Jonathan Hess
Section 3.2

# Week 10 Lab Report: Free Lab 1

Lab Report Rubric

Category	Student Score	Grader Score
Organization		
Appropriate sections	1/1	/1
Appearance and formatting	2/2	/2
Spelling, grammar, sentence structure	1/1	/1
Work		
Experimental procedure	2/2	/2
Results (data, code, figure, graph, tables, etc.)	1.5/2	/2
Conclusion	1.5/2	/2
Total	**9/10**	**/10**

Introduction

For this free lab I worked on the challenge to make a rock paper scissor game. I decided on using the arduino with a countdown using the 7 segment display and using push buttons as the input.

Systems Level Perspective (When applicable)
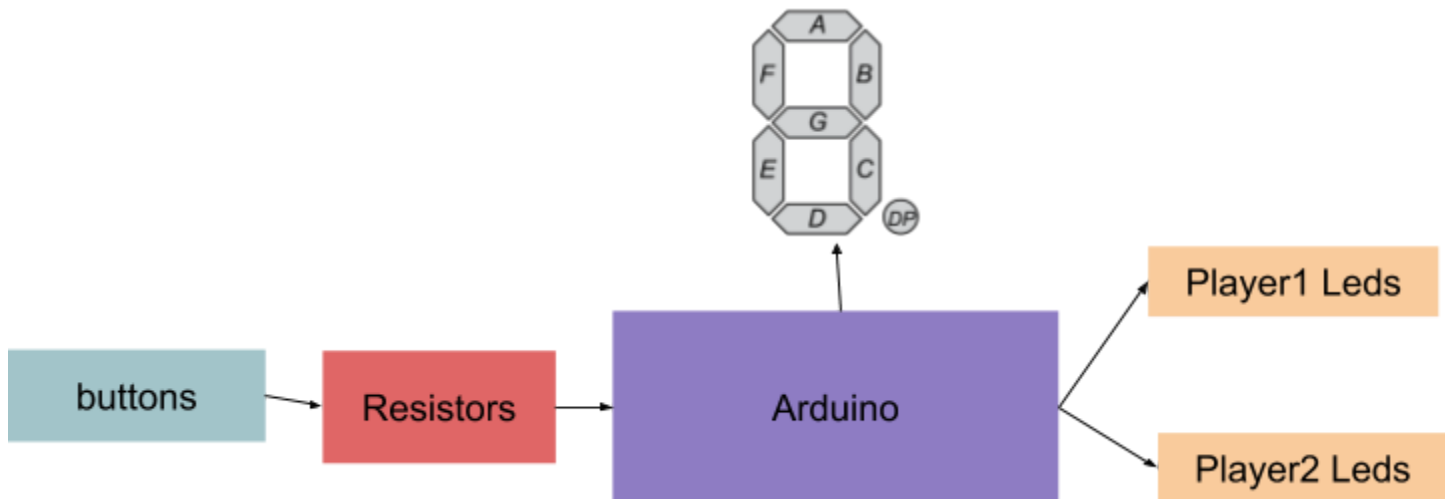
Procedure

State the Task/Problem/Question Attempted
I was attempting to make a rock, paper, scissors game with arduino with a countdown.

Procedure
My original plan for the game was to have 3 leds for the timer counting from 3 to 0. (Rock Paper, Scissors, Shoot!). I then saw the 7 segment display and thought it would be a fun challenge to make it count down. So I first started with the display.
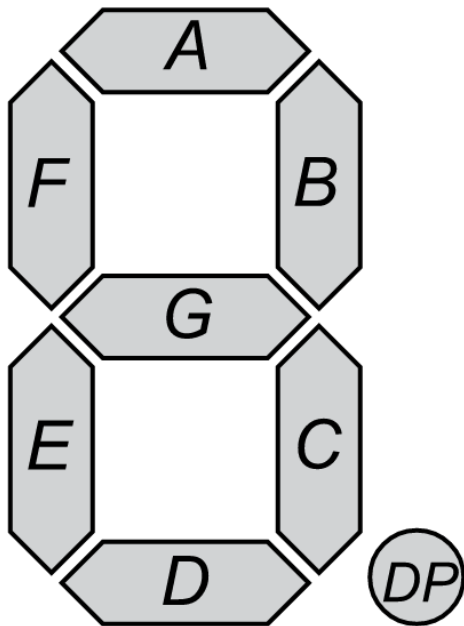
I first started with the display's parts, the type I had shared an anode so I connected the pins to the arduino and set pins 2-8 to output. Then I connected the display's ground pin to ground with a 220 resistor.

