Tremolo Guitar Effect Pedal

Product Design Specification

AJ Trantham
Version 1.0

Table of Contents

1 Introduction

1.1 Product Purpose

The goal of this project is to create a tremolo guitar effect pedal prototype using the STM32F411RE microcontroller. The tremolo effect is a rhythmic low frequency amplitude modulation of the guitar signal. This device is intended to be placed between the guitar and amplifier where it will apply the effect to the guitar signal.

2 General Overview and Design Decisions

2.1 Overview

Opposed to harmonic tremolo which alters the volume of a particular frequency range, the classic tremolo effect (the focus of this project) simply attenuates the guitar signal at a specific frequency. Both the level of attenuation and rate of attenuation are controlled by user input potentiometers. The tremolo effect is to audio what a strobe light is to light. When engaged, the tremolo effect gives the guitar a pulsating sound.

Tremolo is an old sound which has always been available as a playing technique on instruments with a bow, like violins or cellos. While available on some early electronic pianos via rotating fins or opening and closing a sound port, the first electronic tremolo wasn't available to guitarists until 1941. Note there is no way to make the tremolo sound on a guitar without the aid of electronics and amplification. The DeArmond company developed a tremolo for guitar that utilized a water-based electrolytic fluid that when shaken by a motor would splash onto a pin attached to the positive connection on the guitar cable shorting it. Periodically shorting the hot pin of the signal cable resulted in the volume-wavering tremolo effect. This may have been the first guitar effect pedal ever made as it (like modern effect pedals) was designed to sit between the guitar and the amplifier.

The effect was popularized in the 60s when it was included in mainstream amplifiers. While originally implemented with tubes and variation of bias voltages, Fender's 60's blackface amps used a photocell to oscillate the tube voltage. Now it is common to have optical based tremolo effects and the tremolo effect here also uses an optical design approach. The tremolo's signal attenuation is accomplished by a Photo FET Optocoupler. Instead of using vacuum tubes, this tremolo effect uses the STM32F411RE (STM32) microcontroller to control the voltage oscillation across the optocoupler. Managing the internal IRED of the Optocoupler is the primary purpose of the STM32. The STM32 also reads the user input potentiometers and generates the wave forms which are sent to the optocoupler. This use of the optocoupler for attenuation allows the entire guitar signal path to remain analog. This benefit means no signal resolution is lost due to the digital conversion process.

As is typical of guitar pedals the tremolo effect operates off of 9V DC. It is equipped with mono ¼" input and output jacks. Alongside the Rate and Depth input potentiometers which control the modulation frequency and level, the device has an Intensity potentiometer dedicated to controlling the gain of the effect. The human ear perceives the modulated signal as quieter since some of it is missing. (This is similar to the way the human eye sees a PWM LED as dimmer when PWM is at a lower percentage). The Intensity input's purpose is to allow the user to increase the gain to make up for this perceived volume loss.

2.2 Assumptions and Constraints

Major assumptions include: i) the user will supply 9V power supply, ii) the user will connect guitar via ¼” input/output jacks, and iii) (at the moment) the user will power the STM32 via computer. Constraints of this device are related to the range of the effect parameters. The Rate potentiometer sets the modulation frequency between 3 Hz and 10Hz. The Depth potentiometer sets the level of attenuation which ranges from 0% to 100% attenuation.

3 Architecture Design

3.1 High-level Overview

Figure 1 shows the high level overview of the tremolo effect pedal. Intensity, Depth, and Rate are user input potentiometers.

