Caroline Le
SID: 861220191
EE 120B
Final Project - Jukebox

High Level Description:

- **LCD Screen in start mode**: Welcome note is displayed. Press any button to exit to menu. In menu, user is able to browse through a selection of songs.
- **User presses buttons to select Song**: User is able to select songs, pause, play, and stop songs with button presses.
- **LCD Screen must display song being played**: Always displays which song that is playing.
- **User may press stop button to stop song**

Bonus: Visual Feature that reacts to song being played - LED display that light up corresponding to musical notes.

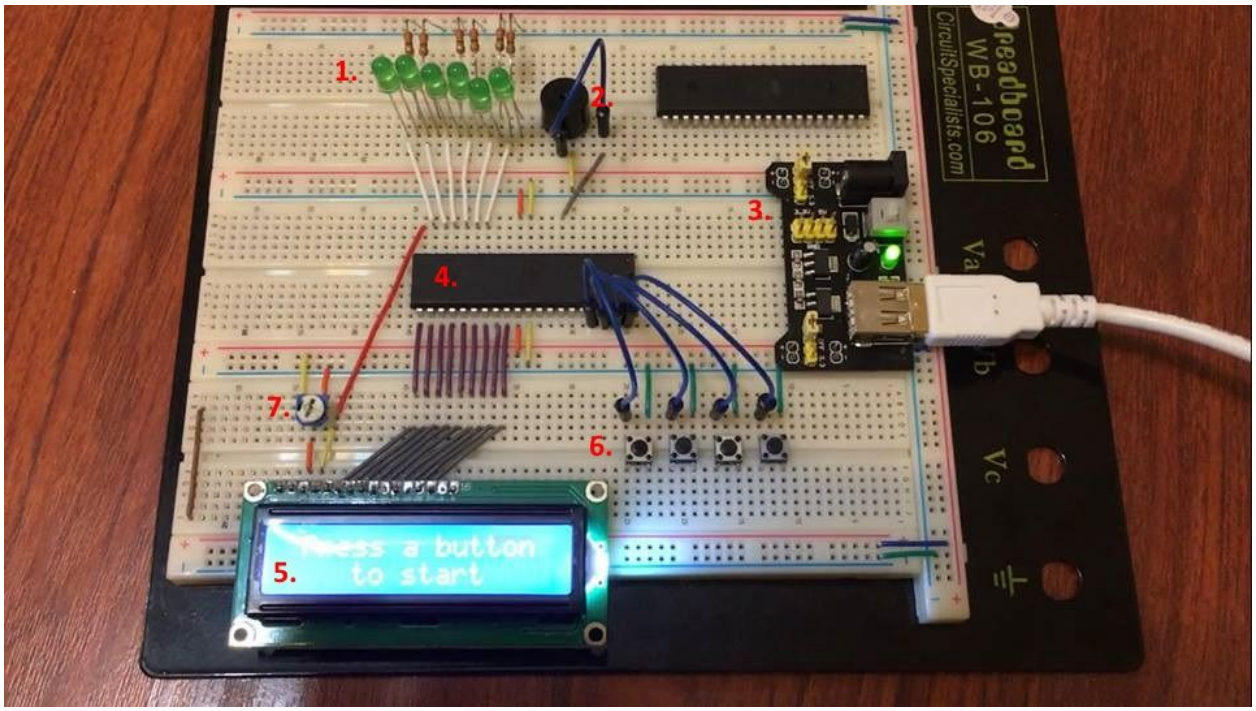
User Guide:

The jukebox brings the user to a welcome message when it is first powered it on. Once a button is pressed, the user is brought to a menu where they may scroll using the LEFT button or RIGHT button and choose a song with the SELECT BUTTON. A short press will scroll to the immediate left or right on the song selection menu. A long press will scroll through multiple songs until the button is released. If the user is viewing the on last song in the song selection menu and presses RIGHT, the menu will cycle to the beginning. Conversely, if the user is on the first song on the song selection menu and presses LEFT, the menu will cycle to the beginning. While the song is playing, LEDs will light up corresponding to the note that is being played. While the song is playing, the user is able to press the STOP or PAUSE button. Pressing the stop button will bring the user back to the song selection menu. Pressing the PAUSE button will pause the song and pressing the same button again will continue the song where it was left off. Additionally, when the song is paused, the user is able to exit from the song by pressing the STOP button.

