

Arduino LED Sound Simulation

Zachary Vanscoit
zachary.vanscoit@siu.edu

Abstract—Programmable LED Lights that react to the sound of fire alarms to assist the hearing impaired with older non-flash alarm types.

I. INTRODUCTION

Young individuals who suffer from hearing impairments, either by birth defects or through injury, never experience auditory stimulation or never get to experience it again. These individuals are also at risk if they are ever found in a dangerous situation with a fire but cannot see the fire immediately.

However, having a visual alarm would be more effective at alerting the individual in question. This type of system would also have the potential at providing these individuals with a different type of stimulation from auditory responses as being able to deliver a visual stimulation.

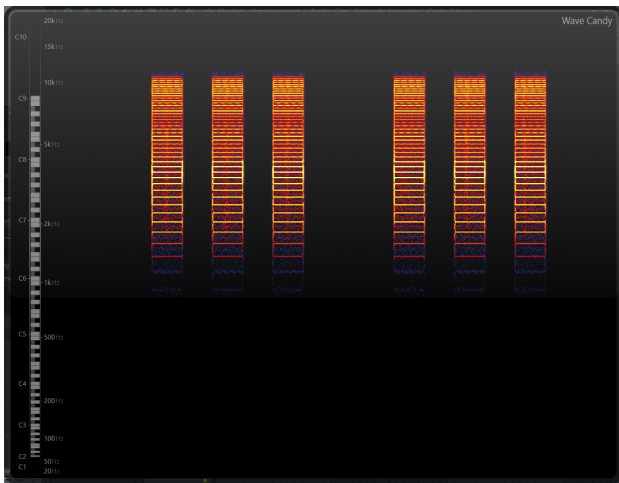


Fig. 1: Edwards-Integrity-Code-3 alarm frequency spectrum.

II. PHYSICAL SETUP

Choosing the type of LED lights was easy to determine as any brand of Digital LED light would be appropriate. I had chosen Adafruit NeoPixel Digital LED strip with 30 LED lights per 1 meter strand. This strand seemed to have the largest amount of community support behind it so finding resources on how to interact with these lights wouldn't be difficult as I have never used an Arduino before this project.

I chose an Arduino for the board to control the lights over as it also was the most community friendly board on the market which has a lot of support for it too. This microcontroller runs off C++ for its programming language which is a language I have never programmed in before the project.

Approaching the first problem of the project comes with the need to use a sensor to connect to the Arduino. The

sensor would be a sound sensor which takes sound from the surrounding environment, and converting the sound waves into usable data.

The Arduino accepts voltages between 0V and 5V, but the mono 3.5mm jack used to connect the microphone to has an output range between 10mV to 100mV AC meaning that it would damage the microcontroller board. If I had plugged it straight in as it would be an AC to DC problem that would need to be solved.

This means that signal would need to be amplified to around 2V and shift the current 2.5V adding a DC bias to the signal as to not damage the board and to avoid any clipping of a raw audio signal.

However, the parts I ordered are still trying to make their way to me from China as they are the cheapest I could find on the market. So instead a "Microphone Controller Module" was used but the microphone quality was almost non-existent as it would fall off completely ignoring any sound past 2 feet away from the sensor.

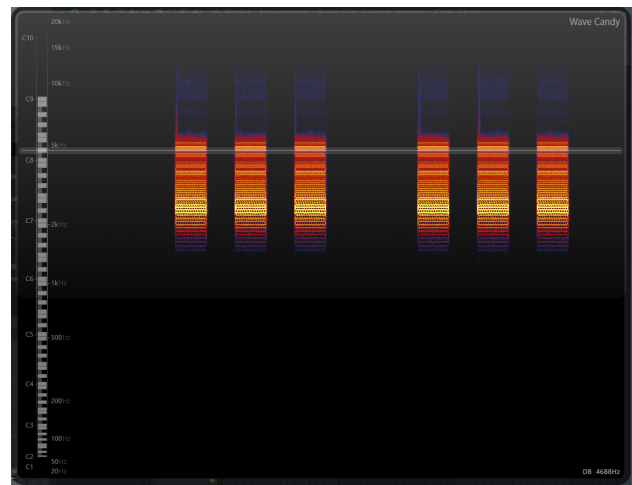


Fig. 2: Gentex-Commander-3-Code-3 alarm frequency spectrum.

III. PROGRAMMING SETUP

Using the IDE that Arduino provides is a bit difficult as I need to look up the usage of every method that could exist with any object that exists in the resource that I want to use. Getting past the language barrier, being the object oriented programmer that I am, was a bit difficult as every code snip that online users post uses hard coded numbers without the plan on expanding the code or using it later or for other applications. I got the lights to react with any little slam of the

microphone down on the table and touching the microphone directly which would cause the lights to react from setting the greyscale value between 0 and 255 based off the amplitude of the sound based off a mapping between the range of 513 and 600 as the microphone module permits.