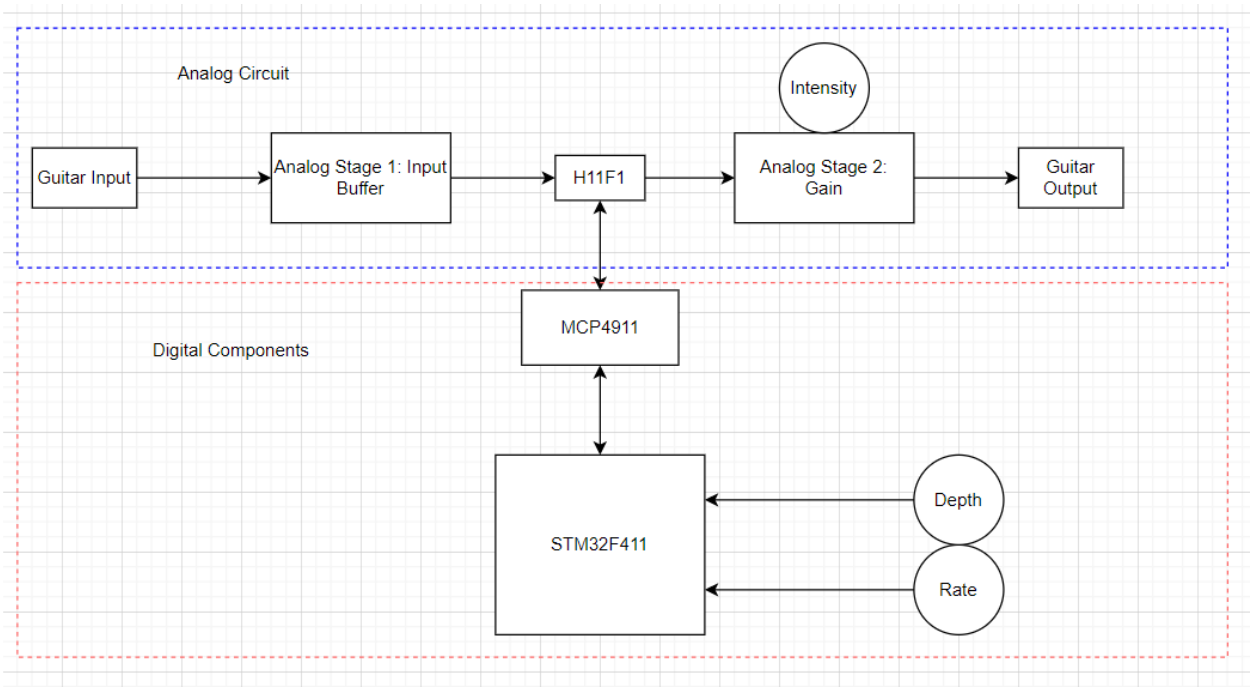

Figure 1: High-level Overview

The device is comprised of two sections: the analog and digital. The analog section is responsible for reading user inputs Depth, Rate and Wave Shape Selection (B1 on STM32) and then using these values to construct the appropriate wave form to send to the Octocoupler.

3.2 Hardware Architecture

Figure 2 gives a more detailed look at the inner workings of the digital circuit. STM32’s primary role is to adjust the voltage across the H11F1 Optocoupler. The MCP4911 lends a hand here since the STM32 does not have an onboard DAC.

