Music to Visual Projector

Gustavo Monaco, Jaaquan Thorpe, Danielle Garsten, Marcos Berrios

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — Taken from inspiration from Neil Harbisson, the cyborg who can hear colors, our device will allow for individuals to see sound. In the simplest terms, this project's intended purpose is to transform music into light. It receives an audio input chosen by the user and showcases a visual signal that is correlated to and therefore representative of the sound data. It can work with any smartphone via file upload. It converts and displays any musical data into visual display.**

***Index Terms* — EVM, Music Visualization, Beam Expansion,**

I. INTRODUCTION

The AUDIOVISIBLE device has been devised to fulfill a single yet daunting objective: to serve an underprivileged audience by delivering the visual experience of an audio experience. In other words, this device will attempt to make music, visible, for the deaf and hard-of-hearing population. According to United States Census data in 2011, roughly 600,000 Americans (0.22% of the population) were "functionally deaf", while over 35,000,000 Americans (13%) reported some sort of hearing trouble or impairment. Seeing such a huge audience living with a decreased ability to hear, AUDIOVISIBLE was started to make a previously unattainable artform much more accessible.

Not only must the device transform music into light, it must be lightweight, accessible, easy to use and share with multiple people in a room the same way a portable music player would be. The team envisioned a small portable box resembling an old television or radio which could receive data wirelessly and be easily setup and operated. One could simply leave it in a common area in their home and turn it on whenever they had guests or wanted to enhance their music-listening experience.

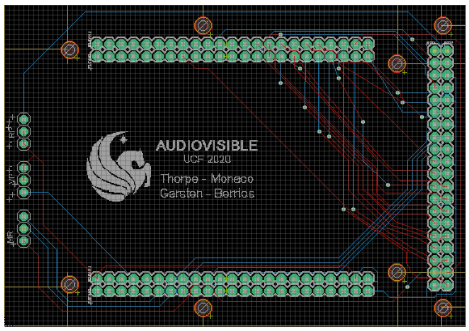
Figure 1 - Early conceptualization of final product. It is sleek, comfortably fits in your home and is aesthetically pleasing

The system was designed with cost in mind, the budget for initial prototyping and building was set to be $500, with the expectation that this could be greatly reduced once a final design had been implicated.

II. SYSTEM DESCRIPTION AND COMPONENTS

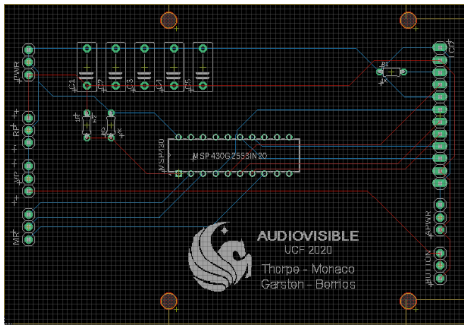
The device is implemented on customized PCBs, with the signature the AUDIOVISIBLE brand name on it, plays the WAV or MP3 file of choice. Using MSP430 Microcontrollers with embedded C code, this allows us to gather information on the optical hardware. We are using a DLPDLCR2000 EVM with customization to the lenses and Texas Instruments Digital Micromirror Device (DMD).

A. PCB DESIGN



GPIO PCB

The PCB above is used to directly connect the Raspberry Pi 4 to the Evaluation Module (EVM). [1]In the design, there are multiple connections being made. Using the header pins, the connection not only powers the EVM with 5V/3A but communicates and sends input signals at approximately 33GHz based on the recommended GPIO pins. The EVM uses I2C to receive signals from an external host processor. With that connection, we were able to run test and program the device based on its limitations. On the other hand, the GPIO PCB have three 3-pin wires connected to the MSP PCB to also send signals to the microcontroller to display messages to a LCD screen.



MSP430G2553 PCB

Our MSP430 PCB, is a simple design that allows us to send simple messages to an LCD screen from the Pi for user notification. For example, the LCD will display a welcome notification to show that the device has been successfully powered on. However, if there is an error during startup the user will be notified, because the EVM will most like not display anything.

