Zelda Two Step (DDR)

Custom Project Final Report

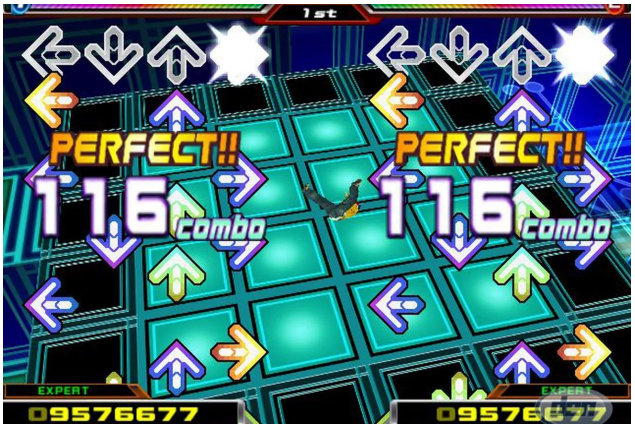
Spring 2019

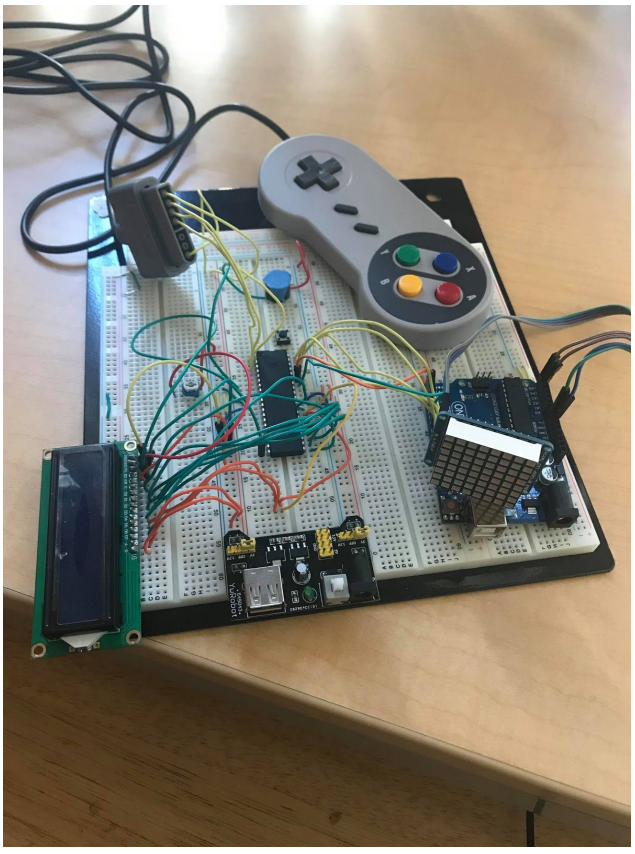
Luis Sanchez

Table of Contents

Introduction

Dance Dance Revolution is a two-dimensional dance game in which the user matches the icons on screens in time with inputs. The player will use a SNES controller to match the icons as they go up the LED matrix. The game will continue until the song is completed. There will be 3 different speed difficulties. The aim is to match as many icons as possible in order to gain a higher score.





User Guide

Rules:

1. Press the corresponding direction pad arrow when the block reaches the middle of the two red lines.

Controls:

1. Directional pad arrows
2. Select = first song
3. Start = second song
4. L & R = Reset high score

Technologies

In this project, I learned how serial data encoding works. I learned how a clock and latch works to control the data being sent from one component to the other. I also learned how EEPROM memory works and how to write and read from it. I created custom characters by setting bits high, drawing, on the LCD display and how to write to the LCD memory. I learned how Arduino is used and how LED matrices are made and used. Lastly, I learned how concurrent state machine task management is programmed.

