being used on the microcontroller. The seven segment displays are coupled with the output shift registers and seven segment decoders and will also only use three IO pins on the microcontroller. The input shift registers can read 16 buttons or more inputs using only four IO pins on the microcontroller.

3.3 DESIGN ANALYSIS

Our design seems to have the promise to solve our main problems of outputting two signals and using an interface that allows a musician to adjust pedal values in live performance settings. Although the testing has been a slow process, we are making ground on understanding the capabilities of the Teensy 3.0 microcontroller. The microcontroller had some issues displaying the input frequency to the serial port, and after spending enough time with that we decided to upgrade to the newest version of the Teensy microcontroller. The Teensy Audio Adapter (3) has allowed us to have very clean input and output audio.

We have made progress on testing our pedal mat as well, with the shift registers working exactly as we expected. The output shift registers are able to control 16 LEDs individually as well the input shift registers are able to read 8 push buttons. Also, the output shift registers are able to display correct numbers with the seven segment displays.

A pedal mat library is being designed and tested to abstract the states of the Mat Cells, Mat Columns and Mat Pad using object oriented programming. This allows the Mat Pad to be flexible, reusable and extensible. The Mat Cell class keeps track of toggle states and writing the corresponding LED states. The Mat Column class manages the states of the top and bottom Mat Cells depending on the corresponding knob or toggle column state. Finally, the Mat Pad class encapsulates all of the Mat Columns for easy access and flexibility.

4 Testing/Development

4.1 INTERFACE SPECIFICATIONS

The interface for this project will include the hardware interface for the pedal and the pedal mat, and include a software interface between the pedal and the mat. The hardware interface of the pedal should be a few push-buttons to control the effect the user wants to hear. With the use of some knobs that will control the adjustments to the manipulated signal, like volume or tone. The pedal mat interface should use push-buttons/push-platforms to control the functions of the pedal mat, which will then communicate the functionality back to the pedal using communications between the microcontrollers. The software from the pedal mat will interpret the button presses with digital logic to then determine how to communicate the changes necessary for the microcontroller in the pedal.

4.2 HARDWARE/SOFTWARE

The hardware and software used so far in testing the capability of our pedal and pedal mat have been limited to mostly using a function generator to create the sine wave representing an input signal and then trying to measure the frequency using the Teensy microcontroller. The microcontroller then displays the serial output of the frequency read going through the ADC of the microcontroller. After reading the value we will adjust the frequency either up an octave, down an octave, or both signals at the same time. The output will then be measured by an oscilloscope on the original signal out and the manipulated signal out to check for accuracy. We will also use a multimeter to measure the input and output voltage to make sure the guitar/bass amp is receiving the proper offset voltage to play without harming the amplifier.

The pedal mat will mostly be checked using a microcontroller displaying a serial value of a high input or low input depending on if a switch has been pressed. We will set up a basic pedal mat layout that has various functions for each section to guarantee that the input shifting registers are changing states when pressed, and that the input shift register are read clearly by the microcontroller to increment/decrement the volume by one level of increment/decrement. The final test of the pedal mat is to ensure that the values are clearly communicated between the microcontroller of the pedal mat and the microcontroller of the pedal.

4.2 PROCESS

The process of testing usually is derived from how the signal is designed to flow through our pedal process. The signal originates from the guitar/bass, then travels to the input jack of our pedal. From there the voltage of the guitar/bass signal must be offset so the lowest voltage value is above 0V, this includes our first measurement test to be done. Once we verify the input voltage is adjusted properly we can send the signal into the microcontroller's ADC port, from there the microcontroller measures the frequency of the input signal. The next test is done to confirm that the input frequency is indeed measuring a correct value and is sent to the designated output port, the manipulation then occurs with the microcontroller manipulating the frequency either up an octave, down an octave, or both. This signal is then sent to the second output port, where we measure the frequency to confirm that the adjusted values are accurate to the octave values we expect. The output signals then are converted back to analog signal with the microcontroller's DAC. We also need to measure the amplitude of the combined output signal, and normalize this signal if the amplitude is too large. Then the signals are adjusted back to the original offset voltage of the guitar/bass signal, where we check to confirm the voltage value is safe to enter other pedals or the amplifiers of the musician's choice.

To test the pedal mat, we need to monitor the communication process between the momentary push switches and the perceived state of the pedal mat through the microcontroller. If the section of the pedal mat is set to an on/off state, the microcontroller needs to switch the state with every button press. When the section of the pedal mat is in an impulse state the microcontroller needs to confirm the number of button presses. After this the microcontroller needs to communicate back to the microcontroller in the pedal the adjustments, which will ensure are adjusting the correct values.

