

CS/EE 120B Custom Laboratory Project Report

Bop-It Recreated

Jayan Patel

December 7, 2022

Introduction:

Bop-It recreated is a game that the user will be involved in. On the breadboard there will be 4 LEDs and according to the LED that is lit the user will have to complete the action for the LED. The actions that this lab will have are bringing an object closer to a sensor, pressing 1 of the 2 different buttons, and tilting a breadboard. If an incorrect action is completed the game will end and restart, if the correct action is done the game will give the user a point (which varies on the task done). The game will go on indefinitely and output the highest and current scores to an LCD screen however I was not able to successfully output the actual numbers just the letters to the screen. The highest score will also be stored by the external EEPROM if the power is turned off.

Complexities:

- EEPROM:
 - I was able to successfully implement this complexity as it does store the highest score when the Arduino has no power.
- Ultrasonic sensor:
 - I was able to successfully implement this complexity as it does correctly measure how far an object is from the sensor.
- NOKIA 5110 LCD:
 - I was able to mostly implement this complexity since I was able to write letters to it, however, when I had to output individual numbers for the score I was not able to do so. I did have technical difficulties with my first screen as it had come in broken (confirmed by Marios in OHs) and my replacement screen only came on December 2nd which did not allow me to figure out how to fully use the screen.

User Guide:

This is a game that a user will interact with. There are 4 different LEDs that correspond to a certain action that will need to be done. When the red LED is lit the user will have to bring an object closer than 5cm to the sensor to register that the action has been done. When the white LED is lit the user will have to push the leftmost button. When the green LED is lit the user will have to push the rightmost button. And finally when the blue LED is lit the user will have to tilt the second breadboard with the tilt ball sensor to register that the action has been done. Once the light has been lit the user will have 3 seconds to complete the action. If the action is successfully completed before the 3 seconds do nothing and wait till a new LED has been lit. If an incorrect action has been done the user will see “Game over !!!” displayed on the screen and the game will restart along with the score. After every game, the highest score and current score will be output.

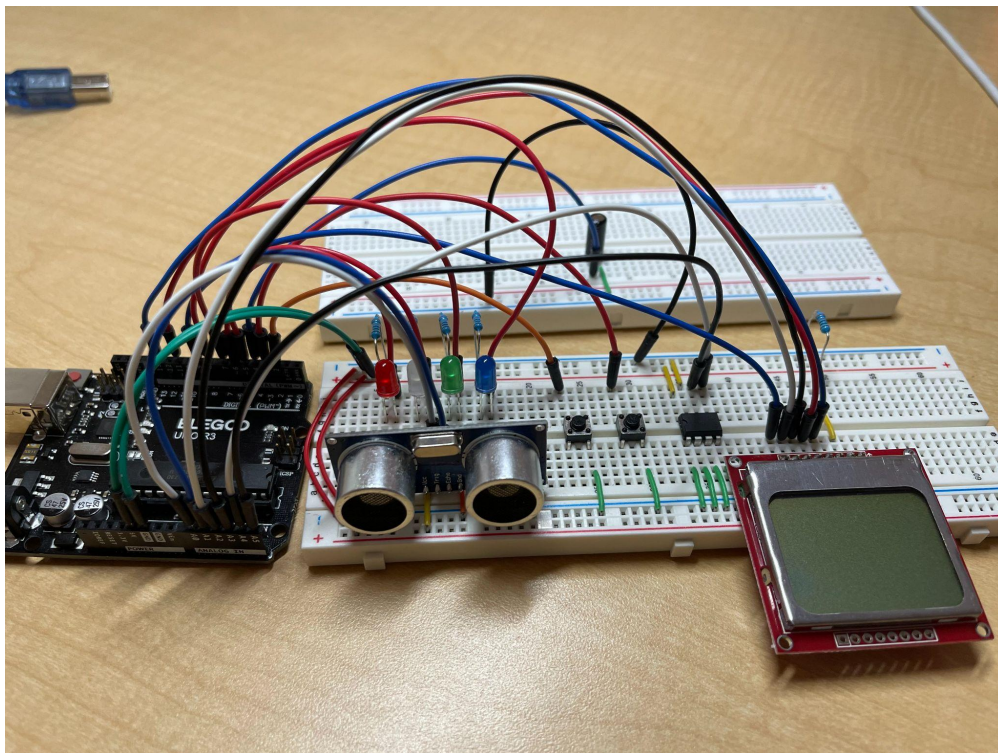
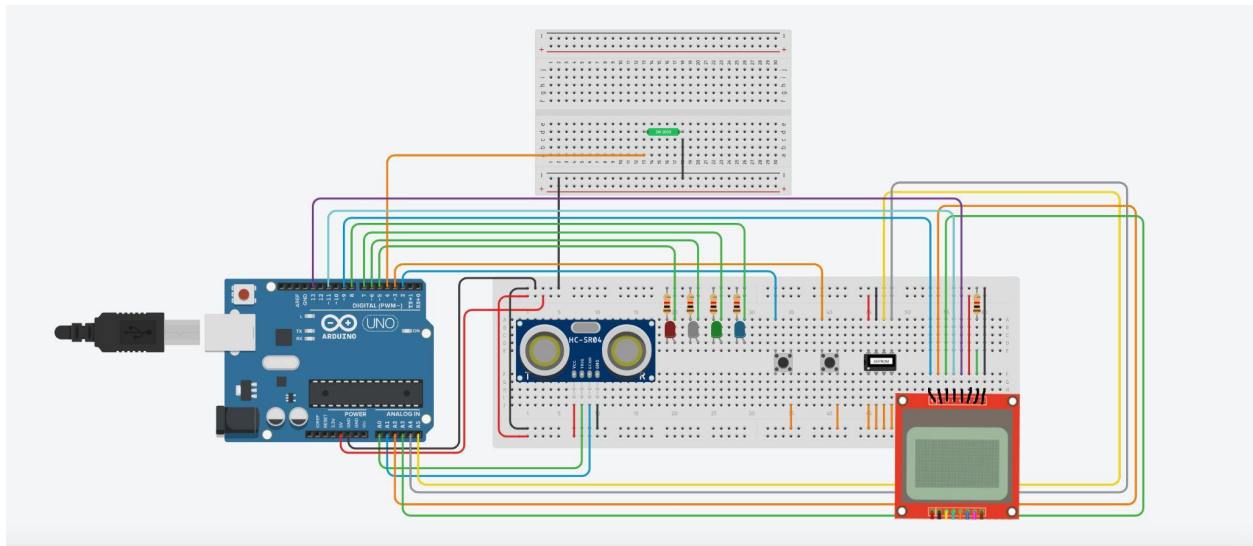
Hardware Components used:

