

CS/EE 120B Custom Laboratory

Project Report

Hangman

Karam Shanti

12/07/2022

Introduction

The new game I have chosen to make is hangman. It will be played by one player and they will go through a series of levels to unravel many words. Get a guess right, get a bonus lifepoint, get a guess wrong and you will lose a life point. The game will keep going on until your life points go to zero.

Complexities

Linear Regression: The linear regression aspect of the game worked properly when you beat a level, you would move onto the next level while maintaining your current lives and get either a 3 letter, 4 letter, or 5 letter word depending on the level you were in.

RNG: The RNG did randomize the words perfectly almost every time the game was refreshed in every level of the game until the player lost or won.

Servo Motor: The servo operated according to the pulses I gave it and only went off when it was going into a new level and only rotated once.

User Guide

So for my hagman game I designed a letter scroller and enabled a joystick that a user could use to scroll through an array of letters and select the letters they think would be correct. If they guessed the right letter then their letter would be output on the displayed dashes and gain a life (max of 6) or if not they would lose a life. They will keep doing the same thing for levels two and three. Activating the servo once they enter a new level. For level two you'll get a 4 letter word and for level 4 you'll get a 5 letter word both completely randomly generated.

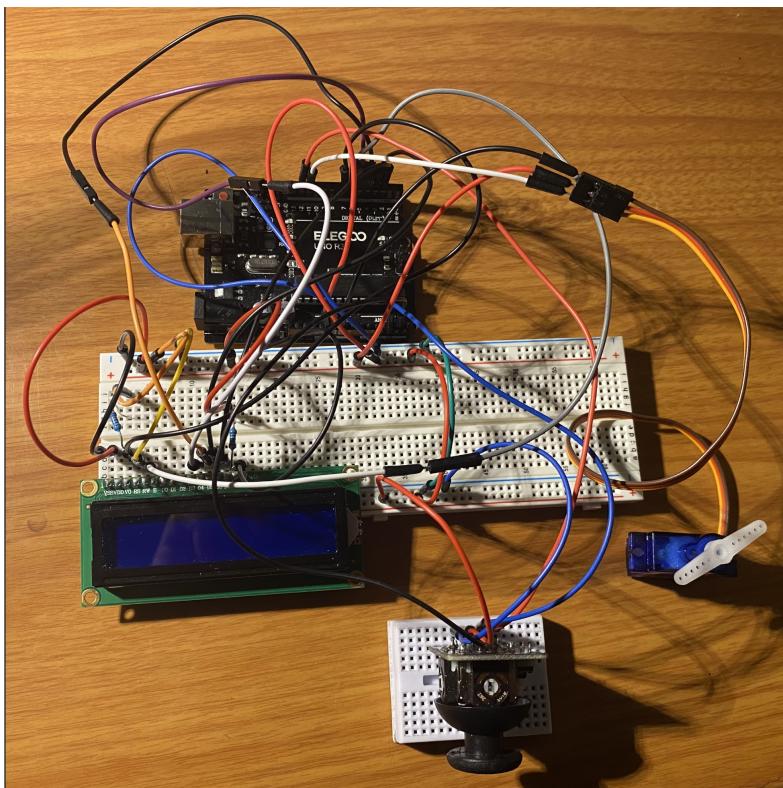
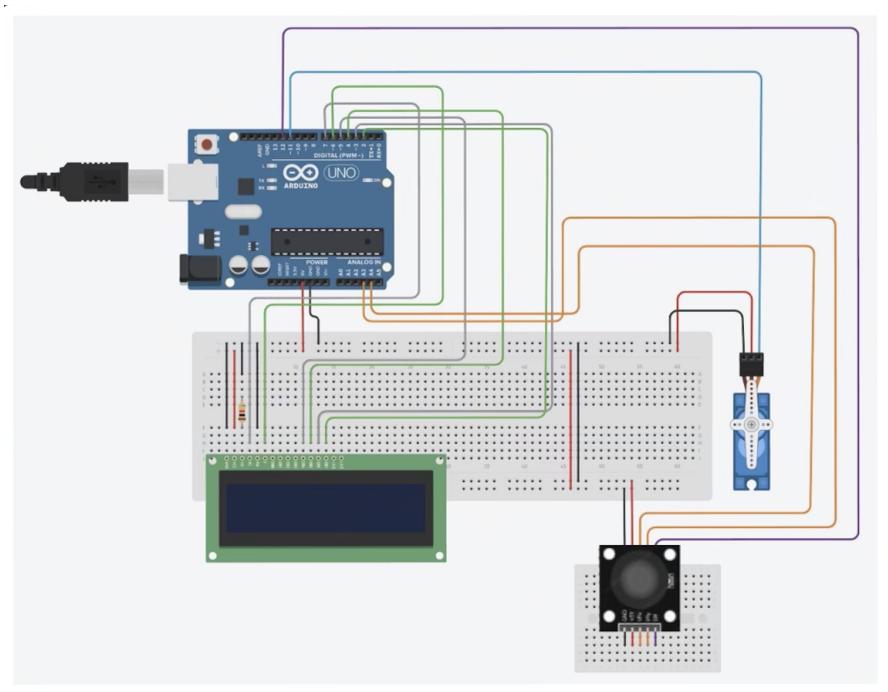
Hardware Components Used