B. OPTICAL MODULE

The optical module was designed to produce an image of high color quality and resolution. We chose to use DLP projector technology, with LEDs and a high resolution Digital Micromirror Device would give us exactly that. The DLP2000 Evaluation module is a pocket projector produced by Texas Instruments, illuminated by an RGB diode array with an nHD pico™ DMD with a display resolution of 640 × 360, which is exactly one ninth of a Full HD (1080p) frame and one quarter of a HD (720p) frame.

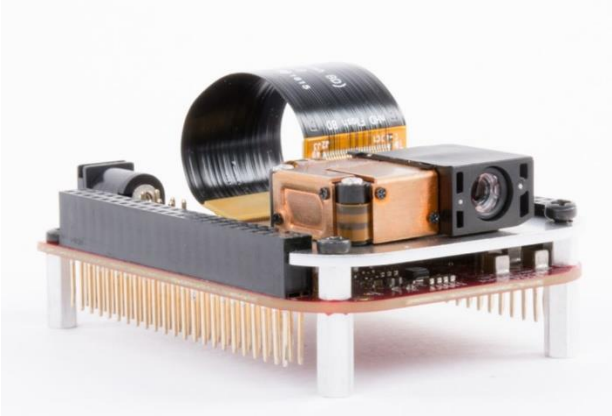


Figure 2 - The Texas Instruments DLP Lightcrafter™ Display 2000 Evaluation Module

The projector we chose has an adjustable focus with two positions, focusing at distances of 9 and 13 inches. The first has vertical and horizontal throw ratios of 3 and 2.25, respectively. The focal plane at 13 inches away has vertical and horizontal throw ratios of 4.85 and 2, respectively. This means that at 13 inches from the projector, it produces an image with dimensions of 5”x6.5” and at 9 inches it has dimensions of 3”x4”. This is not acceptable as our system specifications required an 8”x11’ image. This means that an optical system needed to be designed to re-image the image produced, enlarge it, and bring it nearer to the device so that it fit within an appropriate casing. What was needed was a system resembling a beam expander.

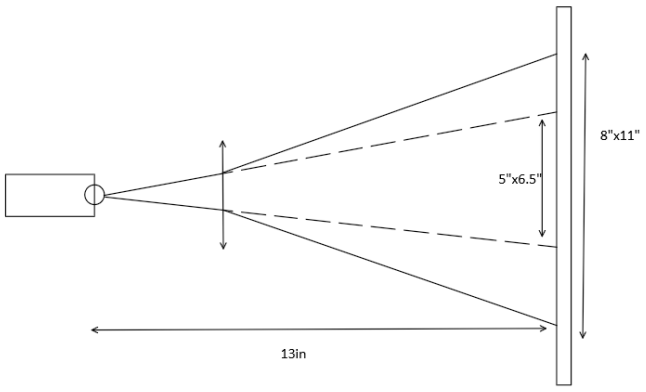


Figure 3 - Rough sketch depicting the projector (left) and the idealized lens system (center) defocusing and creating a larger image at the same distance

The system requires a negative lens to expand the beam and a positive lens to allow the light to focus. In order to magnify the image from 5”x6.5” to 8”x11” at 13”, we’ll need a magnification of about M = 1.6.

Initially, the intention was to create a system with enough magnification to create an 8”x11” image at the shorter focal plane (distance of 9 inches), which would have required a magnification of M = 2.75. Using simple Gaussian imaging equations

$$\frac{1}{z'} = \frac{1}{f} + \frac{1}{z}$$

And

$$M = \frac{h'}{h}$$

Where focal length of the lens required has an inverse-sum relationship to the distances between the image and the object, and the magnification is just the ratio of their heights, it was calculated that a lens of effective focal length EFL = -41.5mm would create the magnification we needed. Due to the availability of parts, a lens of EFL = -50mm was purchased, but upon testing, could not produce a focused image at the desired distances (or any distance for that matter.

The final design for the magnification of the image consisted of a two-lens design, first utilizing a positive lens of focal length EFL = 50mm to create a virtual object closer to the system, and a long focal length negative lens of EFL = 250mm to expand the size of the image at this new focal plane.

