Self Leveling PCB System

Dates: 03/04/2024-03/30/2024

Alexander Suen

Table of Contents

Circuit Specifics

Intended Use

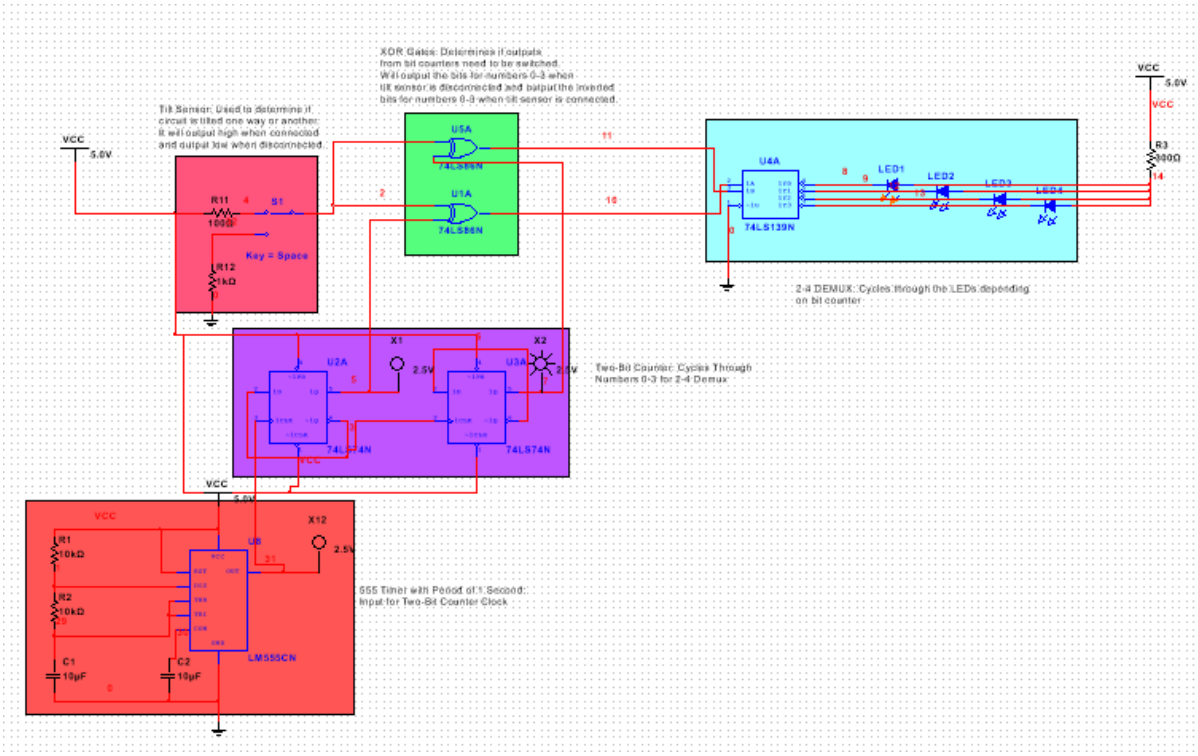
The intended use is that the circuit would light up the LEDs from left to right if tilted left, and if the circuit was tilted to the right, the LEDs would light up from right to left. This would serve as the automatic leveling system because the tilt ball sensor only disconnects and connects when the tilt is greater than 15 degrees in one direction. I designed this circuit on my own and didn't use any online references or resources.