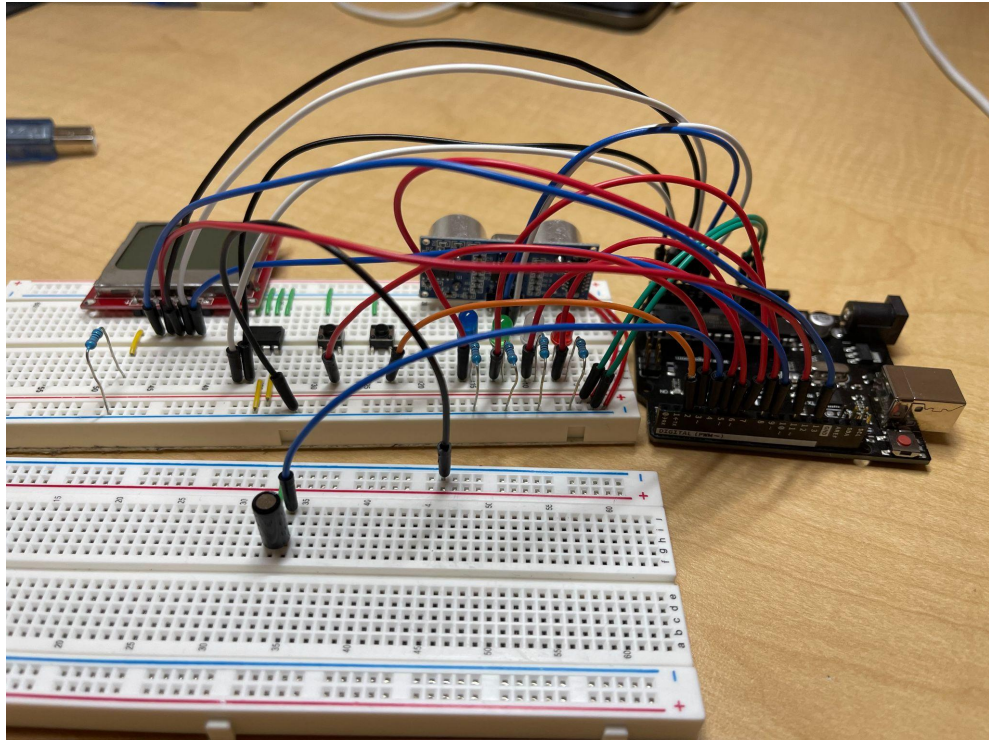
- Arduino UNO R3
- Wires
- 4x LEDs (red, white, green, and blue)
- 5x 330 Ohm resistors
- 2 buttons (Pullup)
- Tilt ball sensor
- Ultrasonic sensor (HC-SR04)
- External EEPROM
- NOKIA 5110 LCD

Software Libraries used:

- Wire library
 - The wire library allowed me to read and write to the external EEPROM.
- SPI library
 - The SPI library allowed me to interact with the NOKIA 5110 LCD so I could turn each bit either HIGH or LOW depending on the letter I wanted to display.

