Michelle Davies
Final Project - Sound Syncing Color LEDs
MAE 3780 (Mechatronics)
December 11. 2020

Introduction - Introducing the Design

For my project, I chose to build **Sound-Syncing Color LEDs** in response to the *Kinetic Art* prompt. I chose this for the Kinetic Art prompt because music and visual arts are two of the most pervasive art forms, and I wanted to apply mechatronics and signals concepts to connect them. Within this prompt, I generated four questions, each one expanding on the previous question, that I wanted to explore: (1) *What connects different art forms?*; (2) *What connects auditory art to visual art?* (3) *How can I create a mathematical model of the relationship between these art forms?* (4) *How can I use such a model to automate or stimulate a visual, artistic representation?* The goal of my project is to provide an immersive entertainment experience and to complement the experience of music as a form of entertainment.

Materials & Budget Information

Material	Total Price
DAOKI High Sensitivity Sound Microphone Sensor (Class Order from Amazon)	$6.99
Arduino RGB LEDs - 2 lights	$0
Amazon Package - 1 package	$0
2 Napkins	$0
Plastic Dome Cover - 1 bowl	$0
Arduino Uno (from kit)	$0
Wires & 220Ohm Resistors - 16	$0
	$6.99

Functionality & Design

Functionality

The Sound-Syncing Color LED lights is a system that takes ambient sounds as an input and outputs a visual, colorful representation of the given sound. A DAOKI Sound Sensor captures these ambient sounds. The Sound-Syncing Color LED lights is a system that takes ambient sounds as an input and outputs a visual, colorful representation of the given sound. These ambient sounds are captured in analog form by a DAOKI Sound Sensor, and then they are interpreted and translated using Fast Fourier Transforms ("FFTs"). The FFTs convert the

Analog input into sound frequencies, which correlate to musical notes. I have preassigned these notes to different colors in the form of tuples of red, green, and blue values ("RGB values"), where each of these values is an unsigned integer analog input between 0 and 255. The system outputs the color on the RGB Lights that corresponds to the given note frequency for the time duration of that sample.

Design Parameters and Rationale

For the design, to see the lights while protecting the circuit, I used a clear plastic bowl with a napkin as a water spill indicator like you see in the new MacBooks. I also used an Amazon package to create a protective layer for the wiring and battery and to make it portable. Additionally, I made a slit under that bottom layer for the plug. For my circuit components, I chose modules that would provide the most range and adaptability with the values it intakes or displays. The main components that I chose were the Daoki Sound Sensor and RGB LEDs.

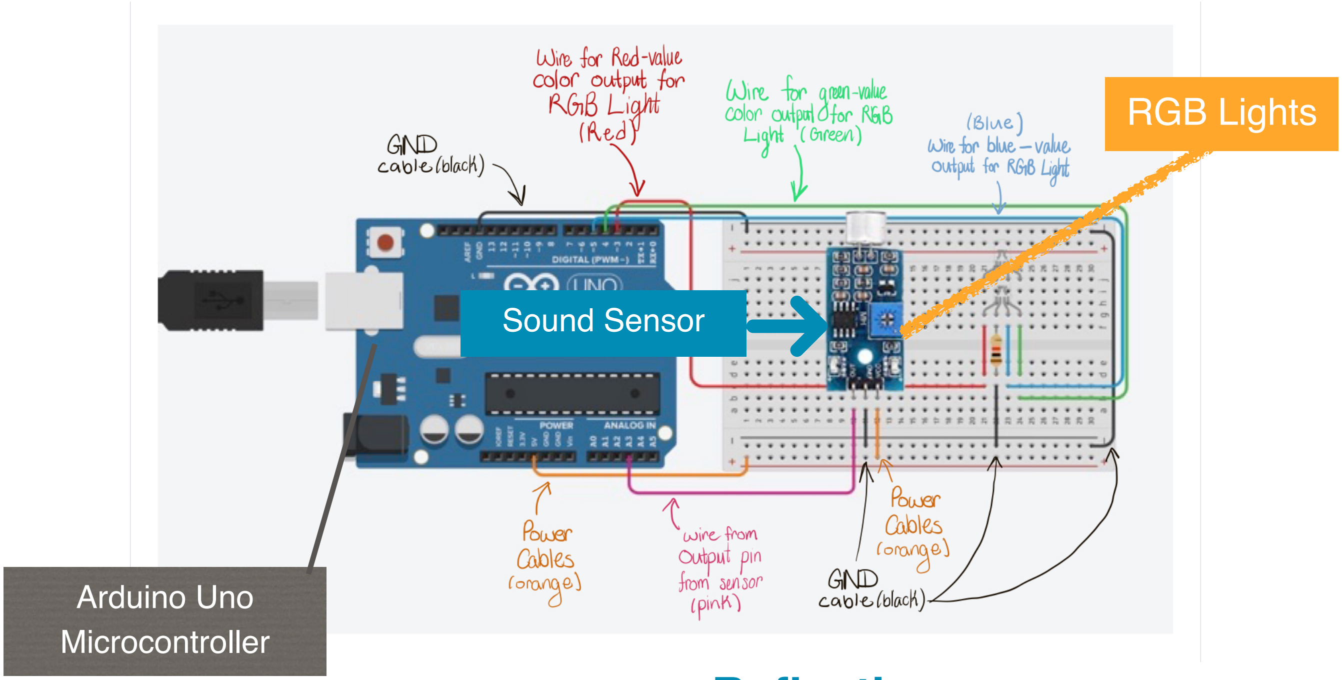
I chose the Daoki Sound Sensor for my project because it allowed me to receive initial data in Analog form, readings ranging between 0 and 1024, which in turn made it a more ideal process to decode the input for each audio sample's most dominant frequency, or "peak frequency." The sensor does not record complete soundbites - it captures instantaneous sound waves in analog form and extrapolates frequencies. When the sensor detects a sound, the frequency is measured, and then the lights dim when "silence," or constant, low-level noise below -40dB is detected. When the sensor detects a valid registered noise, the Arduino board decodes that information. That is a process we will discuss momentarily and outputs signals to pins 3, 4, 5, telling the light to display tuples of RGB values. The RGB Lights output the color that corresponds to the sound frequency or note detected by the sensor. Using the RGB lights was my choice for the actuation because I think the range in colors that you can display on it with Analog functions suited this project, and I can exhibit more precise color transitions between notes.