Circuit Description

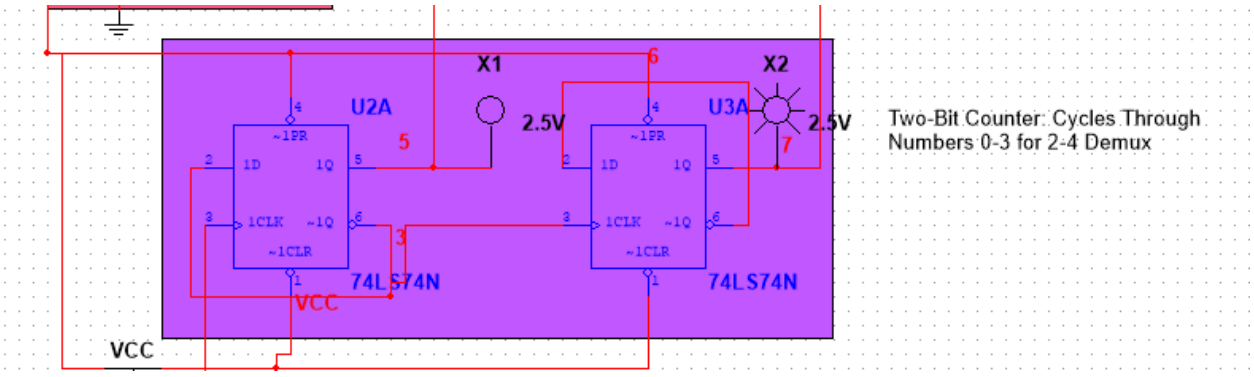
For my circuit, I used two XOR gates, a SW-200D tilt ball sensor, one 2-4 DEMUX, two flip-flops for a 2-bit counter, a 555 timer, and 4 LEDs as outputs. The purpose of the 2-bit counter was to cycle through the numbers 0-3 and act as inputs for the XOR gates. The 2-bit counter was powered by a 555 timer that was connected with 2 10k ohm resistors and 2 10 uf electrolytic capacitors. This gave the 555 timer a period of approximately 200ms and a duty cycle of 67%. It would serve as the clock input for the first flip-flop. Each XOR gate was connected to one bit, and the output was from the tilt ball sensor. The tilt ball sensor connects and disconnects a part of the circuit. If the tilt ball sensor was disconnected, the XOR gate would output the bit-counters normally from the numbers 0-3. Otherwise, the XOR gate would output the inverted numbers from 3-0. This would connect to the inputs of the 2-4 DEMUX and, depending on the result of the tilt ball sensor, would cycle through outputs in opposite directions and light up the LEDs differently. The 2-4 DEMUX outputs are low, so the LEDs are connected to a 300-ohm resistor in parallel and a power source in order to light up.