For the code I set the up the pins to match the segments

```
const int a = 7;
const int b = 6;
const int c = 4;
const int d = 3;
const int e = 5;
const int f = 8;
const int g = 2;
```

Then I made an array of values telling what segments of the display to turn on (default would be off)

```
int numbers[4][7] = {
        {a,b,c,d,e,f,0}, //0
        {b,c,0,0,0,0,0}, //1
        {a,b,g,e,d,0,0}, //2
        {a,b,g,c,d,0,0} //3
};
```

You could continue past 3 if you wanted to use the display for all numbers 0-9 by just adding onto the array.

After setting up the array I created the code below to run through the array turning on all the segments needed. I could see a problem with making empty values 0 because I might have wanted to use the 0th pin for something else but I didn't need to change it in the end.

```
void numberWriter(int num){
  for(int i=0; i<7; i++){
    digitalWrite(i+2,LOW);
  }
  for(int i=0; i<7; i++){ //error because of overflowing into next array
    digitalWrite(numbers[num][i],HIGH);
  }
}
```

I encountered a large problem here because in the for loop I originally had "i<=7;" because I was thinking that it needed to go up to 7 but it was going through 0-7 which is actually 8 values which threw an error that I ignored. This caused it to overflow into the next row of the 2d array. This is because an array actually works like a pointer that just breaks up the values for you, so overcalling leads into the next value which would be the next rows first column.

Now that was set up all I had to do was add the leds and the buttons. I then got concerned that I was going to run out of digital pins on the arduino so I decided to try out a led matrix. I used a matrix for the display lights because nothing would be happening while they were on.

Ar Pin 5	Ar Pin 5

Ar Pin 5	R1	R2
Ar Pin 5	W1	W2
Ar Pin 5	G1	G2
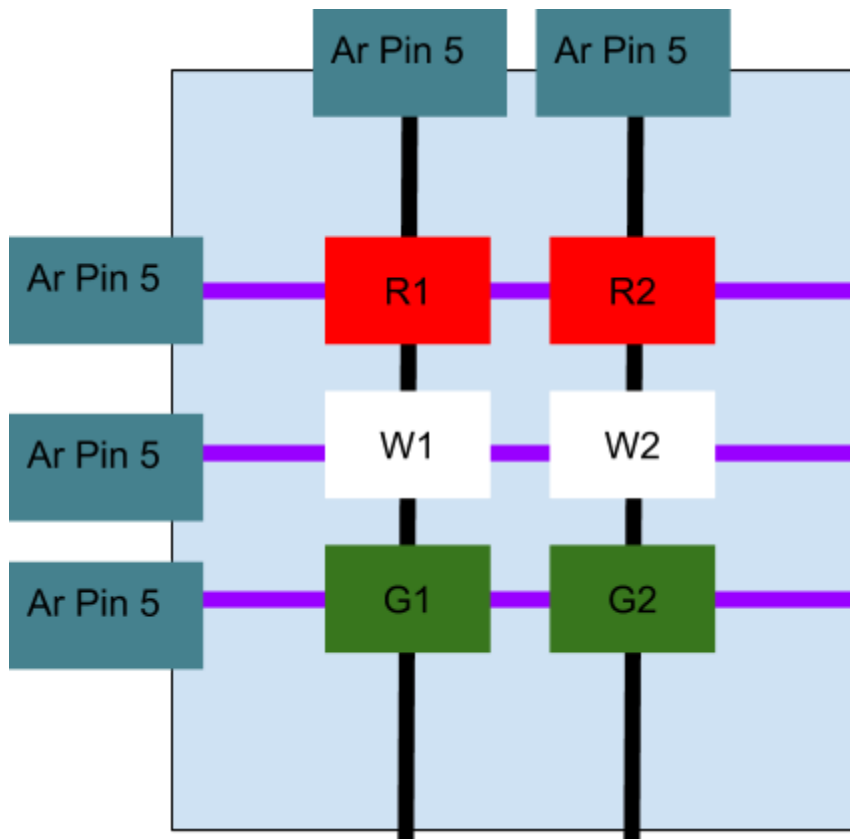
I set up the Matrix like this so that way I use 5 pins rather than 6. In this case I don't save many pins but if I had a much larger matrix I would save a lot more pins.