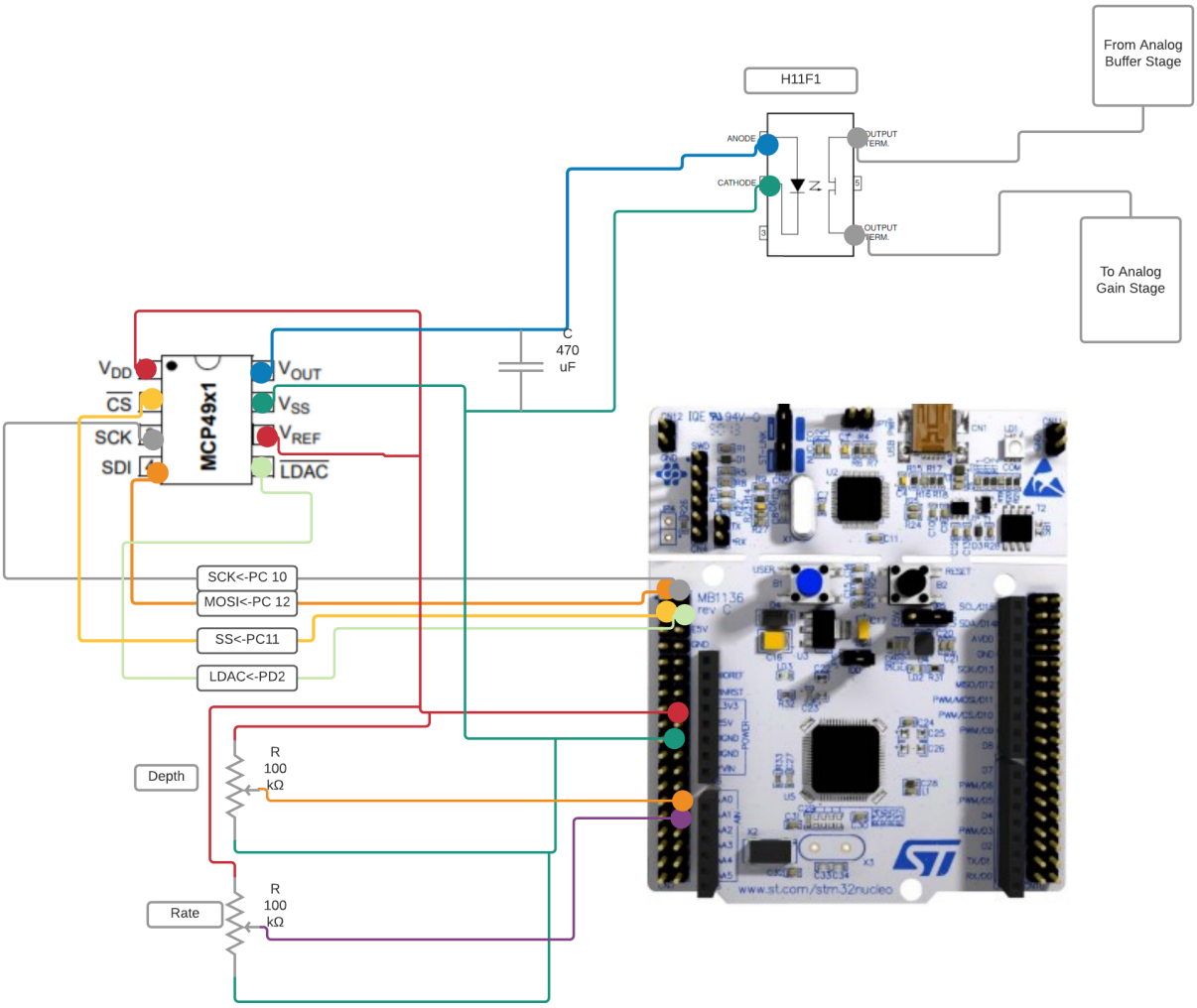

Figure 2: Digital Circuit

It is important to note that the 470 uF Capacitor C was necessary to remove a popping sound that would otherwise occur when the microcontroller went from high to low. This popping sound was especially apparent when the shape was set to a square wave. The abrupt on off sound was unbearable. The value of 470 uF was arrived at by experimenting. Any bigger capacitance seemed to smooth out the waveform to where it was difficult to tell them apart. A smaller capacitance allowed the popping sound when the amp was set to a high gain.

Figure 3 shows the analog audio circuit in more detail. The first op amp serves as a buffer stage with no gain. Both op amps are TL072CP from Texas Instruments. These are designed to be extremely transparent. This is important because the tremolo should not add any distortion or clipping to the guitar signal.

The second stage does provide a variable amount of transparent gain. This gain potentiometer (Intensity) allows to compensate for the perceived volume loss of the periodic attenuation.

