Software Overview

Year: 2023 Semester: Spring Team: 18 Project: RDNT

Creation Date: 23 January 2023 Last Modified: 28 January 2023

Author: Anurag Numboori Email: anumboor@purdue.edu

Assignment Evaluation:

| **Item** | **Score (0-5)** | **Weight** | **Points** | **Notes** |
| --- | --- | --- | --- | --- |
| **Assignment-Specific Items** | | | | |
| **Software Overview** |  | x2 |  |  |
| **Description of Algorithms** |  | x2 |  |  |
| **Description of Data Structures** |  | x2 |  |  |
| **Program Flowcharts** |  | x3 |  |  |
| **State Machine Diagrams** |  | x3 |  |  |
| **Writing-Specific Items** | | | | |
| **Spelling and Grammar** |  | x2 |  |  |
| **Formatting and Citations** |  | x1 |  |  |
| **Figures and Graphs** |  | x2 |  |  |
| **Technical Writing Style** |  | x3 |  |  |
| **Total Score** |  | | |  |

5: Excellent 4: Good 3: Acceptable 2: Poor 1: Very Poor 0: Not attempted

General Comments:

*Relevant overall comments about the paper will be included here*

1.0 Software Overview

Our device will allow the user to switch the device’s power ON and OFF. In external buttons/switches, the user will be able to change the mode of operation. There will be 3 modes: Microphone (default), aux-in, and bluetooth. In the first mode, an onboard microphone will capture analog input. In the second mode, analog input will be accepted through an aux socket (TRRS jack). In the third mode, digital input will be accepted through the ESP32's bluetooth module or an external Bluetooth input device. We are unsure currently if the ESP32 is fit to handle high quality digital inputs. Analog input to the device will be read by the ADC and sent to the FFT operations that we will implement. These FFT operations will isolate the driving percussion frequencies. Now that the frequency is isolated, we can send the frequency to our message sender. The message sender will use the default setup to send a color and pin number message to our LED strips.

We will also build a mobile application that the user will use to set the number of total LEDs. Our message sender will pick a color that mostly closely matches a frequency mapping for a given amount of LED output pins. This frequency to color mapping is dependent on the total amount of LEDs. Therefore, our message sender will be able to pick more appropriate colors/pin numbers for customization if more LEDs are connected.

2.0 Description of Algorithms

Fast Fourier Transform [1] is an algorithm that can be used to isolate frequencies. The algorithm uses Discrete Fourier Transform recursively to find the frequencies that make up a given audio input wave. The frequencies that we want to isolate will correspond to common percussion instruments. This algorithm is O(Nlogn) instead of O(N^2) which will make it faster to compute on our microcontroller by reducing computations for known values.

3.0 Description of Data Structures

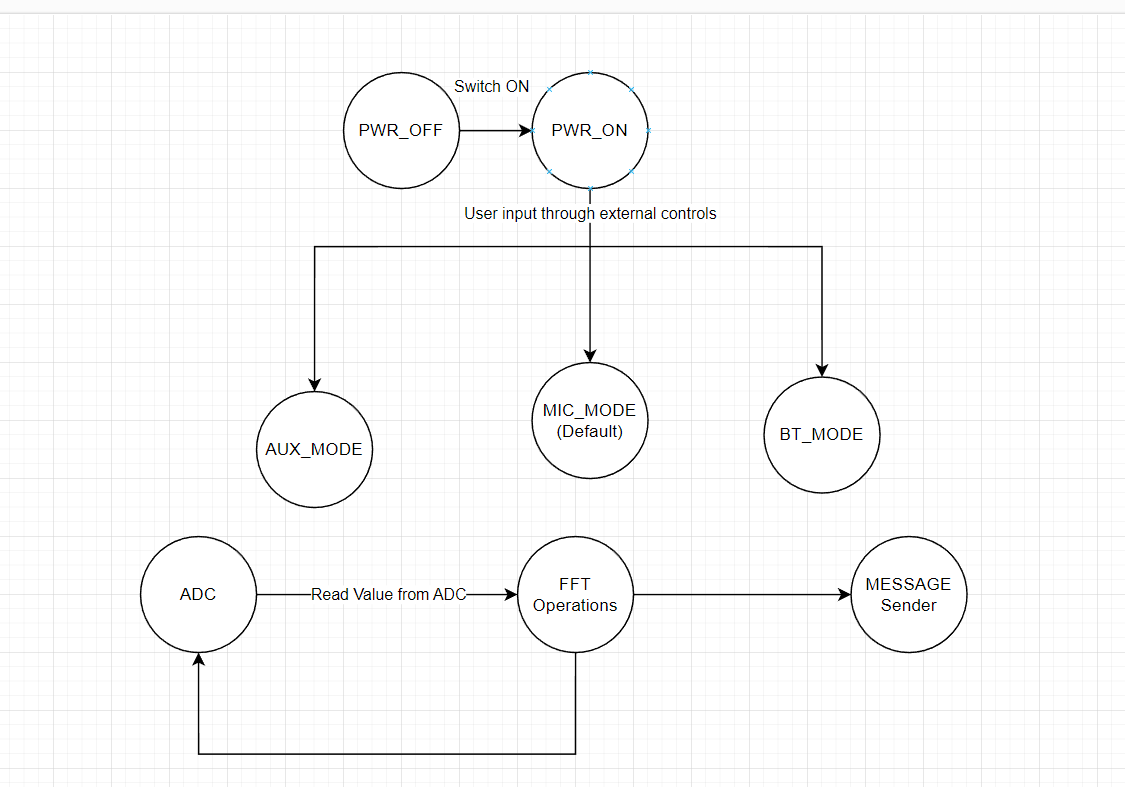
The most important data structures are arrays used to store a single waveform and the output of our FFT. Our single waveform array is used as an input to FFT. We expect to have this array overwritten periodically so that the LEDs are always synced to a song as it changes. Our output array from FFT will clarify where the peaks in frequency are located. Each index in this array represents a “bin” of frequencies. The size of this array will decide the accuracy of our output colors. This array’s most useful index will be sent to the message sender.

Libraries control the driving of LEDs from our message sender. The libraries send variable period duty cycles to change the colors for a given LED on the strip. To change a given LED on the strip, it requires a 24-bit signal which is made of 0s and 1s, which are represented by a single period of the duty cycle.

4.0 Sources Cited:

1. https://jakevdp.github.io/blog/2013/08/28/understanding-the-fft/

Appendix 1: Program Flowcharts

**

Appendix 2: State Machine Diagrams