This works by pulling pins 11, 12,or 13 high (based on the selection) and pulling 10 or 9 low while keeping the other (if 10 then 9, if 9 then 10) high. This means that only one of the LEDs will have a potential difference at a time so you can control 2 led cathodes with one pin.

The code for the sensors will find what led to turn on and then choose the correct pin for both players. (defined as player1Pin and player2Pin)

```
for(int i =0; i<100; i++){
  digitalWrite(10,LOW);
  digitalWrite(9,HIGH);
  digitalWrite(player1Pin,HIGH);
  delay(10);
  digitalWrite(player1Pin,LOW);


  digitalWrite(9,LOW);
  digitalWrite(10,HIGH);
  digitalWrite(player2Pin,HIGH);
  delay(5);
  digitalWrite(player2Pin,LOW);

}
```

Now the last thing I needed to do was input from the players. The only problem is that I didn't have enough digital pins to connect one button to each or even enough to make a matrix. I then decided on using a potentiometer, after a bit of work I got the sensor wired up. I then wanted to see the values at different angles of the potentiometer so I typed the following code.

```
Serial.println("sensor value: " + player1Sensor);
```

This caused an error because rather than printing a string with the number added to it, it added the number to the string's value.

The TA then recommended using the buttons rather than potentiometers because you can set the buttons with resistors such as:
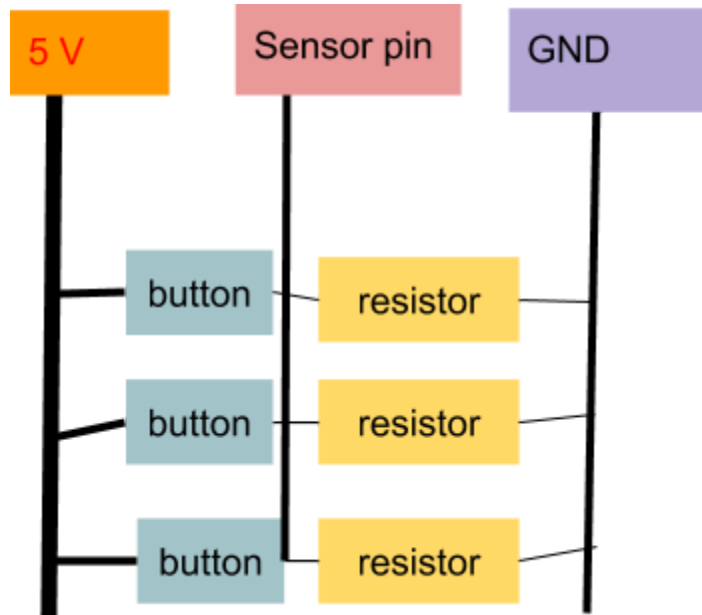
Each of the resistors would have a different value, unfortunately I couldn't get them both to work at the same time because I was using too small of resistors so it drew too much current.

I wrote code that would see the differences in the values and it works for one at a time but not with 2 plugged in. I think the problem was the wiring because I was getting consistent data with the first one.

Results

Results

I got to use a Led Matrix as well as mess around with the analog pins. I got many ideas from this lab for future projects and got to use arrays in code as well as for loops. I really wish I could have gotten the analog sensor to work and I think it was probably because of my wiring. I will finish the device on my own and provide a follow up.

Video of Counter working

The Complete Code

Conclusions and Reflection

I cannot wait for the next free lab to work on lots of projects, I don't know if I have all the parts needed for a project me and my friend have talked about though. It requires an accelerometer to detect when you have hit a chest that activates a piston to open it. I also think I should work more with the hardware because that is where I struggle more so my projects should probably be more circuit based so I can learn more.