5 Results

We have found that combining the frequencies at the same time can lead to a much higher amplitude, which can cause clipping in a given speaker. To counteract this effect, we need to make sure that we normalize the combined output signal by taking the summed signals and dividing by the maximum absolute value of the combined frequencies (6). This will ensure that the maximum amplitude is set to 1.

Another experiment we attempted was using the Teensy. We first attempted to use the teensy 3.0 to measure the frequency of a sinusoid. We used the audio libraries that were provided with the teensy library. We used an oscilloscope and a frequency generator to test and verify. We tested frequencies between 500hz - 1000hz with a signal range of 0 to 1.2V. We set the test up to write serial messages to the computer to tell what frequency was found. Our results were the following:

- Serial messages were verified to be working correctly
- The computer did not receive any messages from the controller, so we assumed the signal was not measured.
- We were not able to verify why the teensy was not able to measure the frequency.
- We decided to move to the teensy 3.6 in the event that it is a hardware issues with the Teensy 3.0.

Once we moved to the Teensy 3.6 with the Audio Adapter, we were able to get clean 16-bit audio at a sampling rate of 44.1 kHz for input and output. Therefore, there are no more problems with the frequency range that we were having trouble with earlier.

The Teensy Audio library also had successful tests. Bit crusher, reverb and mixing were all able to be used for input audio and controlled with knobs and buttons and output to a headphone jack for listening.

6 Conclusions

The goal of our project is to create a unique pedal that takes in a single input and delivers two outputs, one signal that is identical to the original and one signal that is manipulated. This will allow the musician to have more room to influence their signals and to fill the gaps where there might have been missing parts. The pedal mat allows the musician to have more control over our pedal in a live setting, allowing them to change any feature of our pedal wirelessly from a convenient mat.

We believe the best plan of attack is to manipulate the signal digitally using the Teensy 3.6 microcontroller. We can also adjust the amplitude of the output signal to ensure that we are not clipping the speakers of the amplifier. Since we are using a mostly digital design we should be able to control the features of the pedal wirelessly from the pedal mat, which can allow adjustments from the communication between microcontrollers.

The solution we have chosen allows the user the most flexibility when using our pedal and pedal mat, and would work well together as a digital coupling rather than a mix and match of analog adjustments. The programming aspect allows us to change multiple assets of our design very quickly, making fine adjustments with ease and adding more features down the road. Our pedal and mat will revolutionize the way musicians can control their sound on stage.

7 References

1. http://www.generalguitargadgets.com/pdf/ggg_gro_sc.pdf
2. https://www.sparkfun.com/products/14057
3. https://www.pjrc.com/store/teensy3_audio.html
4. https://www.pjrc.com/teensy/td_libs_Audio.html
5. https://learn.sparkfun.com/tutorials/shift-registers
6. http://www.simonheather.co.uk/pages/articles/science_hearing.pdf

8 Appendix

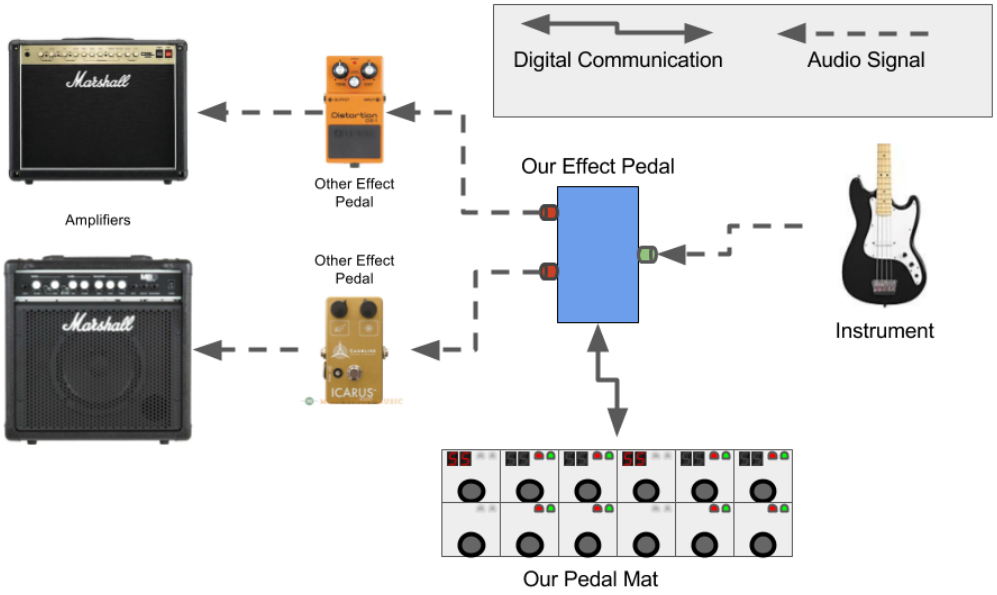


Figure 1: Pedal and Pedal Mat Flow Chart