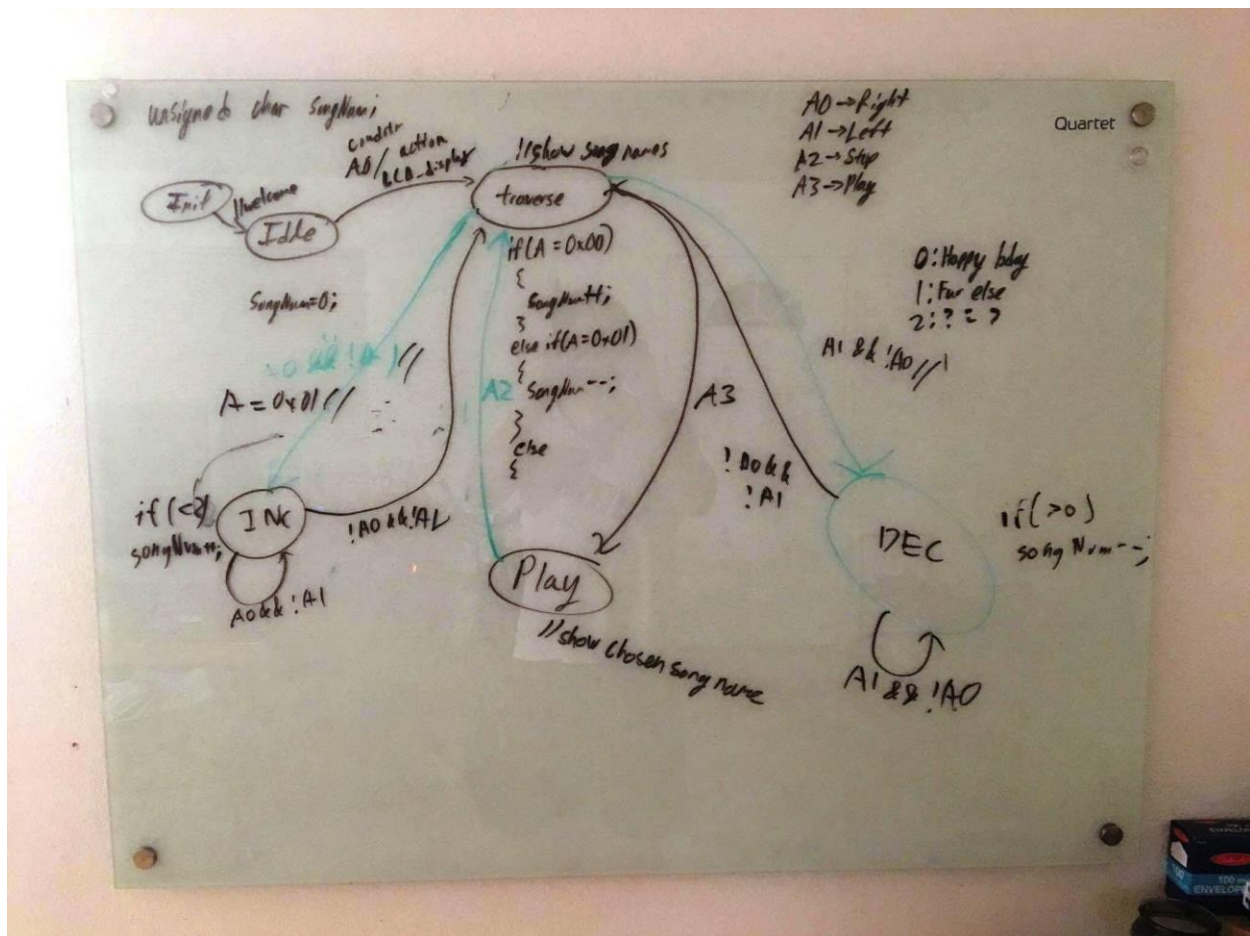


FSM



High level description of custom lab and user guide

For my final project, I chose to make a jukebox. The materials that I am using for this project is an ATmega 1284, four buttons, and a speaker. The jukebox works by initially (idle state) by showing a welcome screen that directs users on what to press next. After the user breaks out of the idle state the user can then toggle with A0 and A1 to choose the song of their choosing. When the user chooses their song of choosing the user presses the A3 button to play the song. The A2 button serves as a menu button. The menu takes the user back to the idle state where it shows the welcome screen.

Learning moments

Initially I tried to interface an SD card onto the ATmega 1284 but unfortunately time was not on my side. I learned more in depth about the LCD and its connections. I learned what MOSI (master out slave in) and MISO (master in slave out) stand for and how they work. An SD is read in by SPI (serial peripheral interface) which uses the chip select, MISO, MOSI, and taking in more than 3.3V will cause damage to the SD card. Upon learned all of the information my problem with interfacing it with the ATmega 1284 was due to old/obsolete methods to code the SDHC card. The problem is that the code was using methods of connecting SD cards that had memory 3GB or below whereas mine was 16GB. I hope to find a way to interface it in the future.

Going back to PWM lab I used frequencies to make simple 8-bit melodies. The problem I came across with the project was finding a way to put a duration between the notes. After getting it to work I had a problem with the menu button (A2). The melody function is stuck in a while loop. When pushing A2 the user must keep pressing or holding the button to break out of the loop. Fortunately, it will break out of the loop. Another thing about using the while loop for the melody function is that a timer messes up how the song plays.