Hardware

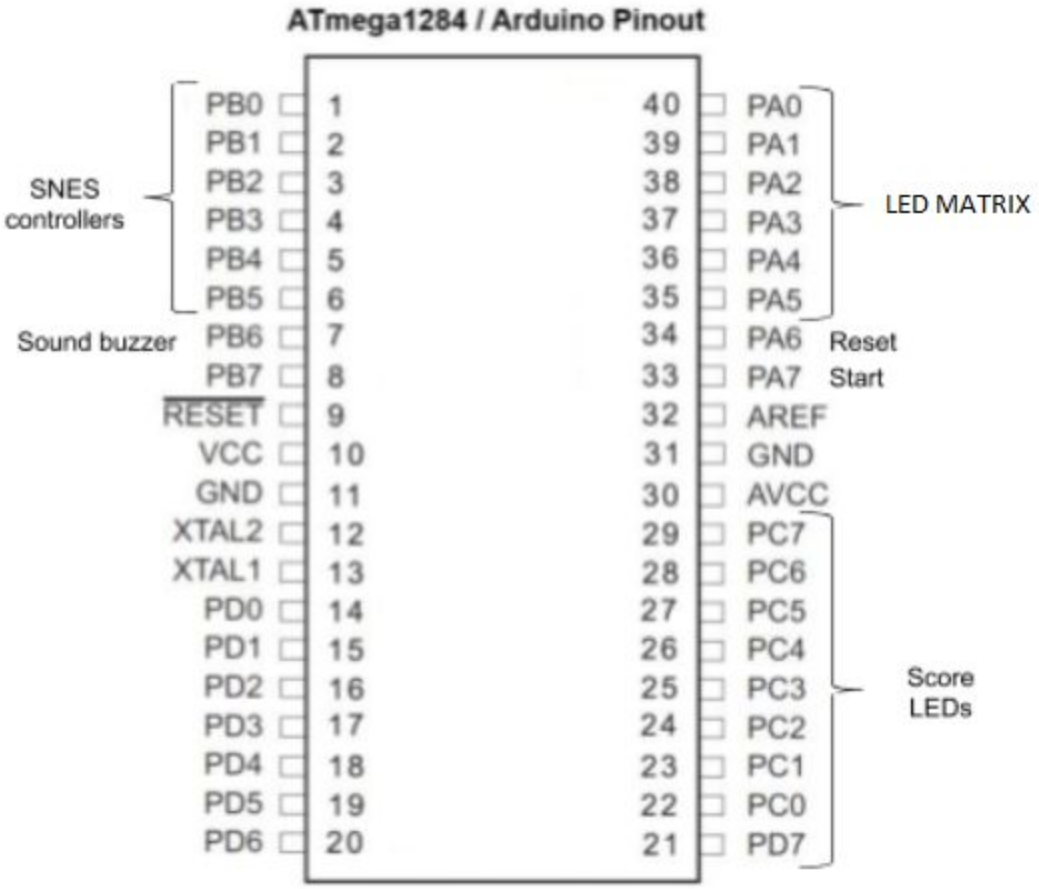
Parts List

The hardware that was used in this design is listed below. The equipment that was not taught in this course has been bolded.

- ATMega1284 microcontroller
- **Arduino UNO**
- **Adafruit led 8x8 matrix**
- Buzzer
- 16x2 LCD display
- **Custom Characters**
- **SNES Controller**
- **EEPROM**