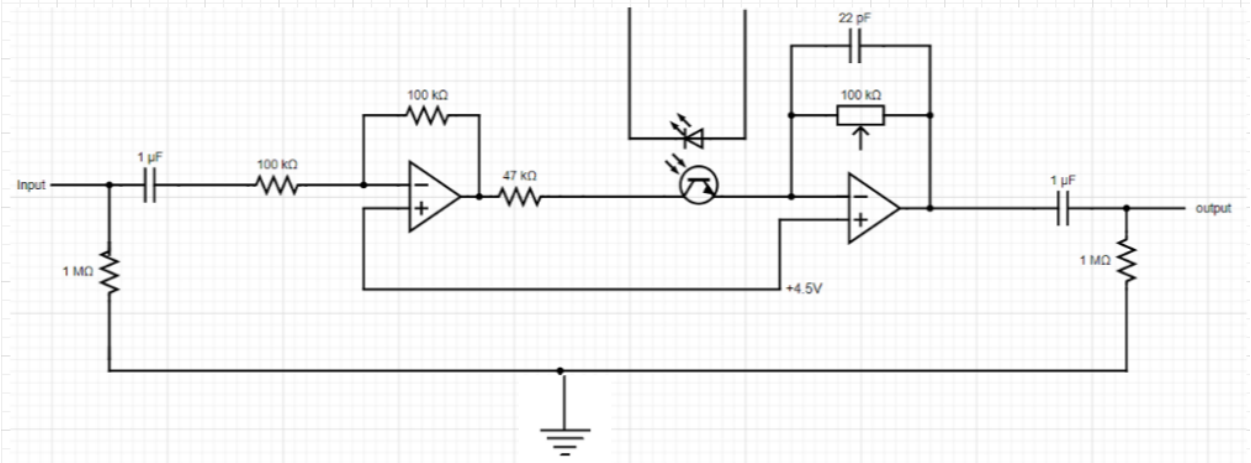

Figure 3: Analog Circuit

3.3 Software Architecture

Timers

As this is a timing-based effect, the interrupt service routines are the most critical component of the tremolo firmware. In particular TIM2 is used to control the rate of collection of the Rate and Depth parameters and the SysTick timer is used to control the rate at which samples are sent to the Optocoupler. TIM2 is set to fire at 100ms. The SysTick fires at the Rate parameter.

ADC Acquisition

The ADC is responsible for reading the Rate and Depth potentiometers. Rate is connected to PA1 and Depth is connected to PA0. When TIM2 fires an ADC read is done on PA1 and the Rate parameter is updated in the code and an ADC read is done on PA0 and the depth parameter is updated in the code. Figure 4 shows the code for updating the rate. update_depth() is very similar. Notice that each time the rate is updated the systick timing is also updated so the generated wave has a period matching the new rate.

```
309 /** Reads Rate Pot PA1 and updates rate global variable */
310 void update_rate(void) {
311     ADC1->SQR3 = 1;                 /* conversion sequence starts at ch 1
312     ADC1->CR2 |= 1;                 /* enable ADC1 */
313     float pot_perc = read_pot_percent();
314     rate = (pot_perc * (MAXRATE - MINRATE)) + MINRATE;
315     // reset the period of our wave
316     wavepoint_time_space = rate / num_wavepoints_per_cycle; // in ms
317     set_sysTick_interrupt(wavepoint_time_space); // reset the wave period
318 }
```

Figure 4: update_rate() implementation

SPI DAC Communication

As mentioned in the Hardware section the MCP4911 DAC translates the digital values from the STM32 into a voltage value over the Optocoupler. The MCP4911 uses the SPI communication protocol to receive these values. An important note here is that the STM32 board needs PA4 to be set to the NSS alternate function and be tied high for the SPI3 module to work. When writing to the DAC one needs to send the data to the MCP4911 and then bring the LDAC pin Low. This tells the DAC to begin the conversion.

