**Creating a Musical Note Detector that Translates Sounds into Notes Using a Raspberry Pi**

**Final Report**

**CS 450**

**Spring 2022**

Divya Manirajan

dmanirajan@bellarmine.edu

May 3, 2022

**Executive Summary**

250-word summary of your report.

1. **Project / Problem Introduction**

The reason behind this project is that I believe identifying notes is the most crucial part of any musical process. All instruments are used to make musical notes, these notes can be combined to create songs, and these songs can be played using any instrument. In this process, the ability to identify notes based on sound frequencies allows people to not only visualize any song, but also play any song on any instrument. From this idea, I decided that I wanted to create a generalized musical note detector that can pick up audio frequencies emitted by instruments and produce a visual output that displays both the name and the octave of the musical note that corresponds to that audio frequency.

1. **Methods**
   1. *Project Idea*

The device I developed is a musical note detector that runs on a Raspberry Pi. It can detect and interpret audio frequencies inputted through a USB microphone, and it displays the name and octave of the corresponding musical note on a monitor connected to the Raspberry Pi. I also incorporated an RGB LED that produces a different light for each type of note: sharp, flat, or natural.

A picture containing text, guitar

Description automatically generated

There were two main objectives I wanted to accomplish with this project. The first was that I wanted to create the musical note detector software using Python. The second was that I wanted to implement an electric circuit to integrate an RGB LED that notifies the user if the note detected is sharp flat or natural.

A picture containing text, table

Description automatically generatedA picture containing table, desk

Description automatically generatedA picture containing table, worktable, desk

Description automatically generated

Currently, this device can only be used through running code on the Raspberry Pi which requires some computer science knowledge but continuing this idea could lead to a more widely usable product. If I were to continue this project, I think it would be possible to create a user-friendly GUI that would allow anyone to be able to use this product. I think for the most part, this type of project is targeted towards anyone with an interest in musical instruments, but with the current interface, it can only be accessed by people with some experience in computer science.

* 1. *Tools/Materials*

This project consisted of both a hardware and software portion. The hardware tools I used were a bread board, a set of 1 kilo ohm resistors, a Raspberry Pi 2, an RGB LED, several connection wires, a HDMI wire, a monitor, and a USB microphone. These components were assembled to create the electrical circuit that detects incoming audio frequencies through the microphone and displays the name and octave of the corresponding note through the monitor. It also connects the RGB LED to the Raspberry Pi which is used to signify if the detected note is sharp, flat, or natural.

The software tools I used were Python and Jupyter Notebooks. Python has a library called PyAudio which is used for interpreting audio on a variety of platforms. I was able to use this library throughout my project to transform incoming audio into a specific frequency. I also relied on Jupyter Notebooks as my development environment because it provided an easy-to-use GUI for all python code and the files produced by Jupyter Notebooks can easily be converted into simple python (.py) files which were needed for the Raspberry Pi to run the code.

1. **Results**
   1. *Project Results*

I was able to

The hardware portion of my project was relatively straight forward. I had to order some of the components which took about a week or so to arrive. Once I had all the parts, I began connecting the circuit to the Raspberry Pi, starting with the USB microphone. I was able to verify that the microphone was detecting audio and sending this data to the Raspberry Pi.

In the section, outline the results of your project. This section will include not only narrative but also images (graphs, screenshots, etc. as appropriate). Include various diagrams and software engineering tools that depict how your project fits together.

* 1. *Adherence to Timeline*

During the initial steps of this project, I was able to follow my proposed schedule but once I progressed with my project, I found that certain aspects were taking much longer than I had anticipated. It was easy to estimate the length for most of the hardware aspects of this project, such as, ordering the components and assembling the circuit. In general, the software aspects were much harder to anticipate. Looking back, I also realize that I had originally planned to have every part of this project completed by the Celebration of Student Research Event (including this paper and the Kaltura Video). I did not realize that the event only needed our actual project and the poster completed. This change greatly affected my proposed schedule as it gave me almost a month longer than originally planned. Due to this, many of the items that I had originally planned to accomplish in a week ended up taking two weeks because I had extra time to improve certain parts of my project.

In the future, it would be helpful to keep track of certain deadlines so I can more efficiently use my time. I think it would also be beneficial to give myself more time than I estimate because it is better to be ahead of schedule than behind.

* 1. *Future Work*

I believe that this project has a variety of future applications. Since this device can detect any frequency and display its corresponding note, it could further be modified to act as a tuner for any instrument. Ideally, this project could implement a user-friend GUI that allows for a selection of instruments to choose from, and this choice can modify the note detector to act as a tuner for the selected instrument. For example, if someone needs to tune a guitar, the standard tuning is EADGBE for the six strings. The device could have a guitar setting that notifies the user if the detected frequency of an open string matches any of the predetermined guitar string frequencies, and if not, the device can tell the user whether to tune a string up or down in pitch to match the correct frequency.

Another future application is to use this project to write out the notes of a specific melody in either guitar tablature (tab) or sheet music form. These modifications rely solely on changing the output to match a specific form. The internal calculations for the frequencies and note names and octaves would stay the same. This change would essentially act as a speech-to-text function for music which is an interesting concept. If I were to continue this project, I would like to add this feature because this could be used by a variety of people.

1. **Conclusion**

Wrap up the report (remember, do not introduce any new material here).

**References**

<https://www.seventhstring.com/resources/notefrequencies.html>​

<https://www.instructables.com/Raspberry-Pi-Guitar-Tuner/>​

<https://www.benchodroff.com/2017/02/18/using-a-raspberry-pi-with-a-microphone-to-hear-an-audio-alarm-using-fft-in-python/>​  
<https://github.com/katrinamo/RPiPitch/blob/master/freqDetect.py>​  
<https://www.instructables.com/Using-a-RPi-to-Control-an-RGB-LED/>