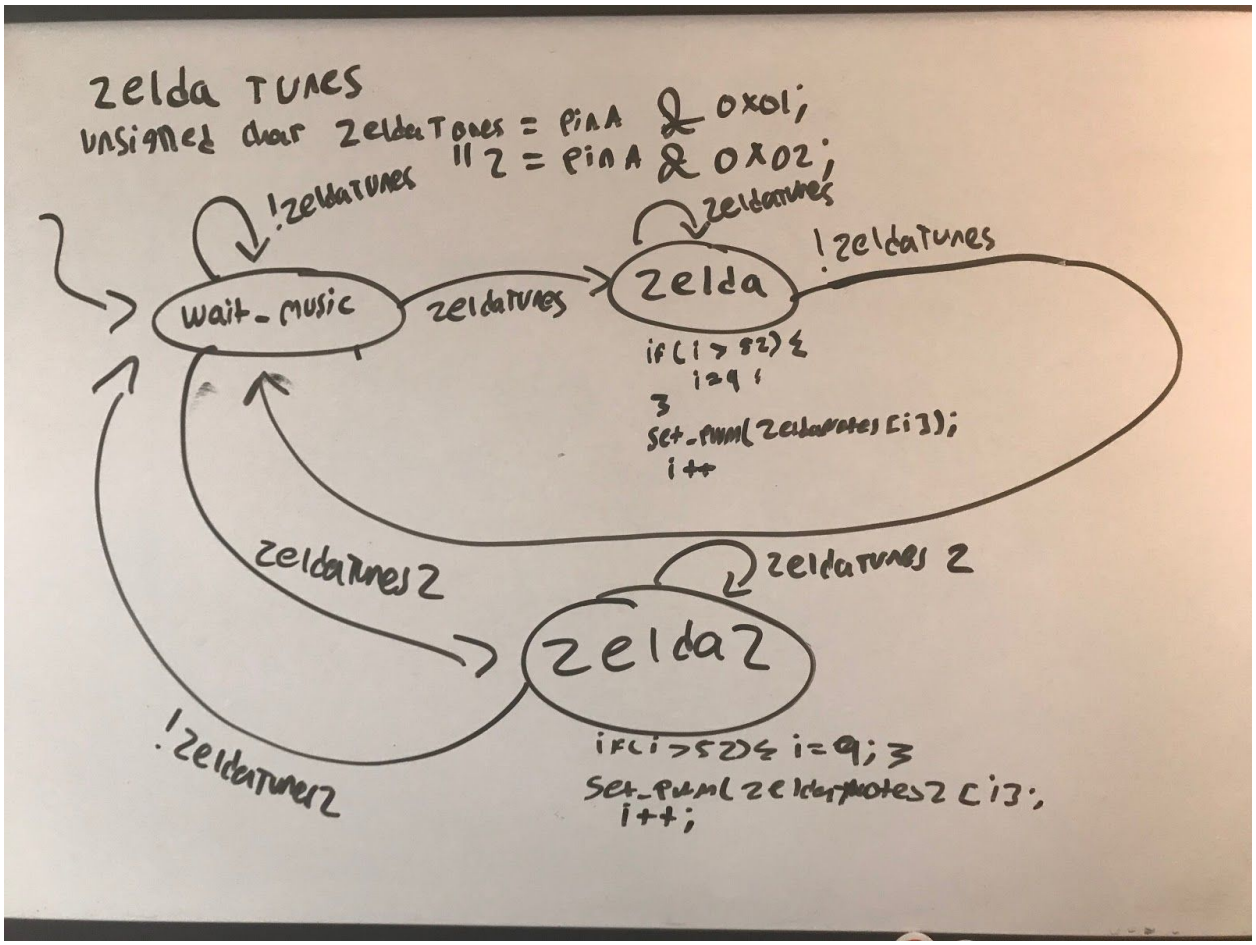
Components (Pin-out)