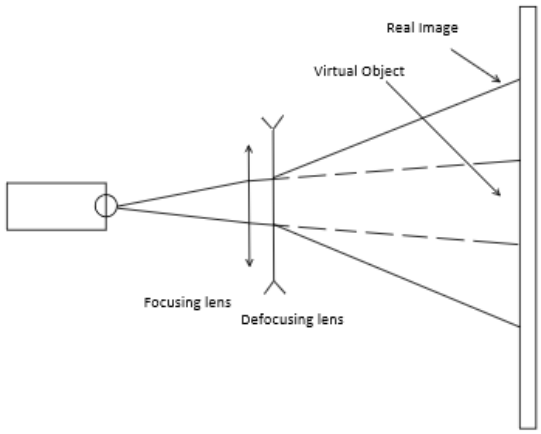


Figure 4 - Simplified rendering of final lens design. From left to right: Texas Instruments pocket projector module, positive lens for creating virtual image, negative lens for expanding actual image, viewing screen

The final result, as designed, became a perfectly focused image of dimensions 8.25”x11” at a distance of 13 inches from the front of the projector.

C. MICROCONTROLLER

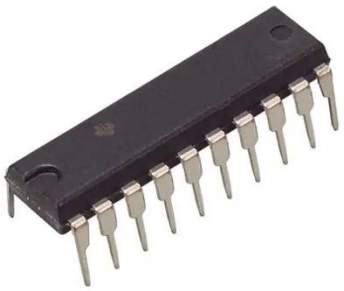


Figure 5 - MSP430 Microcontroller used for control of the power distribution

AUDIOVISIBLE only needed a simple microcontroller that could be used for basic embedded systems. A commonly used Microcontroller, Texas Instrument MSP430G2552IN, was the best option for this project. It was chosen because it is very low cost, a low power consumption, reliable, and c-programming capabilities.

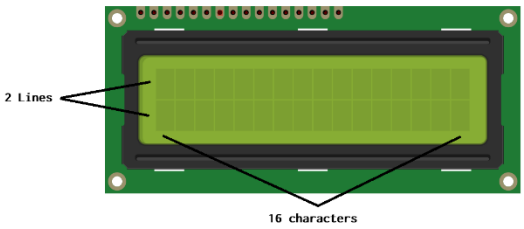


Figure 6 - LCD Screen used to indicate current process status

The Microcontroller creates simple message notifications to a similar screen like the above figure. There the state of the device will be expressed, even when the EVM is in standby mode.

Three messages that displays:

1.“Initializing”

2. “Ready”

3. “Playing”

First, when the device is powered or when the switch is turned on, the display should show the “initializing” signal until the app.py program has started. Once the program has started, the device will then switched to Ready mode until music is played. When the music begins, the device moves into an active state called “Playing”, until the music stops, and no data is being

transferred. Then the system will refer back to “Ready” mode waiting for another song to be played.

D. AUDIO OUTPUT SPEAKER

The audio from our device plays crisp, clear sound, which is correlated to the visuals that are outputted. With an included speaker <insert speaker name> and audio amplification, a louder volume can be outputted across a room. Although this feature is not serving the deaf and hard of hearing audience directly, it is a simple and common feature found in many wireless speakers, necessary to allow this a device to be shared with others in a group, where these others might have good hearing, and therefore expect to hear the music.

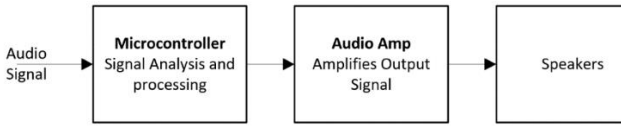


Figure 7 - Audio Output Block Diagram

III. SYSTEM CONCEPT

Conceptually, the brain of the operation is the Raspberry Pi, it is the Pi that sends most of the data to the other components and they just respond to the decisions that are made. Fortunately, the pi is powered independently unlike the EVM that depends on the Raspberry Pi to be give it it’s source of power. The Raspberry Pi sends data to the EVM, MSP430, Speakers, and indirectly to the LCD Display through microcontroller and to the outer screen through the EVM.

IV. HARDWARE DETAIL

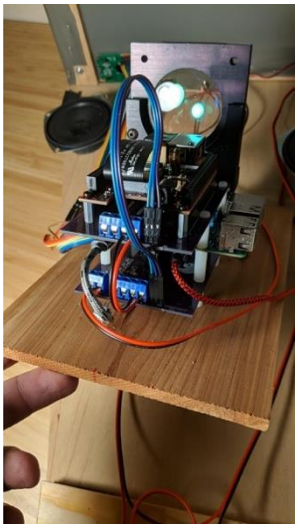


Figure 8 - Final hardware assemby before assembly of housing, including both PCBs, Raspberry Pi, DLP2000 EVM, and expansion optics.