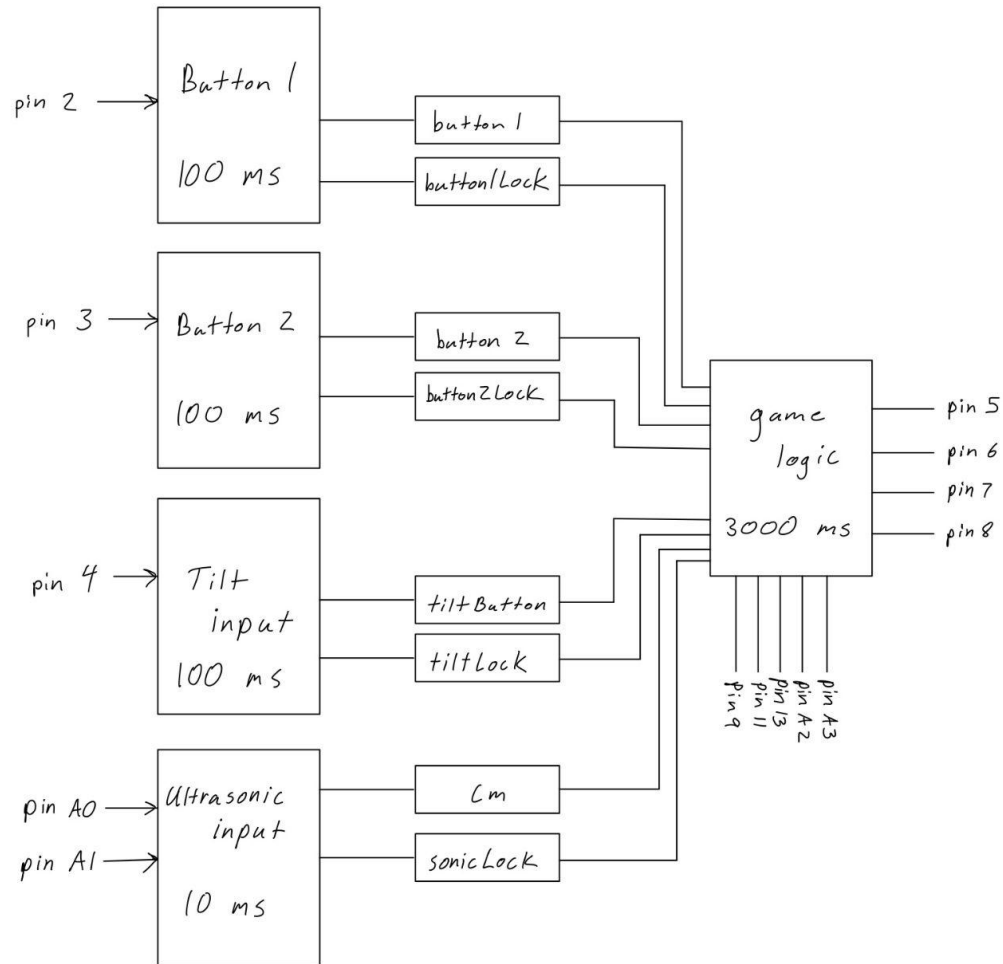
Wiring diagram:





Task Diagram:

Task Diagram



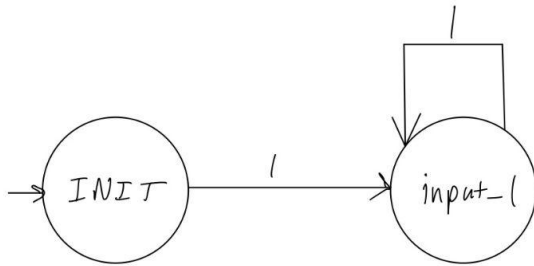
SynchSM Diagrams:

Button 1

period = 100ms;

bool button1 = false;

bool button1Lock = false;