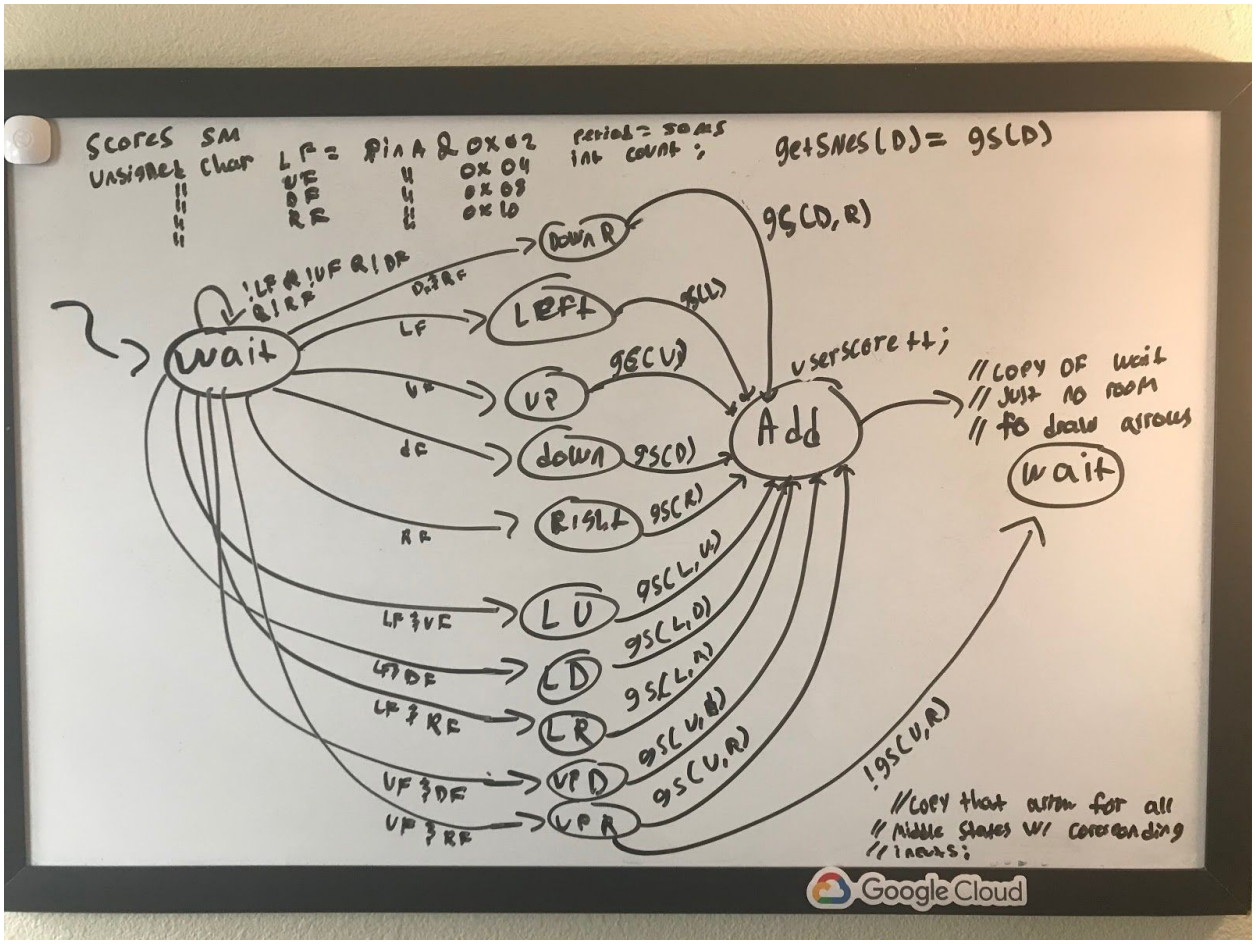
- **Inputs**
  - SNES Controller
  - Reset and start buttons
- **Outputs**
  - 8x8 LED matrix. This will be used to display the game
  - LCD screen to display the score and start menu
  - Sound buzzer for music