~ Servo Motor

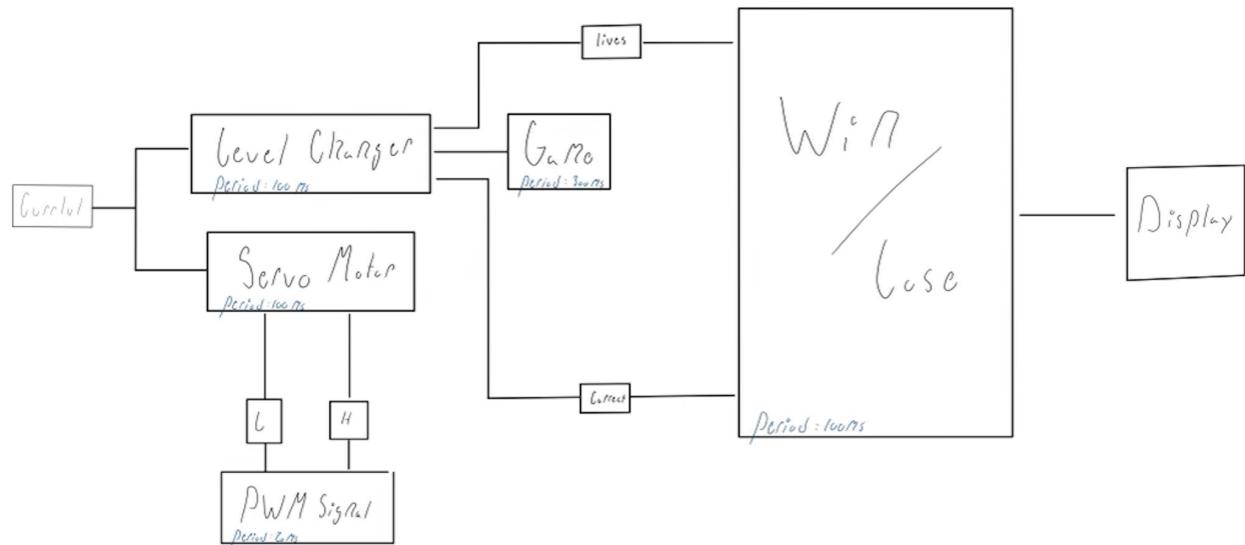
Software Libraries Used

~ LiquidCrystal Header File: I used the library to help me display the dashes of the potential word a player is trying to guess, update the score, and display life points.

