

Jukebox

Jukebox is a partially automated music-playing device, usually a coin-operated machine, that will play a user selected song from the self-contained media. The classic juke box has buttons with letters and numbers on them, when entered in combination, are used to play a specific selection. My goal is to create a simplified version of this embedded system. I used keypad, buttons, LED, LCD and piezo buzzer to help me to reach the goal.

I recreated the juke box with a Atmega 1284 microcontroller on a breadboard. There is LCD Screen in start mode to display welcome note and instruction. It also displayed list of songs that contained in the jukebox. There are buttons to press left or right to know what songs are available in the jukebox. User can press the number corresponding to the song to select and play the song. The piezo buzzer will start playing the song. User can also press button to stop song or pause song. The LCD Screen will always display the song title of the song that is playing. When the song is playing, there are also LED to react to each beat of the song that is playing. LED's visual feature is a bonus, which give user a better visual experience.

I included multiple libraries to make the program be more organized. I included `io.c` and `io.h` files for the LCD display function. I also included `timer.h`, `keypad.h` and `pwm.h` so there will less code the `main.c`. I defined all the frequency that I will be using so it will be more organized in the code. For each song, I used three arrays to store the frequency, duration and rest time between each note. I used `songplayflag` to determine which song is requested. I also used `stopflag` and `pauseflag` to notice other state machines to stop or pause the songs when the flag raises.

I used AVR studio 6 and an ATmega1284 microcontroller to implement the program. I set the LCD on PORTC and PORTD6...7 for data control lines. I used the rest the PORTD to display a light show on LED. The keypad is set on PORTA, PA7...4 output init 0s and PA3...0 inputs init 1s. It is important to set A0...A3 as output and A7...A4 as inputs for the keypad to work. I also set PORTB3...0 as input for the buttons and PORTB6 as output for the piezo buzzer.

There are some challenges during the project. I had a hard time to find the songs that I want because some websites do not provide the correct frequency or rest time. It also took me awhile to understand how the keypad works because lab 10 is not required. I have a hard time to understand that how the keypad is set up. There are some mistakes when I was making the stop and pause button. I ended up using a flag to notify other state machines that stop button or pause button is pressed. There are some incomplete ideas due to the lack of knowledge and lack of time. I was planning to show "Now Playing:" on the first row of LCD and song title on the second row of LCD. However, the first song title "Twinkle Twinkle Little Star" is too long to fit in the second row. Therefore, I planned to have scrolling text in the second row for the first song title. However, I am not able to complete that idea and I hope I can learn how to do that in the future. If I have more time, I will include the skip button too. From this project, I have a better understanding of the importance of flag and the importance of concurrent state machines. Flag helps different state machine communicate in a better way.

