**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**

The goal of this project was to create a musical note detector using a Raspberry Pi connected to a USB microphone. This project had two parts, a hardware and software portion. The hardware portion consisted of an electrical circuit that connected a USB microphone, a HDMI monitor, and an RGB LED to a Raspberry Pi. The USB microphone was used to detect external audio signals that the Raspberry Pi would interpret. The monitor was used to display both the name and the octave of the corresponding note for each detected frequency. Lastly, the RGB LED was used to notify the user if the note detected was sharp, flat, or natural. The software was written in Python and executed on the Raspberry Pi.

This device has a variety of future applications in the music field. Modifications to this program could lead to tuners for any instruments or tools to visualize a melody through either guitar tabbing or writing sheet music. This note detector software could further be modified to better suit a specific purpose. Additionally, adding a user-friendly GUI would make this application functional for a larger population.

1. **Project / Problem Introduction**

I chose to create a musical note detector using a Raspberry Pi because I believe identifying notes is the most crucial part of any musical process. Instruments are used to play musical notes which when combined make a song. These songs can then 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 (Light Emitting Diode) 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 creating a musical note detector software using Python. The second was implementing an electric circuit to integrate an RGB LED that notifies the user if the note being 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. However, 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. This type of project is targeted towards people 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 breadboard, 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 an 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*

Using Python and Jupyter Notebooks I was able to create a note detector program that interprets incoming sounds into note names and octaves. The first step was to utilize the PyAudio library to detect the frequency of input audio. During my initial research, I found some online sources that I was able to implement and modify to better suit my project. In these sources, I found that Benjamin Chodroff, a Software Engineer, had previously used PyAudio to hear an audio alarm using FFT in python. In this project, he used Python to interpret sound into frequencies. I was able to adapt and modify this code in my project so that it could use the detected frequencies to identify which note and on which octave the frequency corresponds to. Once I created this program on Jupyter Notebooks, I converted this file into a Python file to eventually run on the Raspberry Pi.

Simultaneously, I was creating the hardware circuit that connects the Raspberry Pi to the RGB LED using a breadboard, resistors, and connecting wires. Once I completed this hardware component, I uploaded my note detector program onto the Raspberry Pi using a USB. I then had to verify that the program was executed properly. This step took some trial and error as I had to download some external packages for the Raspberry Pi to properly run the program. Once I verified the note detector program executed properly, I further modified it to incorporate the LED.

* 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, 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 use my time more efficiently. 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*

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**

This project allowed me to explore both the hardware and software components needed to create an application. I was able to incorporate an electrical circuit that utilized an RGB LED, while also focusing on Python development using Jupyter Notebooks. Throughout this project I not only learned how to create an application within a certain time limit, but also how to better schedule my time to fit any deadlines. This project set up a foundation for a variety of future applications which would be interesting to further investigate. The idea that this note detecting software could further be used to act as a tuner for any instrument or as a tool to translate sounds into visual notes via guitar tabs or sheet music is exciting.

**References**

Biyoknateesuwan, & Instructables. (2017, September 19). Raspberry pi guitar tuner. Instructables. Retrieved May 2, 2022, from https://www.instructables.com/Raspberry-Pi-Guitar-Tuner/

Chodroff, B. (2020, February 29). Using a raspberry pi with a microphone to hear an audio alarm using FFT in python. benchodroff.com. Retrieved May 2, 2022, from https://www.benchodroff.com/2017/02/18/using-a-raspberry-pi-with-a-microphone-to-hear-an-audio-alarm-using-fft-in-python/

Katrinamo. (n.d.). Katrinamo/rpipitch: Pitch detector. GitHub. Retrieved May 2, 2022, from https://github.com/katrinamo/RPiPitch

Michaelkouznetsov, & Instructables. (2017, October 3). Using an RPI to control an RGB led. Instructables. Retrieved May 2, 2022, from https://www.instructables.com/Using-a-RPi-to-Control-an-RGB-LED/

Note frequencies. (n.d.). Retrieved May 2, 2022, from https://www.seventhstring.com/resources/notefrequencies.html