Software Wave Generation

Wave generation is the key software feature and is the heart of the effect. The wave we are generating controls the amount of resistance provided by the optocoupler. Providing max voltage to the optocoupler sets a very small resistance and the guitar signal passes freely. Sending a small signal to the optocoupler, gives it a large resistance heavily attenuating the guitar signal. The wave amplitude ranges from 3.3 volts (small resistance, no attenuation) to Depth (a value between 3.3V and 0V). When Depth is 0 the signal is completely attenuated at the min value of the waveform. This concept is illustrated below in Figure 5.

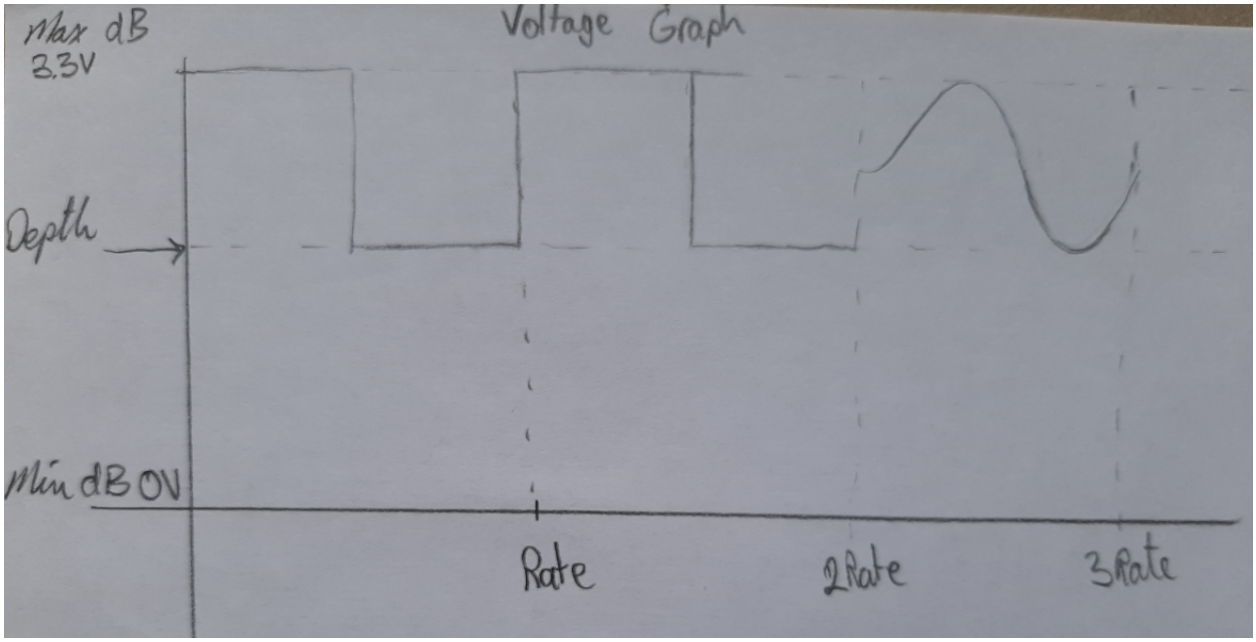

Figure 5: Optocoupler Voltage based on Rate and Depth Parameters

Figure 6 shows the code that generates the waveform based on the Depth and Rate parameters. This happens inside the SysTick_handler. Also, note that `current_wave` is a 2-d array with 50 samples per period of each of the four waveforms.

```
114     float wave_val = current_wave[wave_index][sample_index++];
115
116     // apply depth control to base wave value
117     int digital_attenuation_range = MINDEPTH - depth;
118     short scaled_wave_point = (digital_attenuation_range * wave_val) + depth;
119
120     DAC_write(scaled_wave_point);
```

Figure 6: Creating the waveform based on Rate and Depth Parameters

3.4 Power Considerations

Most all guitar pedals are 9V DC and expect an external power supply or run off a 9V battery. Pedal boards generally contain a power supply unit that generates isolated power supply voltage to each individual pedal. With this precedent in mind, it is expected that this tremolo device will also run of an external 9V DC power supply or 9V battery. Because of this power is not a chief concern and at this point I have done no power optimization. However, this is something that I would like to explore further. Since my fastest timer is only 100ms, there is likely an opportunity to slow down the clock.