Wiring Diagram



Task Diagram



* Global Variables *

int Currentl = 1

int lives = 6

int Correct = 0

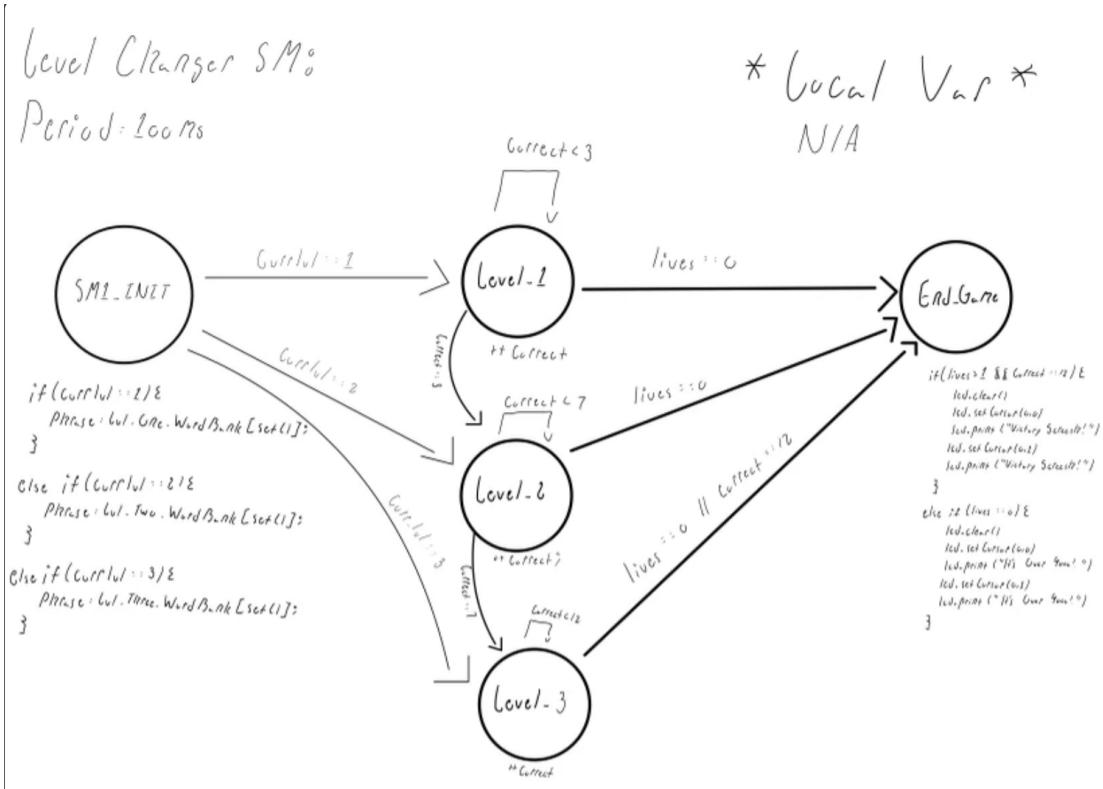
int L = 9

int H = 1

int Servocnt = 0

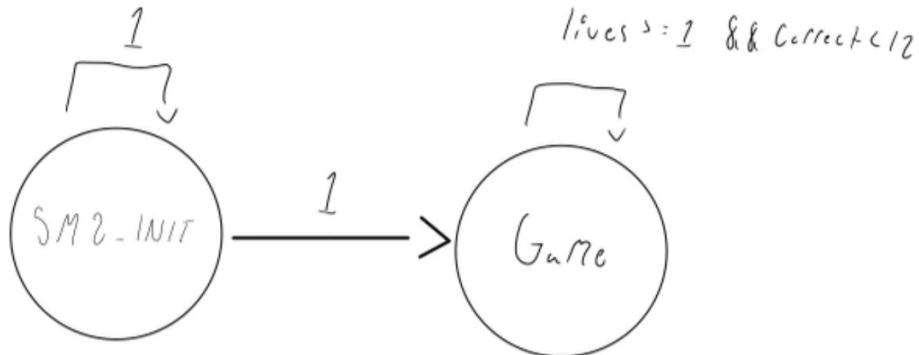
int Servocnt+2 = 1

SynchSM Diagrams



Game SM^o

Period: 300ms



* Local Var *

N/A

```
Screen()  
if (A3 > 800) {  
    ++letter_Changer2  
    if (letter_Changer2 == 26) {  
        letter_Changer2 = 0  
    }  
}  
else if (A3 < 300) {  
    --letter_Changer2  
    if (letter_Changer2 == -1) {  
        letter_Changer2 = 25  
    }  
}  
else if (JoyStick_Btn2) {  
    updateScreen(Correctness(letters2[letter_Changer2]) ||  
                guesses[guessNum] == letters2[letter_Changer2])  
}
```

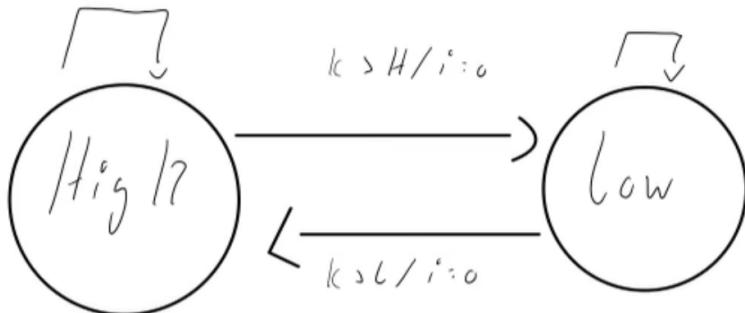
Servo Motor SM^o

Period: 20ms

* Local Var *

int k:0;

k < 14



```
if (currutl == 2 && ServoCnt < 2 / 8)
    for (current = 2000; current < end; current += increment) {
        Servo_Pin = 13
        # SERVOA
    }
}
else if (currutl == 3 && ServoCnt < 2 / 8)
    for (current = 500; current < end; current += increment) {
        Servo_Pin = 23
        # SERVOB
    }
}
```

```
if (currutl == 2 && ServoCnt < 2 / 8)
    for (current = 2000; current < end; current += increment / 8) {
        Servo_Pin = 03
        # SERVOC
    }
}
else if (currutl == 3 && ServoCnt < 2 / 8)
    for (current = 500; current < end; current += increment / 8) {
        Servo_Pin = 03
        # SERVOC
    }
}
```