Embedded Systems LED Project

By Chase Bennett-Hill

# Project Description

Simple Game. Using an 8x8 matrix the player is at the bottom of the screen and can move side to side. Every X seconds a “brick” led will fall towards the player and they must “catch” it. As the game progresses the rate at which the bricks drop, and their speed will increase. Every time the player catches a brick, they earn 1 point which will be displayed on a 7 segment display. The lives of the player will be displayed using the same 7segment display.

The game will be able to continue until the player loses all lives.

There will also be a buzzer for win / lose sounds.

# Possible Improvements

* The score is limited to 99 because of the number of digits available when the player reaches the max amount the game could play a win sound before restarting.

# Components

* Buttons x 2
* 8x8 Red LED Matrix
* 4 Digit 7 Segment Display
* Buzzer
* Shift Register

# Use Cases

**Game Start**

Press button on Arduino to restart game.

**Game Over**

User runs out of lives. – Play the game over sound

**Move**

User Press button to Move.

**View Score**

The user can see their score with the 7segment display.

**Score Points**

Player can score points by catching the brick – play score sound.

**Lose Life**

Player will lose a life if a brick is missed – play miss sound.

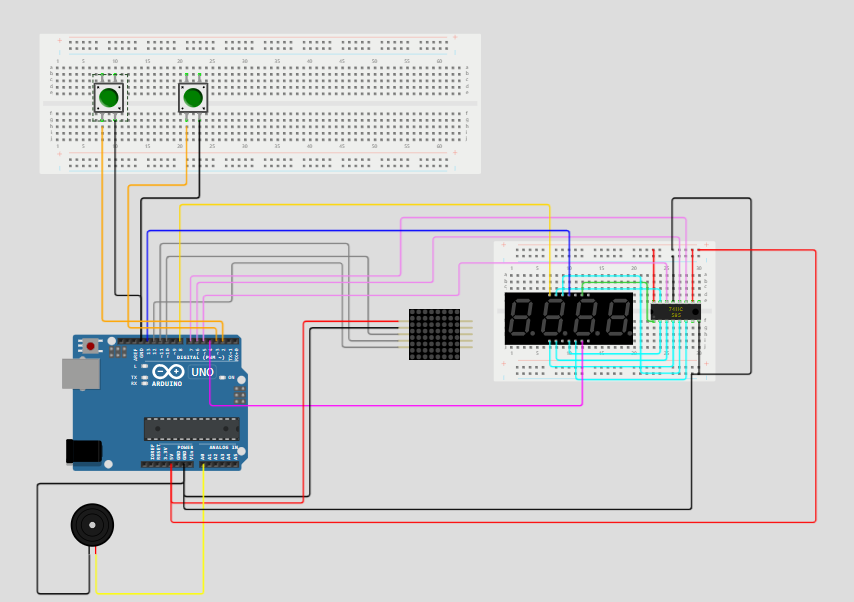
**View Lives**

The user can see their lives with the 7segment display.

# Code

<See FallingBrickGame.ino>

# Breadboard Schematic



[LedSchematic - Wokwi ESP32, STM32, Arduino Simulator](https://wokwi.com/projects/394669071473892353)

# Issues

* One major issue I discovered was about midway through the project.

I had connected the 7-segment display to a shift register as I needed more pins and I found a library that could work with a shift register and got it to display properly.

The issue was integrating it into my project. One of the limitations with it was the library had to constantly run a refresh function. Using delay () would cause the refresh to not be called and cause issues with the display. I ended up making changes to my code to avoid using the delay function as much as possible and instead using.

If (currentTime - lastDropTime >= dropInterval)

This has a very similar effect, and it waits until the time between the last drop and now is more than the drop interval. This way the loop isn’t put on hold during the delay and the refresh can still be run.

* Another issue I had was displaying the score and lives on the same display. I had assumed that I would be able to control each digit separately but the library I had installed only allowed a set of numbers (or characters) to be sent to it at once time. I could only allow which digits were powered but the library assumed they were in order. So, it thought 1 2 and 4 was actually 1 2 and 3.
  + This meant that the fourth digit would automatically be set as the one’s column. My workaround was to display the lives as a decimal and add it to my score so that 5 points and 3 lives would be sent to the display as 5.3 and the display would show 05-3. Where the – is a blank space.