Multi-Effect Guitar Pedal

COSC 4342 – Computer Networks

By: Noah Peña

Abstract

The Multi-Effect Guitar Pedal is a device and mobile application combination that allows for a user to mimic basic effects and their settings through a mobile app instead of switching between different guitar pedals. The pedal uses the Intel Edison System-on-a-Chip board to play the audio effects while also taking Bluetooth input from an Android application. The Android application has a variety of settings for users to choose from on what kind of effects they would like added to their guitar. This system allows for a lot of future modifications and the ability to add new features and new effects.

Introduction

I have a lot of friends who are guitar players and one thing that all of them seem to have mixed opinions on are guitar pedals. Some guitarists think that if you’re using 2 or more pedals, then you’re making your song to mucky, while others think that having a lot of effects can really expand your creativity and sound. For those that use a lot of guitar pedals, a guitar pedal board is often used. A guitar pedal board has a multitude of different pedals on them, and are selectable by either switching the plugs, or by using a large switch to change input and output channels. Either way, this tends to be a large hassle, and makes it difficult to use multiple effects in the same song. I thought of this idea to act as a compromise between the two. This guitar pedal is only a singular pedal, but is capable of having multiple effects being processed through it.

Implementation

The underlying technology behind this guitar pedal is Bluetooth Low Energy or BLE. BLE is a form of Bluetooth, but operates at a smaller scale than Bluetooth Classic does. Bluetooth Classic maintains a constant connection between the Host and Peripheral device, while BLE sends short bursts of data between each device. Due to this, BLE can be implemented on small embedded devices and can be more efficient with battery usage. The device that does all of the processing on the guitar pedal is the Intel Edison System-on-a-Chip board. The Intel Edison has a 500 MHz Dual Core Intel Atom x86 processor, with 1GB of RAM and 4GB of flash memory. For processing the audio, a program called Sound eXchange (SoX) is utilized. The Edison also has a Bluetooth 4.0 + BLE card onboard. A node.js script was utilizing the “Bleno” library was used to control how data was sent and received via BLE.

On the Mobile App side, Android was picked to create the application in. Since the project uses BLE, an iOS application can also be developed with equal features. The Android application was programmed natively in Java with no external libraries aside from the libraries that are provided by Google. The Android app has an interface for users to select base effects and send them to the guitar pedal for playing. Users have access to 10 base effects, but are also able to chain those base effects together in order to make their own custom effects. Up to three base effects can be chained together with varying properties on each of them. Effects can be saved on to the user’s phone and can be traded with other users who own the application.

Testing

The two components that were testing the most, were the user interface on the Android application and the Bluetooth on the guitar pedal. The Android applications user interface was redesigned many times to make sure that users would have a very easy time creating their own effects. The user interface uses a variety of different views and so bug fixing was a large portion of them. Testing the user interface on multiple screen sizes was also a very big issue. I tried not to use images so that it would scale better on different screen sizes and so it could work well on lower end phones.

Testing Bluetooth was also very important as it is critical that there is very low latency in sending commands to the pedal. To minimize the processing on the guitar pedal side, commands were created and processed on the application side, and then sent as raw commands through BLE. The Intel Edison would then take these commands and directly execute them.

Use Cases

The main use case for this project are for musicians that want to minimize the amount of equipment they need to bring to play at different shows. Often times guitarists will have to bring multiple pedals for their set list and then have to swap pedals in-between songs and even during songs sometimes. This is an elegant solution in that it makes it easier to travel with as you only need one pedal, and the phone interface makes it easy to manage what affects you need to play next.

Future Work

The biggest thing to work on in the future is an iOS application. The majority of users in the United States use iOS devices, so having it work for iOS is an absolute must. There aren’t any Android specific features, so the port for iOS should be relatively equal to that of the Android application.

The next planned feature is to ditch the Intel Edison. The Intel Edison is a board with very great hardware, but Intel significantly dropped the ball on the software side. Bluetooth itself won’t work out of the box, unless you run extra scripts on startup. Even more so, is that the Edison is incapable of running Bluetooth Classic and BLE at the same time. This makes it difficult to be usable for a wide range of users. The software repositories for the Edison are very barren, and so most programs needed for the Edison have to be cross-compiled. For an end-product this is fine, but for prototyping this quickly becomes annoying and time consuming. Lastly, the Edison is a very costly board, with its main draw is it’s x86 architecture and small size. Unfortunately, this project doesn’t require either of those and so it feels very lackluster in this project. I originally chose the Edison because I won two of them in a contest, so I already had them. If I had to choose what board I want to use now, it would be the BeagleBoneGreen. It has the same features of the Edison, but with a high clock speed, better Bluetooth Module, lower cost, and better software support.