Wah Pedal Project

Eric Walden & CJ Tulowiecki



# Objectives

The objective of this project is to design and build a digital “wah” pedal effect that can be implemented into a guitar audio processing chain. A “wah” pedal consists of a filter that sweeps its cutoff frequency in response to the movement of the expression pedal by the user. A potentiometer will be used to sense the position of the expression pedal. Different filter models can be used that provide different sound characteristics.

# Summary

The project began with the purchase of a broken Dunlop Cry Baby “wah” pedal. This is an old analog pedal that had become the gold standard for “wah” pedals over time and are still widely used. However the pedal was bought simply because it could provide us with a completely assembled pedal enclosure for our project. The analog circuitry (aside from the potentiometer and kill switch) was ripped out so that it could be replaced by embedded processing. A Teensy 3.2 with additional audio board extension provided the necessary processing power and audio fidelity within the given size constraint. Code was developed in Arduino to provide the “wah” effect. The Teensy’s IO was then connected to the potentiometer and kill switch so that the “wah” effect could be modulated or turned off completely. At this point the Teensy’s power regulator blew and we had to use an external regulator to power the board. The embedded system was installed inside of the pedal enclosure. From there experimentation with the filter sweep and resonance characteristics was performed to find the best sounding “wah” effect.

# Hardware

Design

The hardware was designed to be easily assembled while providing impressive performance and feel. The purchase of a cry baby “wah” pedal enclosure took care of the external analog hardware needed to complete the project such as a potentiometer and kill switch. The Teensy audio board extension comes with 16 bit 44.1 kHz audio inputs and outputs with line level amplifiers which provided the board with enough fidelity and power to make a clean sounding effect. The Teensy board was small yet powerful enough to perform the desired filtering.

# Software

Platform

The Teensy 3.2 only supports Arduino, so this became the software platform for the project. However, Arduino had advantages such as quick prototyping and intuitive data communication. Interfacing between the Teensy and the audio board

# Test Results

# Lessons Learned

Valuable lessons were learned throughout the course of the project. One of which was that in order to keep project cost down you must research and determine how much processing power you need. While choosing the correct processor for the application, many professional audio DSP chips popped up that could easily perform the functions we required of it. However these DSP chips cost at least one hundred dollars which was not an attractive price point. The Teensy was really the perfect solution for our needs as it provided high quality minimal audio processing at a low cost.

Another lesson that was learned was that finding a used item that has many of the parts you need already assembled together is much better than finding each part individually, buying them new, then assembling them. The broken Cry Baby pedal proved to be of much higher quality than we could have hoped for in a generic pedal enclosure. It also cost much less.

Proper use of the repository became important as the project’s material continued to grow. Code management became much easier and efficiency was increased.

Another lesson that is fabrication related is do not solder a board while the power is on. This mistake was made and contributed to the destruction of the Teensy’s power regulator. If soldering on a powered board is required, ensure the board is first powered off.

# Final Notes