3.5 Key Components

The key hardware components with linked data sheets are:

- STM32F411xE Datasheet - microcontroller

- TL07xx Low Noise FET-Input Operational Amplifier - op amps for analog circuit
- H11F1 Photo FET Optocouplers - optocoupler for volume modulation
- MCP4911 10-bit Digital-to-Analog Converter with SPI interface -DAC for waveform voltage

4 System Design

4.1User Interface

Currently, the user interface is composed of the Rate, Depth, and Intensity potentiometers and the STM32's B1 button. When taking this from a prototype to a finished product, it will be necessary to acquire an enclosure with labeled potentiometers. Figure 7 shows the circuit and interface as it is at the moment.

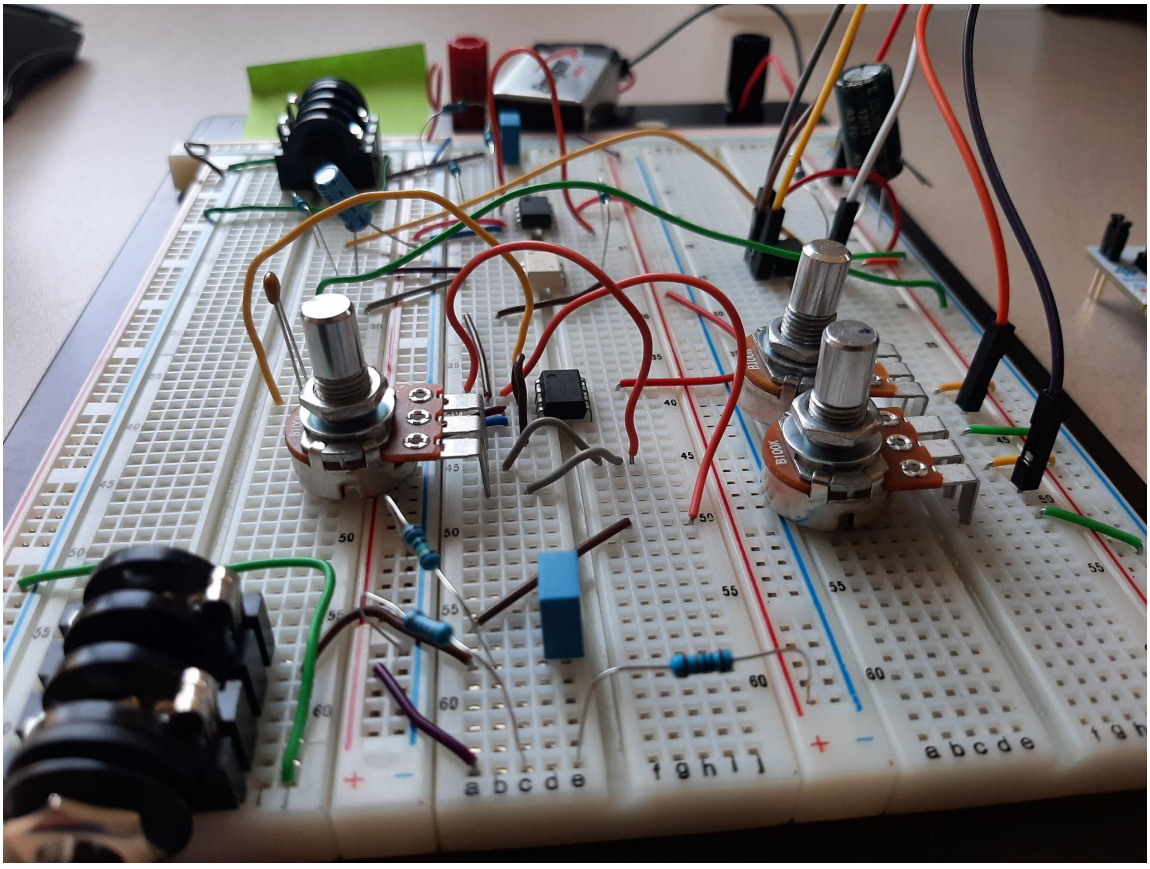

Figure 7: The User Interface

4.2 Sound Manipulation Goals and Results

A goal of this project was to use the STM32 to generate a variety of waveform shapes and as a result a variety of sounds. Currently, the tremolo effect supports sine, square, triangle and sawtooth waveforms. The sine wave has a natural sweeping sound. The square waveform is the most abrupt tremolo sound as the signal oscillates between max voltage and the depth voltage. The triangle waveform sounds much like the sine wave. It is honestly difficult to tell

them apart. The sawtooth wave is a subtle tremolo effect since it rams from depth level signal attenuation back to full volume. In the future I hope to implement different waveforms and combinations of the four that are currently implemented.

4.3 Test Procedures

This project was interesting to test as it required some objective hardware and software testing and also some listening testing. I used usart communication and an oscilloscope to test the wave shape generation and timing. This allowed me to ensure the user parameters Rate and Depth were correctly influencing the wave generation. Then I tested it with my guitar and amp to ensure it actually sounded good. It was during these test that I discovered the popping sound.