V. SOFTWARE DETAIL

Before there is a deep dive into how the code works, there are significant topics that must be discussed first that inspired the software portion for this device. The expectations in the result should be a visual based off tones and the sound file to play aloud. Tones are correlated to a color which is demonstrated by the Sonochromatic Scale, which will be discussed further later on.

Figure 9 - Simplified Sonochromatic Scale, by Neil Harbisson.

The nuance is with assigning colors to sound frequencies. Because music scales are cyclical (that is, each octave note is double the frequency of the next), it is possible to assign colors directly to notes for color representation. In the device, Neil Harbisson’s Sonochromatic Music Scale (Harbisson, n.d.) is utilized to assign colors, so in this instance Red (~450nm light) would be used to represent middle F (~174.6 Hz), Green would correspond to middle A (220 Hz). The true value

of each displayed color will be calculated and converted using the color system HSV, which directly correlates to RGB. The device is expected to have a listening regime in the full bandwidth of the audible spectrum that is typically used in music (from about 20Hz to about 8kHz). The software needs to be fairly accurate in identifying each frequency within a couple of Hz. We are setting a sensitivity requirement 75%, meaning the device must be able to identify at least 75% of the frequencies it receives at any given time (within a couple of Hz).

B. FOURIER SERIES AND FOURIER TRANSFORMS

The word “harmonic” refers to the analog frequency components of a time-varying signal, be it electrical, mechanical, or optical. Every signal, no matter how complex can be broken down into a superposition of a series of basic “harmonic” frequencies. This series of frequencies is known as the Fourier Series, and it is acquired by performing a Fourier Transform on the time-dependent signal in question. The idea is that if you add together an infinite number of simple sinusoidal waves, alter each one’s amplitude and frequency just a bit, you can create any time-dependent signal (Nave, n.d.).

C. PYTHON CODE

Python was the main language for this project, as it is versatile, simple, and has many resources to work with the audio and visualizing. A user must be connected on the same WI-FI source as the Raspberry Pi that is implemented inside. An application opens on a website where a user uploads a WAV or MP3 file and then it shows a visualization of the tones based off the frequency, wavelength, and RGB values. Each note correlates to a color that share the same wavelengths. For instance, F# shares the same wavelength as a dark red, as dictated by the Sonochromatic Scale.

```
app = Flask(__name__)

@app.route('/')

def index():

return render_template('index.html')


@app.route('/', methods=['POST'])

def upload_file():
```

In the above coding segment, the web application utilized Flask for endpoint connections, GET and POST requests (Grinberg, 2020). The files are routed to the current directory and a basic template is placed. As the user uploaded files, a GET request would be enacted and then once the file plays and the visuals appear, a POST request is completed. In the file upload function, that is where the WAV file is opened, played, and extracted.

```
sound = AudioSegment.from_mp3(src)

sound.export(dst, format="wav")

wf = wave.open(dst, 'rb')

p = pyaudio.PyAudio()

stream =
p.open(format=p.get_format_from_width(wf.getsampwi
dth()),

channels=wf.getnchannels(),

rate=wf.getframerate(),

output=True)

data = wf.readframes(CHUNK)
```

In the above coding portion, the Wav file is being opened and streamed through PyAudio, a Python library (Pham, 2006; Instrument, n.d.). In order to stream, it has to receive the channels and frame rates and then read these frames in chunks. For anything involving Python and audio files, this is the basic coding format implemented. From PyAudio, a file can be played or manipulated. For this device’s purpose, it needs to play the file from this library.

For the visualization implementation, streaming was also infused for the constant flow of images. PyGame was added to have a window pop up and showcase the different colors. Ranges for wavelength dictated the tone to color output as well as RGB (Red, Green, Blue) values. This is where the Sonochromatic Scale is really loaded into the code (Pholey, 2016).

