Kyle Marino
5/7/21
EE250

EE250 Final Project Writeup

1. **Project Description**
   The purpose of this project is to be an IoT recording device for electric guitars. It works by recording the output signal of the guitar with an ADC, and then transmitting the measured samples over UDP to a server to be processed. On the server, the time domain and frequency domain waveforms are graphed in real time as they come in from the raspberry pi. Using the frequency information generated by the fourier transform, the server determines what note is being played during the interval of that packet. The server can record what notes are being played, and then save that into a file. That file can later be replayed to hear a reconstructed audio of what was originally played.

2. **Key Questions**
   Motivation: The motivation behind this project was to make a device that could analyze and record electric guitar music. When I was starting with electric guitar, I couldn't find a simple device that would allow me to interface with my computer and record. The device I ended up purchasing was rather expensive (around $200). What I have created is a simpler and cheaper alternative that records the guitar output. This solves the problem that I initially faced of not having a low budget way of recording guitar music on my computer.

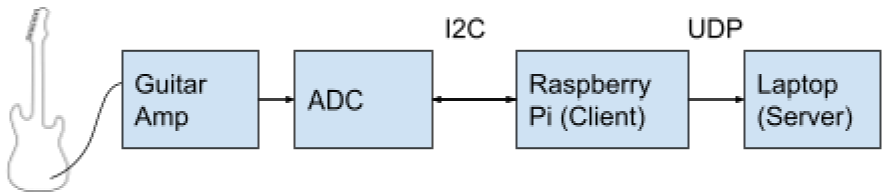
   Ethics: This application is rather self contained and doesn't pose any immediate ethical concerns. A possible future ethical issue could be with the production of this device. Some modern products are unethically produced in factories with poor work conditions and compensation. Another potential ethical issue would be with selling the data that is acquired with this. Since this doesn't contain any confidential user information, the possible sale of the data generated doesn't pose a huge ethical problem.

   Economics: The cost of this system would be around $20 of components. A cheaper Raspberry Pi such as the Pi Zero W ($10) could be used and the ADC I used also cost $10. There would also be manufacturing costs, so my estimate of the total cost to produce would be around $30. Since this product is used by consumers, they would be the ones paying for it.

   Data Analysis: The data in this system can be analyzed for note frequency, note duration, and note amplitude. In addition, the song speed and genre could be analyzed. It could also be possible to analyze the note accuracy, rhythm accuracy, and intonation accuracy of the player. I am only currently analyzing the frequency and duration of the notes. This could be used to control a light system to be used at clubs or dances if live music is being performed.

Self Directed Learning: I had to learn how to create live updating graphs with matplotlib, and how to play audio in python. I found documentation and examples online on the libraries' respective websites. I also had difficulty with reading the keyboard input, and found information on how to solve the issues I was having on stack overflow. Another thing I learned was that the packet size of a UDP packet does not affect the packet speed. I read some articles online, and ended up going with larger packets since a lot of data needed to be transferred.

3. **Block Diagram**



4. **Description of components, platforms, protocols used, and processing/visualization techniques**
There are four components in this signal chain. The guitar output goes directly to a guitar amp which has a headphone audio output. The headphone audio output then goes into a differential ADC (ADS1015), which I bought off of adafruit. This ADC has a programmable gain amplifier, differential inputs, and a max output data rate of 3300 hz. This ADC is read by a Raspberry Pi, which then sends the raw data to a server over UDP. To reduce UDP traffic, I use an absolute average to determine if there is any output signal from the guitar. If no signal is detected, then I do not send any packets. Each packet sent is 100 ms of samples, so that the server doesn't get backlogged with unresolved UDP packets. On the server, I perform an FFT on the sample data, and then plot the time domain and frequency domain waveforms. The graphs displayed are updated on every new packet. Using the frequency data from the FFT, I determine the note played by finding the frequency of the largest spike in the packet. I then determine the note by finding which note has a frequency closest to the one measured. By measuring how many packets a note persists, I can also determine a rough estimate of the note duration. This information is pickled into a file, which serves as the recording of the guitar music. This file can be read by another python script, which plays a sine wave according to the frequency and duration.

5. **Reflection, discussion of limitations that demonstrates insights to their cause and possible remediation, lessons learned, future directions**
Some of the limitations of this project were with the sampling speed of the ADC, the transfer speed over UDP, and the processing speed of python. I initially tried using the ADC on the GrovePi shield, but I found that the sampling speed was too slow to resolve any notes. I found an ADC on Adafruit which could go up to 3300 hz which has enough speed to resolve the highest notes on a guitar. However, I found that I was only able to read the ADC at 2400 hz on the raspberry pi. This still worked as it only cut off the 5 highest notes on my guitar. Another issue was the delay over UDP, I measured an average of 50 ms for a single packet, but when sending many packets for a song I

noticed that the server would get backlogged and it would take around a minute for the rest of the packets to go through. A more direct connection, such as bluetooth, that wouldn't have to go through the router would be a better solution for this. Another issue that I ran into was that the notes on a guitar are comprised of several different frequencies, called harmonics. These harmonics made it difficult to determine the notes if multiple notes were played at the same time, so I just limited the project to one note at a time. A future version of this project would filter out these harmonic frequencies and use a more advanced algorithm to determine what notes are being played. A future direction of this project would be to remove the guitar amp altogether, and use a dedicated component to do the amplification. The PGA on the ADC was not enough to adequately measure the signal, but using a guitar amp slightly defeats the purpose of this project.