4.4 Usage

Since tremolo affects the overall volume it makes sense for this to be the last effect in your signal chain. At a minimum it should likely be towards the end of your signal chain. To use a tremolo well timing is key. Tremolo is musically interesting when it is in time and is distracting when it is not. This is why tap tempo would be such a useful feature to add.

4.5 Bill of Materials (BOM)

The current Bill of Materials is shown below in table 1. If I add the additional features I would like, transfer it to a PCB and get an enclosure for it, the cost will increase of course.

Component	Quantity	Vendor	Unit Price (USD)	Total Price (USD)
1/4" Horizontal PCB Switching Mono Jack	2	GuitarPedalParts.com	0.85	1.7
B100K 16mm Pot, RA PCB Pins	3	GuitarPedalParts.com	0.55	1.65
TL072CP Op Amp	2	GuitarPedalParts.com	0.75	1.5
470 uF Capacitor	1	Reuseum Boise	0.5	0.5
1 uF Box Capacitor	2	GuitarPedalParts.com	0.5	1
1M 1% Tollarance Metal Film Resistor	2	digikey	0.72	1.44
100k 1% Tollarance Metal Film Resistor	2	digikey	0.72	1.44
47k 1% Tollarance Metal Film Resistor	1	digikey	0.26	0.26
22pF Ceramic Disic Capacitor	1	GuitarPedalParts.com	0.2	0.2
H11F1VM-ND Optocoupler	1	digikey	3.5	3.5

MCP4911 DAC	1		1.52	1.52
STM32F411RE	1	Mouser	13.55	13.55
Total				28.26

5 Conclusion

Appendix A: References/Resources

Informational References

Name	Link
Reference Project - Coda Effect	https://www.coda-effects.com/2016/04/relay-bypass-conception-and-relay.html
Tap Tempo example implementation - Coda Effects	https://www.coda-effects.com/2016/05/tap-tempo-tremolo-diy-complex-project.html
Tremolo Effect Explanatory Video	https://www.youtube.com/watch?v=ulrxiLpvfGw&t=1s

Table 1: Bill of Materials

Appendix B: Key Terms

Term	Definition
STM32	STM32F411RE microcontroller
Tremolo	An musical effect where the instrument's volume is periodically attenuated
ADC	Analog to digital Converter
SPI	Serial Peripheral Interface - communication protocol