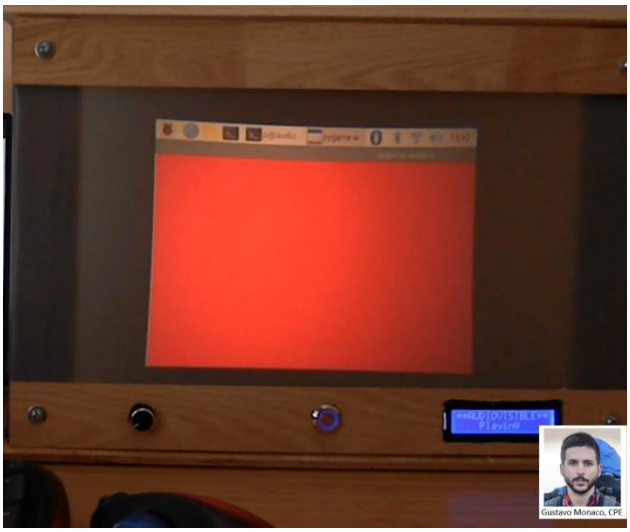


Figure 10 - Display of AUDIOVISIBLE device, visualizing the color red for a pitch with frequency of 375 Hz.

THE ENGINEERS

Gustavo Monaco is graduating in Computer Engineering, who is working with DASI, LLC as a Director of Systems Development

Danielle Garsten is a 21-year-old graduating in Computer Engineering, who is taking a job with J.P. Morgan & Chase in New York City, New York as a Software Engineer.

Jaaquan Thorpe is a 24 year-old Computer Engineering student. He currently works as a supervisor at the Amazon Fulfillment Center, and aspires to be an entrepreneur after a few years gaining industry experience

Marcos Berrios is a 23 year-old Photonic Science and Engineering student. He currently works as an intern at BEAM Co. in Orlando, FL, a company which specializes in diffractive waveplate technology.

ACKNOWLEDGEMENT

The authors would be honored to dedicate their work to those who desire to enjoy music in more ways than sound, in the hopes to make an innovative change for the future as well as Professor Wei and Dr. Hagan; University of Central Florida.

Special thanks to Beam Engineering for Advanced Measurements Co. for their donations to the project as well

REFERENCES

Backyard Brains. (n.d.). *Experiment: Controlling an LCD Screen with Your Muscles*. Retrieved from BackyardBrains.com: https://backyardbrains.com/experiments/MuscleSpikerShield_LCD

BEAM Co. (n.d.). *Diffractive Waveplate Lenses*. Retrieved from https://www.beamco.com/Lens-V2-Diffractive-Waveplate-Lens

Grinberg, M. (2020, July 8). *Handling File Uploads With Flask*. Retrieved from https://blog.miguelgrinberg.com/post/handling-file-uploads-with-flask

Harbisson, N. (n.d.). *sonochromatic-scale-chord-structure*. Retrieved from http://iancdesign.blogspot.com/2013/02/sonochromatic-scale-chord-structure.html

Instrument, T. (n.d.). *TI DLP® LightCrafter™ Display 2000 EVM User's Guide*. Retrieved 12 1, 2020, from

ti.com/lit/ug/dlpu049c/dlpu049c.pdf?ts=1606952427577&ref_url=https%253A%252F%252Fwww.ti.com%252Ftool%252FDLPDLCR2000EVM

Instruments, T. (n.d.). *MSP430G2553 Datasheet*. Retrieved from Texas Instruments Product database: https://www.ti.com/product/MSP430G2553?utm_source=google&utm_medium=cpc&utm_campaign=epd-null-null-GPN_EN-cpc-pf-google-wwe&utm_content=MSP430G2553&ds_k=%7b_dssearchterm%7d&DCM=yes&gclid=Cj0KCQiAk53-BRD0ARIsAJuNhpvaoN18UXHwOxcLbtbaopNrGdQEd68zKr4rbk0ZyhOB

Nave, R. (n.d.). *Fourier Analysis and Synthesis*. (GSU) Retrieved from http://hyperphysics.phy-astr.gsu.edu/hbase/Audio/fourier.html

Pham, H. (2006). *PyAudio*. (MIT) Retrieved from https://people.csail.mit.edu/hubert/pyaudio/

Pholey. (2016, February 16). *Psynesthesia* . (GitHub) Retrieved from https://github.com/Pholey/Psynesthesia

Renshaw, C. K. (n.d.). *Geometrical Optics: Curved Surfaces.* CREOL, The College of Optics and Photonics.