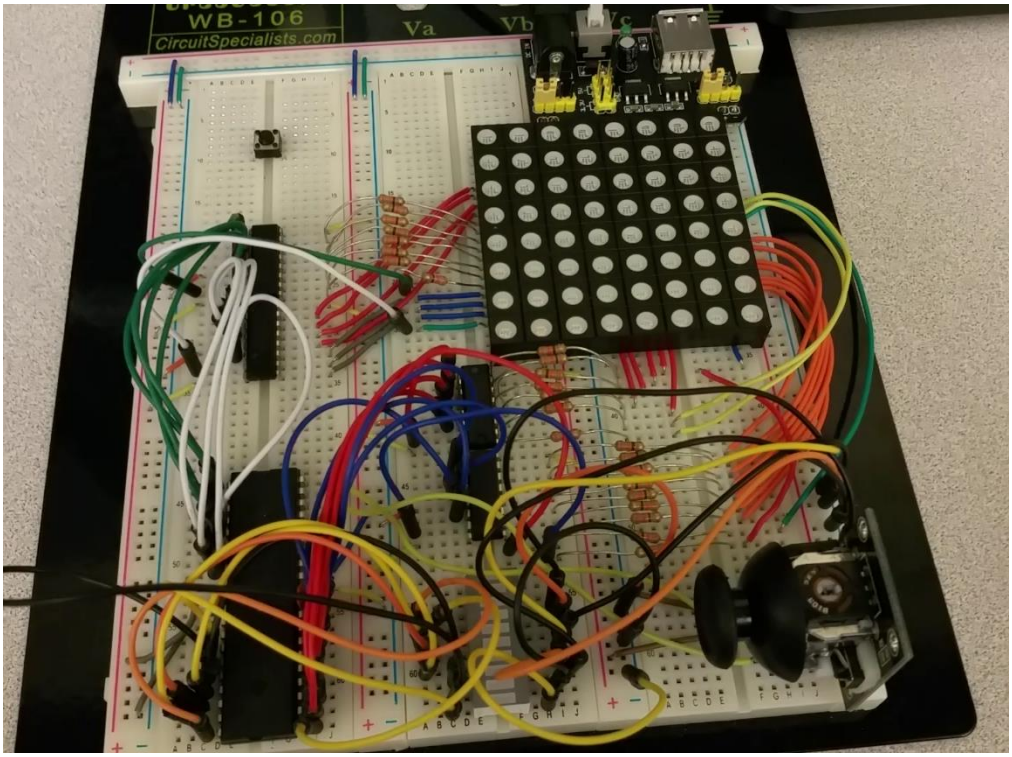
Alex Ardelean

120B Final Project Report

Picture of Wiring Setup:



Explanation:

The micro-controller is setup upright and the shift registers are setup upside down. Port A (top right of micro-controller) is used for analog to digital conversion for the joystick. My joystick has two analog outputs that go to Port A. The output that controls x coordinates of the joystick goes to PINA0, the output that controls y coordinates goes to PINA1.

Port B (the top left of the micro-controller) is used for the rows of the matrix and the color green. Green uses Port B pins 0 – 3 to connect to the inputs of a single shift register, the shift register then outputs to the respective pins on the led matrix. The rows of the matrix uses port B pins 4 – 7 to connect to the inputs of a single shift register, the shift register then outputs to the respective pins on the led matrix similar to how Green is set up.

Port C is used for the red and blue colors of the led matrix. The red colors use pins 0 – 3 and the blue colors use pins 4 – 7. Both connect to shift registers and output to the respective pins on the led matrix. I also connected 330 ohm resistors before each color connects to the led matrix. Finally Port D is used to output to an array of leds. Pins 0 – 7 output to the leds and keep track of score.

How it should Work:

I tried to replicate the game frogger. The joystick controls movement of the "frog" which is a green led light on the matrix. The goal of the game is to go through obstacles to get to the other side of the road. The obstacles that I added were cars which are red dots that move left or right through the screen. I also added water with objects on top to help you pass. The objects that help you pass are stationary lily-pad and logs that move either left or right. The first row and last row do not spawn obstacles but rows 2-7 do.

Everything is randomly generated there's a 1/3rd chance that either lily-pad or logs or cars form on a given row. Once a row type is generated there's a 1/3rd change that either a car, or log will form on that column and a ½ chance that a lily-pad will form on that column. A car or log can only be a maximum size of 3, to guarantee this after a car or log is formed we make sure that the next column is not a car or log. I allow the first column generated to be either a car or a log and I also allow the last column generated to also be a car or log, this gives a small chance that a car or log larger than 3 pixels can form. When a car or log is generated there is 50 percent chance it will move to the left and 50 percent chance it will move to the right. Each log or car row is generated 16 columns wide.

There can only be a maximum of 4 lily-pads and at least 1 lily-pad in a column of a row of lily-pads. A lily-pad can't be next to another lily-pad in the column. A lily-pad row can't be next to another lily-pad row. If the level generated has a row that contains lily-pads then we pick a random row that contains lily-pads, and a random lily-pad from that row. On top of the randomly chosen lily-pad we add a yellow token that the frog needs to collect before moving on to the next level. If a row contains logs or lily-pads every column that is not a log or lily-pad is a water block.

To move on to the next level all you have to do is reach the top row of the matrix. Once you reach the top row of a matrix a new level is generated and the matrix transitions row by row removing the old level and adding the new one in simultaneously until you hit the bottom row just like how you started in, in the first level. When you complete a level your score increases by one. If you touch either water or a car you get reset back to the beginning of the level and your score decreases by 1.

Bugs and Issues:

There are no known bugs or issues. If you want completely new levels seed the srand function with a different number.

Working Links:

https://youtu.be/u_J2-dzd43k

State Machines:

unsigned char cars = 0, pads = 2, logs = 1, left = 0, right = 1, token = 0, rbcsize = 16, score = 0 ;
unsigned char movementlock = 0;
unsigned char rowtype[7] = {0}; unsigned char frog[2] = {7, 3} //frogs position
unsigned char roadblocks[7][16] = {{0}}; unsigned char rowdirection[7] = {0};