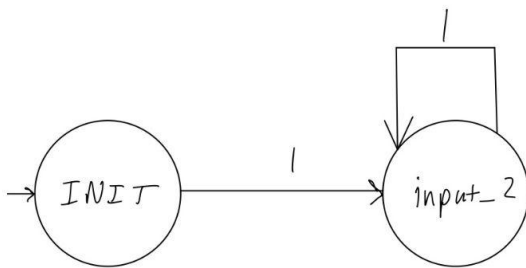
```
if(button1Lock == false){  
    if(digitalRead(2) == 1){  
        button1 = false;  
    }  
    else if(digitalRead(2) == 0){  
        button1 = true;  
        button1Lock = true;  
    }  
}
```

Button 2

period = 100ms;

bool button2 = false;

bool button2Lock = false;



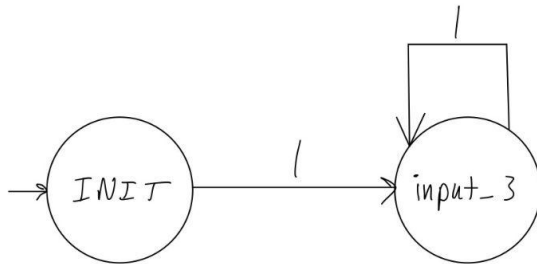
```
if(button2Lock == false){  
    if(digitalRead(3) == 1){  
        button2 = false;  
    }  
    else if(digitalRead(3) == 0){  
        button2 = true;  
        button2Lock = true;  
    }  
}
```

Tilt Input

period = 100 ms;

bool tiltButton = false;

bool tiltLock = false;



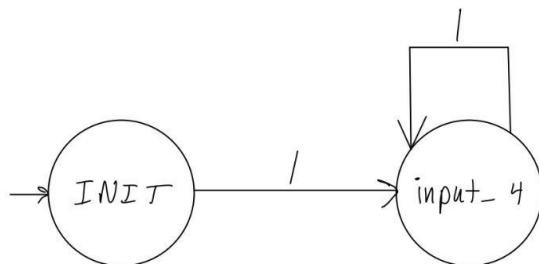
```
if (tiltLock == false) {  
    if (digitalRead(4) == 1) {  
        tiltButton = false;  
    }  
    else if (digitalRead(4) == 0) {  
        tiltButton = true;  
        tiltLock = true;  
    }  
}
```

Ultrasonic Input

period = 10 ms

int cm;

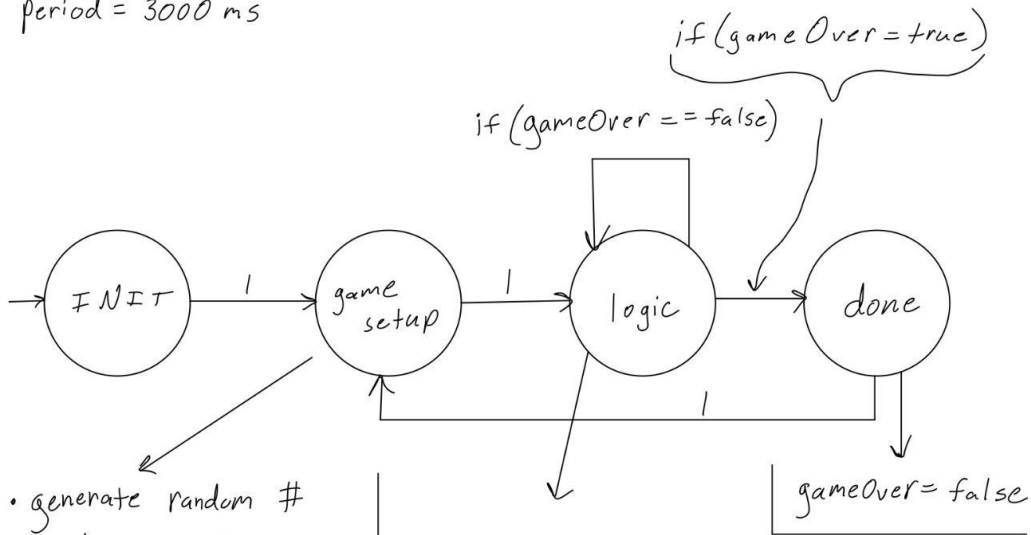
sonicLock = false;



```
if (sonicLock == false) {  
    Send signal to get  
    distance.  
    cm = (duration/2) / 29.1;  
    if (cm <= 5) {  
        sonicLock = true;  
    }  
}
```

Game Logic

period = 3000 ms



- generate random # and output to corresponding LED
- Set all locks to "false"

- Check to see which action was taken
- compare action done to action needed
- if action is true add to the score
- if action is false output game over, check to see if higher than high score, and clear current score.
- Output to NOKIA S110
- generate random # and output to corresponding LED
- repeat logic