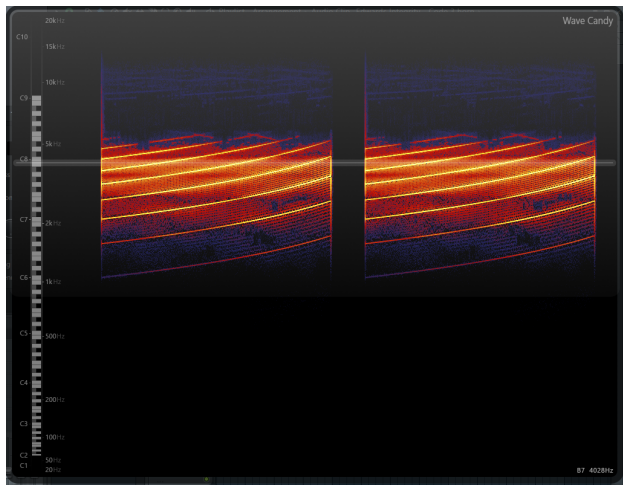


Fig. 3: Wheelock MT-Slow-Whoop-Code-3 alarm frequency spectrum.

IV. JAVA SIMULATION

The idea for a simulation happened when the physical hardware was not adequate for the project within the time constraints needed. The Java simulation uses the Processing© library to do most of the drawing of rectangles to simulate “LEDs” where each rectangle would be an LED with a set color as each LED in a strip would be an assignable color to the rectangle. The library to handle sound and microphone capturing would be the minim library dependent on Processing© as it uses some of the features in the library to make the methods easier to use. Minim has an FFT object which is being used to gather the frequencies across 513 bands as it is half the time segment of 1026.

By using only a percentage of the frequency range to filter out only the frequencies desired from analyzing a few fire alarms and what ranges they fall within will let me determine the activation of the alerting system.

V. ANALYZING FIRE ALARMS

Older models fire alarms fall within the frequency ranges of 2,000 Hz and 3,100 Hz which are the ranges that I’d need to search through. I would also need to consider other sounds being played while those frequencies are being hit meaning that it could possibly be something other than a fire alarm passing through the microphone.

A few of these fire alarms might look like the example showed in Figure 1 through Figure 3 on a frequency spectrum. The bright orange colors in these figures display the louder frequencies, as the cooler colors express the frequency leaking that happens from environment distortion and general distortion of the speaker from the alarm over time.

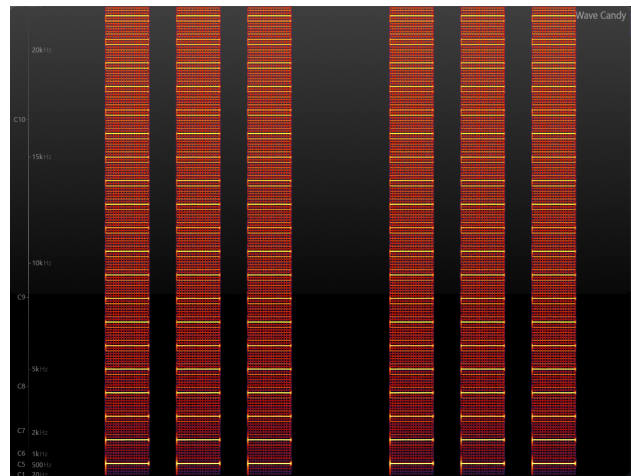


Fig. 4: 520Hz alarm frequency spectrum.

VI. NEWER PROTOCOL

New fire alarm protocol brought about in 2014 has begun to be enforced as studies from the National Fire Protection Association (NFPA) that white flashing strobe lights might not be the best at waking up hearing impaired as the community seeks to develop a new solution to the problem. Also that a different group of people such as those affected by alcohol or general heavier sleepers suggested that only 56% would wake up to a fire alarm tone at 2,000 Hz to 3,100 Hz frequencies, where as a fire alarm tone at 520 Hz woke up 92% of people under these conditions. This alarm frequency spectrum can be seen in Figure 4. This spectrum covers a huge range and most of these models have some type of flashing strobe LED light for a visual alarm for those impaired but again for waking an individual up with hearing impairments will suit to be a difficult task.

VII. CONCLUSION

From what I can tell is that the frequency responsiveness of the project is straight forward, but more research and work went into the usability of this type of work and configuration as the idea for the code was to allow any individual to use the code for any set of 2 dimensional array of LED lights and would be able to interact with them in any way they would like to as if it was an oversized LED screen.

The applicable use of it being an alert system could be paired with a multitude of different types of alarms as hitting extremely specific frequencies but no other frequency is difficult unless intentional given the nature of sound.

REFERENCES

- [1] Towards a Better Smoke Alarm Signal-an Evidence Based Approach, DOROTHY BRUCK and IAN THOMAS, [Online]
- [2] NFPA 72-2010 National Fire Alarm and Signaling Code, Michael B. Baker, http://etnews.org/docs/NFPA_72-2010_Changes.pdf
- [3] Fire Alarm Notification Appliance, Wikipedia, https://en.wikipedia.org/wiki/Fire_alarm_notification_appliance#cite_note-9
- [4] Fire Alarm Sound Files located on YouTube™ and analyzed using the WaveCandy plugin through the program FL Studio®