Audio Analyzer and LED Visualizer

A Software and Hardware Design Document

by Dylan Chase-Woods, John Vincelette

Contents

1. Introduction
   1. Purpose
   2. Scope
2. System Overview
   1. Fourier Transformations
   2. Light Emitting Diodes (LEDs)
3. Interface
   1. Software Design
   2. User Experience

***Introduction:***

*Purpose:*

Our project aims to utilize in class materials to capture an audio signal and create a frequency and audio visual representation on a display of LED’s. This document will explain the multiple layers of our software to hardware integration design.

*Scope:*

A LabVIEW program will be used to interface a microphone and the LED array with a computer. The program will take audio data from the microphone input and use the frequencies present to select which output channels to send a signal to. This will allow rows of LEDs to correspond to different frequency ranges. We will also be able to use the amplitude of each frequency range to determine how many lights in each row will receive a signal.

***System Overview*:**

Fourier Transformation**:**

In order to create our system we utilized LabVIEW’s Fourier Transform function after our initial audio capture. The fourier transform function accepts dynamic data from our audio capture and outputs frequency and amplitude. It is a complex valued function with both real and imaginary components, which is used to derive the amount of a specified frequency out of the original function as well as that signals phase offset. The Fourier transform and its inverse can be defined as:

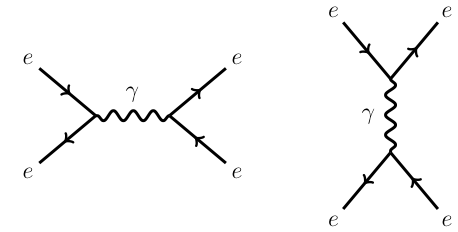
\*Griffiths v.4

Because electromagnetic waves are comprised of two orthogonal waves which can be defined as multiple cosine and sine functions, these integrals allow for the computer to read the wave with respect to both frequency and time. Where in the frequency function of the wave is integrated to define the amplitude, or sum of that signal, at a given time. By breaking down each signal in time into a summation of frequencies, we can pull out specific harmonic frequencies and the intensity of each wave.

Light Emitting Diodes:

Our design also incorporated pairs of LED’s connected in series to receive the true or false +2 volt signal from the DAQ Assist. LED’s or light emitting diodes are similar to diodes, except as the energy of the electrons increases, interactions between the electrons happens creating photons. A diode works by acting as a lattice medium between a positive and negative charge junction, created when one wire of the diode is attached to a higher voltage. By using material that can accept more than one electron in a lattice structure, “holes” can be created as high energy electrons pass through the material, which act as positively charged space.

These holes prevent charge from running the opposite direction as the provided current, and allow areas where the electrons can interact and create light as an energy provided in the form of photons with energy within the visual spectrum.



Arrows pointing in the direction of time in the diagram refer to electrons while anti-electrons arrows are aligned in the opposite direction. As noted from the Feynman diagram above, for electromagnetic interactions between an electron and an anti-electron, a photon can be created as excess energy is released. Any photon does not detemine that it can be visual (i.e. it has an energy that provides a wavelength outside of the visual spectrum), however in the semiconducting material acts as a prism, the energy can be manipulated and slowed to appear in the visual spectrum. In our instance we used blue LED’s using indium gallium nitride as the semiconductor.

***Interface*:**

**Software Design**

*Audio Processing:*

The software was written using LabVIEW 2015. The audio to be visualized is input directly to the computer through a 3.5 mm auxiliary cable, which can be used to input sound from a cell phone, microphone, or any device with an auxiliary output or headphone jack. The audio signal is processed with LabVIEW’s built-in Audio Input function, which converts the electrical signal to useable dynamic data. The data output is in audio amplitude as a function of time, so in order to get it in terms of frequency we must get the fourier transform of the data. To do this we wire the dynamic data to LabVIEW’s Fast Fourier Transform function, which outputs a three-dimensional array of initial frequency, frequency, and amplitude, which is what we need for audio analysis for visualization.

*Interfacing:*

Each column of LEDs on the array are assigned to a frequency range and each row is assigned to an amplitude range. A pair of LED’s in each row were wired in series to represent a single frequency/amplitude range. To accomplish this we need to split the amplitude/frequency data into subarrays. The initial frequency of the fourier transformed data was 0 Hz and data was recorded at every frequency, so we could simply use the one-dimensional array of amplitudes with the index as the frequency. Our LED array had six columns, so we assigned each column to 160 Hz. This allowed our visualizer to cover the full useful frequency range for most music. We found that amplitudes varied dramatically between different songs, so in an effort to make the program more universally useful we assigned a volume control to the front panel of the vi. This essentially adjusts the sensitivity, changing a factor by which the 1-D amplitude array is multiplied before being split. The amplitude for each frequency range was determined by summing the 150 elements of each subarray, giving one amplitude number for each frequency range. Each frequency range had four LEDs to represent four different amplitude levels, each controlled by a single digital output DAQ Assist. To make each DAQ Assist turn on only when a certain amplitude is present, we wired the amplitude number for each frequency to a Greater Than function, which returns ‘True’ when the amplitude is larger than a set value. The boolean result was then wired to the DAQ Assist corresponding to that amplitude. Four Greater Than functions, each with progressively larger amplitude thresholds, are wired to their own DAQ Assist in the same way. The result is that, for a single frequency range, if the amplitude is higher than all of the thresholds all the LEDs will be on, otherwise every LED corresponding to a lower amplitude threshold will be on.

The above analysis determines which LEDs will be on at a single point in time when the input waveform is analyzed, so the entire code is wrapped in a while loop. The waveform analyzed in each iteration comes from a 0.2 second input acquisition, and the entire loop takes approximately 0.25 seconds to run. This means that the LED array changes four times per second to reflect the current input waveform at any given time.

**User Experience**

*Audio Connection*:

User can connect their desired audio content to the computer using a convenient auxiliary cable registered by the computer and necessary LabVIEW vi.

*LED Array:*

12 pairs of LED’s connected in series are controlled by the specified port control on the DAQ Assist. When the program deciphers a ‘true’ statment for a given frequency and amplitude, a +2 volt difference is created, convenient for the minimal +0.6 V required to pass information through a diode. While the user listens to the audio file, the LED’s will light up and act as a visual representation of the amplitude due to a sum of frequencies around the range of notable harmonic frequencies between 0 and 1000 Hz.