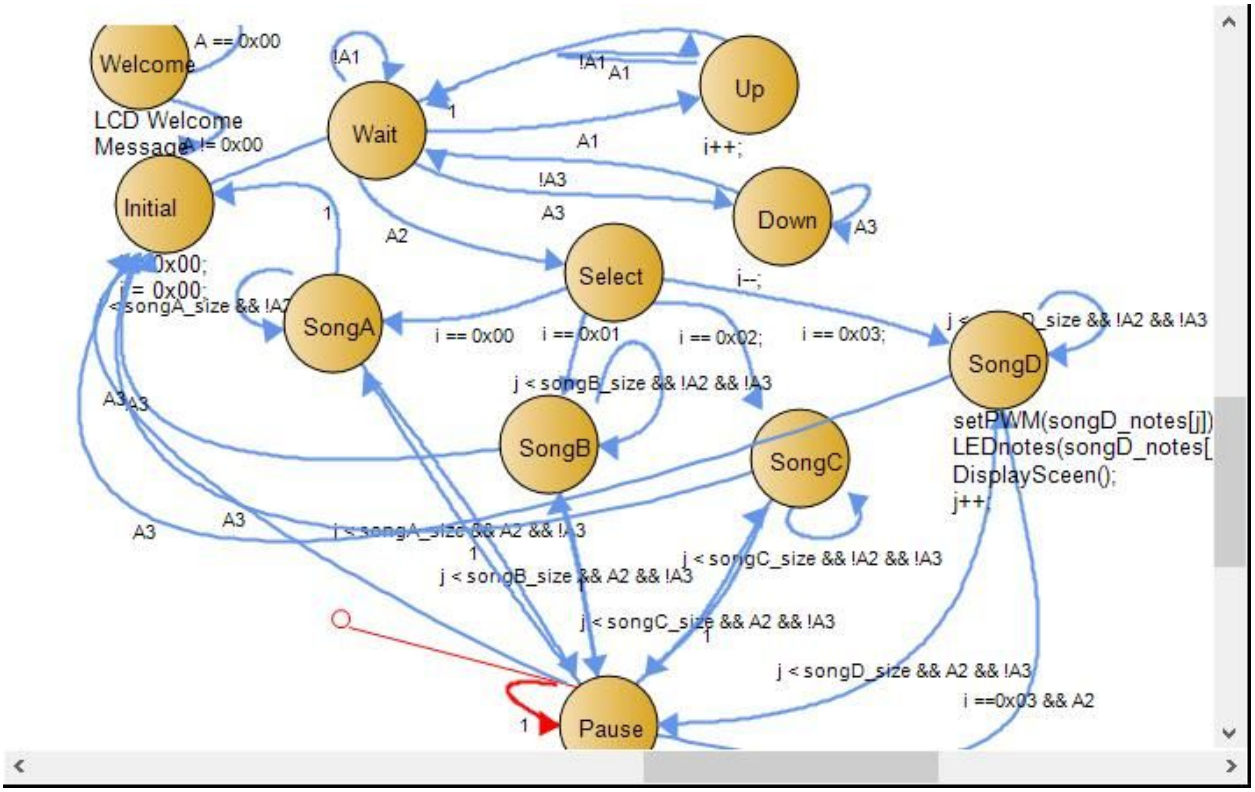
Technologies and Components:

1. LEDs: lights up corresponding to the song's musical notes.
2. Speaker: vibrates at certain frequencies to produce certain notes
3. Powersource
4. ATmega1284: chip in which we program the board

5. LCD Screen: displays messages to the user (ie. welcome message, song name, etc.)
6. Buttons: user presses buttons to communicate to the system
7. Potentiometer: controls brightness of LCD screen
8. Atmel Studio: where we write our code, debug, etc. (not pictured)



The image below is an *extremely* rough SM of the jukebox. Some of the states are missing actions or transition conditions because they are redundant and would make the SM look even more cluttered. This SM was only meant to give an approximate visualization of how each state acts or transitions. States such as SongA, SongB, SongC, and SongD behave the same way, but again, including the actions and transitions for all of these states would be to difficult to follow. For the Song states, refer to SongD to see a pseudocode of the actions and refer to SongB for the transitions.



Obstacles and Learning:

I had not used C before and the last time I coded in C++ was during my first year. This project especially pushed me to become more comfortable with coding in C. It brought together all the components of the previous labs and posed a challenging, yet fun project to work with. When writing the code for this project, I found many problems in my code that I had to work through. I found new debugging techniques and learned to develop my code incrementally. Once I took things step by step, I found that I ran into less obstacles and was able to produce a more efficient and cleaner code when I took things step by step. Additionally, I learned a bit more about uploading to Github. For some reason, when I tried to upload to Github, I ran into many errors and the solutions I found from classmates and Google search results did not solve my problem. I found through experimentally trying to solve my problem with Github, I was able to find an alternative solution to my uploading problems.

Links:

Video demonstration: https://www.youtube.com/watch?v=Y9Nn2RLDis4
Github Link: https://github.com/carolily/jukebox