High-Level Review of Code

In my Arduino program, I have utilized the built-in Analog functions as well as the Arduino FFT library to capture the sound inputs and extrapolate a peak frequency to translate into colors. For the color translation, I have made multiple functions to write the output to the pins for the lights, decide what the outputs are, and make color transitions. In the main logic, I put these functions together in the main logic, which wasn't as difficult. The main microcontroller functionalities that I implemented were polling, for the microcontroller to keep checking the status/reading from the sensor and processing the inputs, and timers to get the duration that the led should display a light based on the length of a note.

The most challenging aspects of programming were perfecting the lights' timing, and defining the analog input to frequency to color. I think that these were the hardest parts because they were both subject to perception and unideal real-world conditions, which meant that I as the programmer had to pick what assumptions I could make to create disproportionate, unpredictable behaviors from the device.

Circuit Schematic



Reflection

Reflecting on my group interactions

Giving updates was helpful because it held me accountable to make significant progress to update my group and get useful feedback. It was also a helpful element of the project because I was forced to explain where I was in my work and what my project does, and where I was going with it. That helped me gauge my own understanding of my project. The most usual feedback I got from my teammates was about the manner in which I assigned colors to frequencies. Ingrid asked me how I was going to go about that assignment, which made me think about the importance of the color assignment, and Anna advised me to be intentional about what colors I included. This made me think about color blindness and how my color choices could affect the product's usability for those who have it, which is a large population. Looking back on this advice, I recognize the ablest assumptions in my programming. They also advised me to think about if I wanted my project to be for music or for general sounds because I thought about potential extensions of this project. The least useful feedback I got was "sounds cool (regarding our progress)," and I was guilty of this too in the first discussion. This wasn't helpful feedback because while it was affirming, this comment doesn't provide any details to give that affirmation any weight. I think that the most useful feedback that I have given was asking questions about how my group members were implementing their project ideas because I think that the act of explaining what you are doing forces you to think about what you are doing and whether it makes sense, and why.

Reflecting on the process of developing my project

Something that worked well when I was working on the project was creating the functions that assigned the notes to different colors and outputted those colors. This mapping and outputting process worked out well because once I broke down the tasks that I needed my program to complete into different functions, mapping the notes to the colors that the RGB lights needed to output was a very doable process.

Something that didn't work as well when I was building my project, which also happens to be one thing I would change if I had more resources, is implementing side RGB LEDs for the background noises that the sensor picks up. This output functionality didn't work out for a couple of reasons. For one thing, I did not have enough RGB lights or other analog lights for background lights to make sense. It would've been difficult to tell which light is representing the colors of the foreground sound, and which is representing the colors of the background noise. Another reason that this didn't work out is because of the limitations of the program in only effectively capturing the major peaks while having trouble with deciphering background noises from valid background frequencies. A valuable lesson that I got from this challenge was that there are always real-world phenomenons and random errors that will deviate from the behavior of an ideal model. I'm glad to have discovered this because it gives me a way to expand upon this project: I want to look more into why this is happening and apply a coding update to my project, as well as add more RGB lights to show foreground and background sounds.

If I had more resources (and if the project allowed for it), I would change the packaging of the material. I am glad that I used the materials I had available because it worked out better than I had expected. Though, I also wish that I was still in Ithaca will access to 3D printing because I have a beautiful design in mind. However, I don't regret not being able to do all of that for two reasons. The first is that it's important to be resourceful because sometimes you won't have everything you want to build what you want. The second lesson is that it's good to build a prototype as a proof of concept before you spend money to make it fancier, and focus on making the machine work well first. Now, I can add on to the project and eventually build a 3D package designed for my project.

The biggest insight that I gained in developing this project is that experimenting with your inputs and outputs early on is not only important for getting accustomed to them, but it can also help with observing expected and unexpected behavior, and them adjusting my plans and models to reflect those observations.

The most surprising part of this project was the connections that I drew between this project and Signals Processing concepts that I learned in ECE 2200. Mainly, I was surprised that I not only used FFTs for the input signal processing, but I also used Nyquist rates and sampling frequencies. This taught me how truly disciplinary different Engineering fields are; I never thought that I would be using signals concepts again, especially in a Robotics context. Now that I know an example of how these two fields can join, I will definitely pursue more opportunities to learn more about Signals.