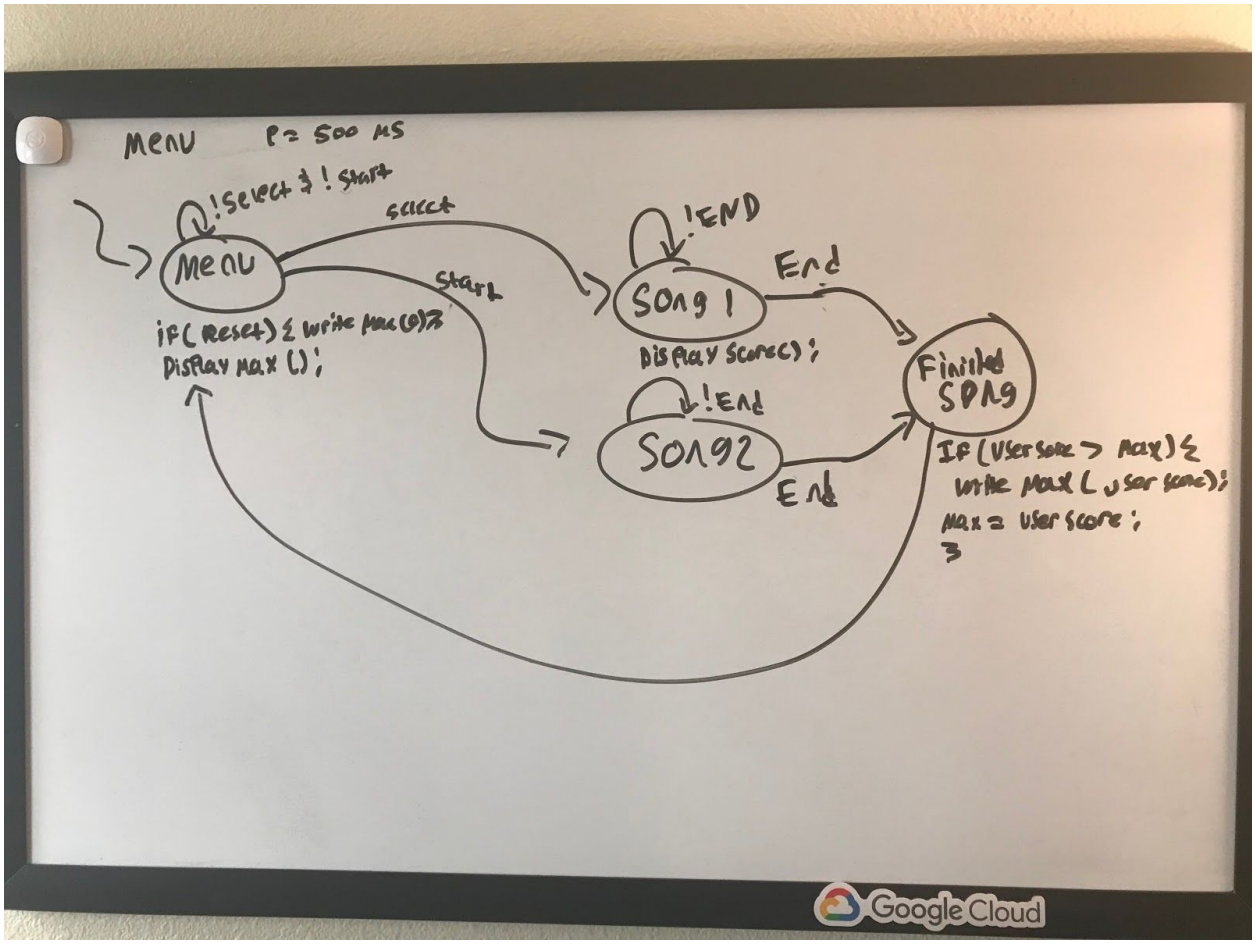


Software

I used three concurrent state machines: ZeldaTunes(500ms) Score(50ms) Menu(500ms)




Scores SM
unsigned char
LF = PinA & 0x02
UF 0x04
DF 0x08
RF 0x10
period = 50ms
int count;
getSNES(D) = gs(D)
gs(D,R)
wait
!LF&!UF&!DF&!RF
DF&RF
LF
UF
DF
RF
LF&UF
LF&DF
LF&RF
UF&DF
UF&RF
DOWN R
LEFT
UP
down
RIGHT
LU
LD
LR
UPD
UPR
gs(L)
gs(U)
gs(D)
gs(R)
gs(L,U)
gs(L,D)
gs(L,R)
gs(U,D)
gs(U,R)
Add
userscore++;
// copy of wait
// just no room
// to draw arrows
wait
!gs(U,R)
// copy that arm for all
// middle states w/ corresponding
// inputs;
Google Cloud



Complexities

1. Using 8x8 LED matrix screen to display the DDR screen
2. Creating custom LCD icons (½)
3. Changing the buttons to a SNES controller
4. Using the EEPROM to save the high score of the players (½)

Youtube Link

https://youtu.be/WVLpBQQFC5A

Known Bugs and Shortcomings

- If the player double presses one of the inputs when the corresponding input is in the red barrier two points will be added instead of one. A fix to this would be to add to not allow any input checking after one input until the pin is set to low.
- The wires into the control sometimes become loose and require them to be pushed back in a little. A fix to this would be to cut the wire on the controller and solder the input wires to the wires on the controller directly.

Future work

I would like to add two player functionality for people to compete head to head on the same song at the same time. I would also want to add more songs and a harder difficulty that requires the use of more button inputs.