Simulation Images

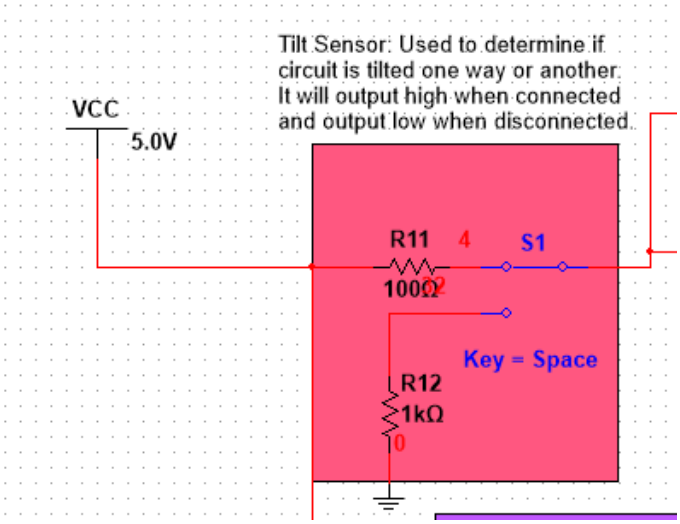
Full Simulation Image



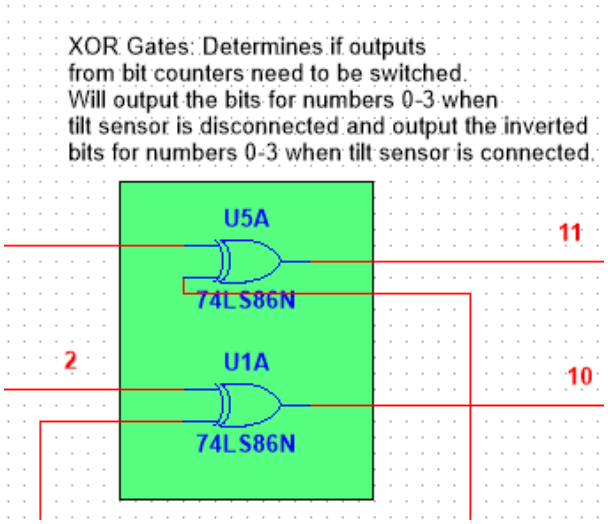
2-Bit Counter



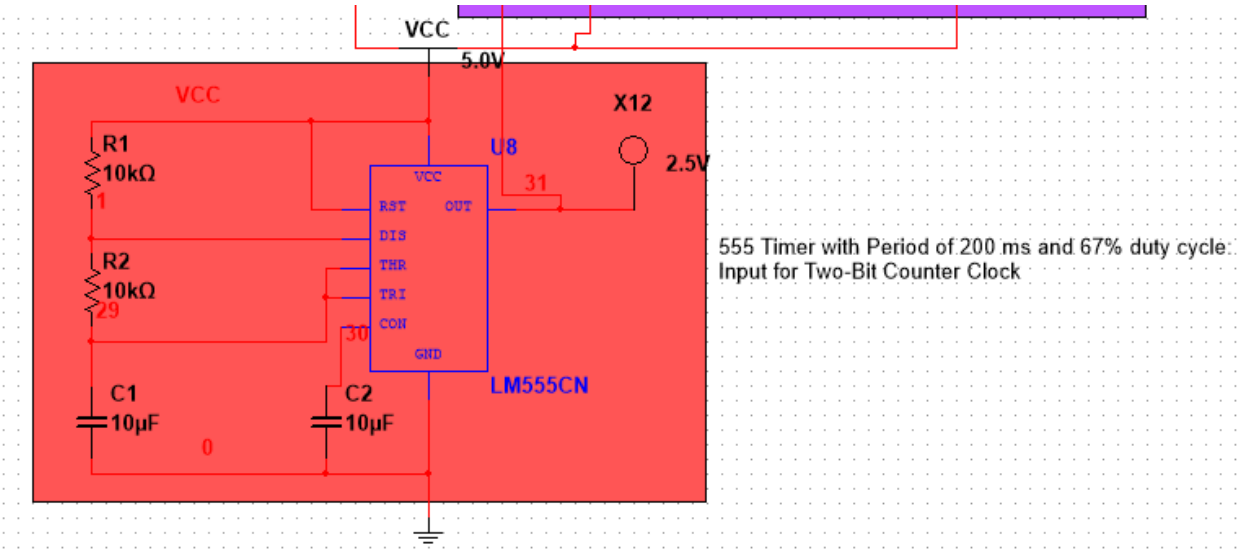
Tilt Sensor



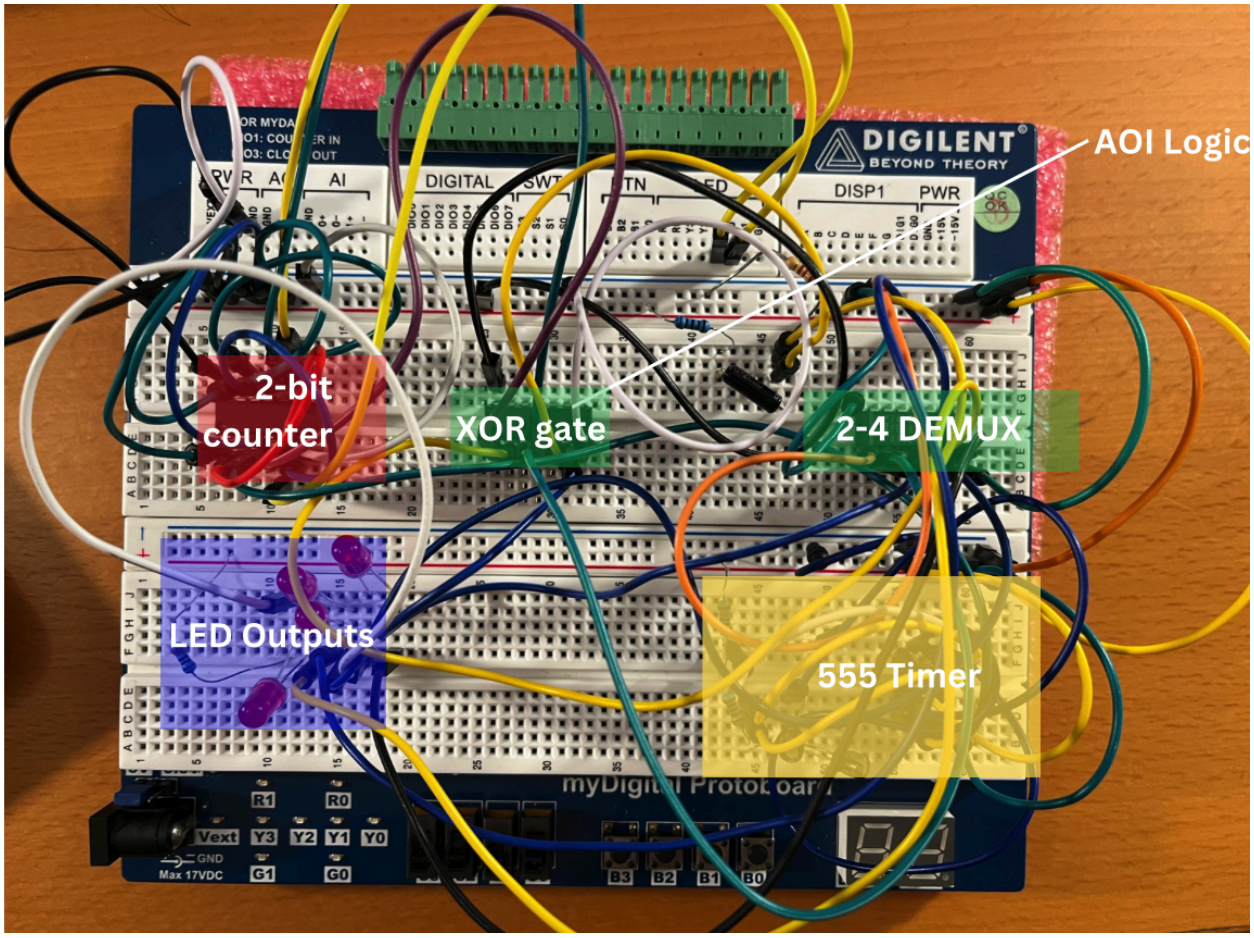
XOR Gates



555 Timer



Final Breadboarded Circuit



Video of the Breadboard operating

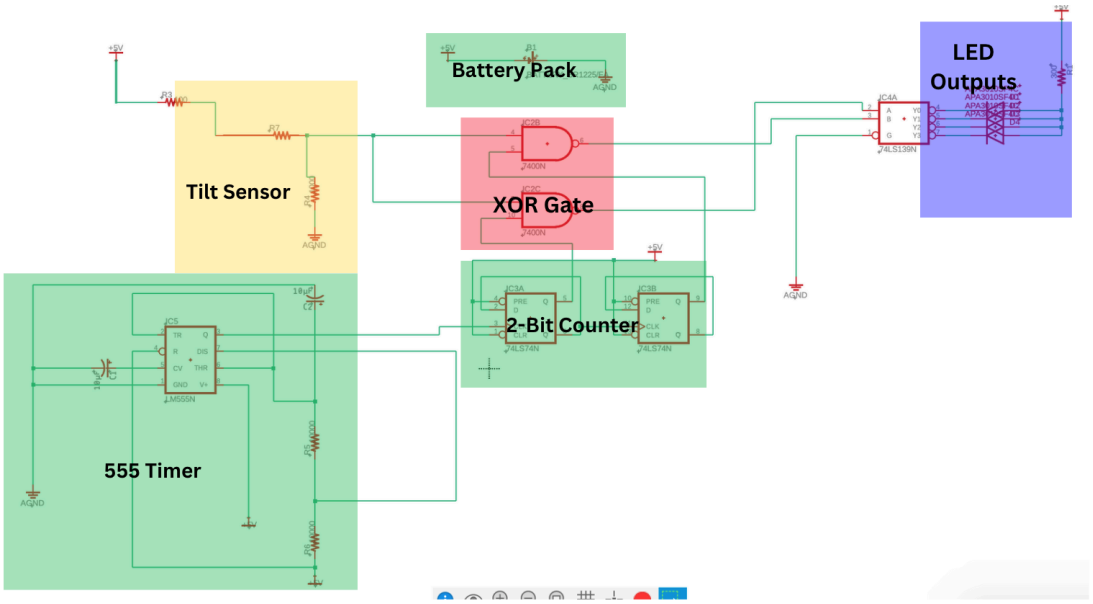
Fabrication Design

Procedure

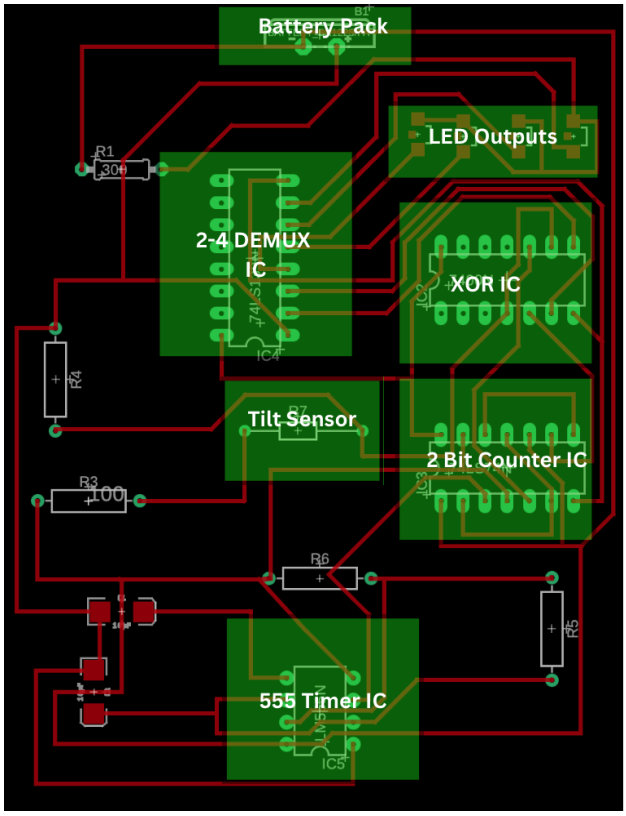
My PCB design was done in Fusion 360 using an electronics design. I first used my Multisim design as a reference for my schematic design in Fusion 360. Since not all the components I was going to use were in the Fusion electronics library, I substituted some components with others. For example, I used a NAND gate instead of an XOR gate, but the component sizes were the same so it would still work. I also had to create my own custom schematic for the tilt sensor, using a resistor as a reference. I made the two pin holes farther out, approximately 14mm to fit the component. Then I moved on to laying it out on the PCB design. I laid out the components many different times trying to find the most optimal spacing. Then I routed them using the routing feature in Fusion 360 to connect

the wires to their respective destinations. After finishing routing, I exported the PCB layout as a Gerber file and imported it into ViewMate for conversion into a PDF and eventually being printed.

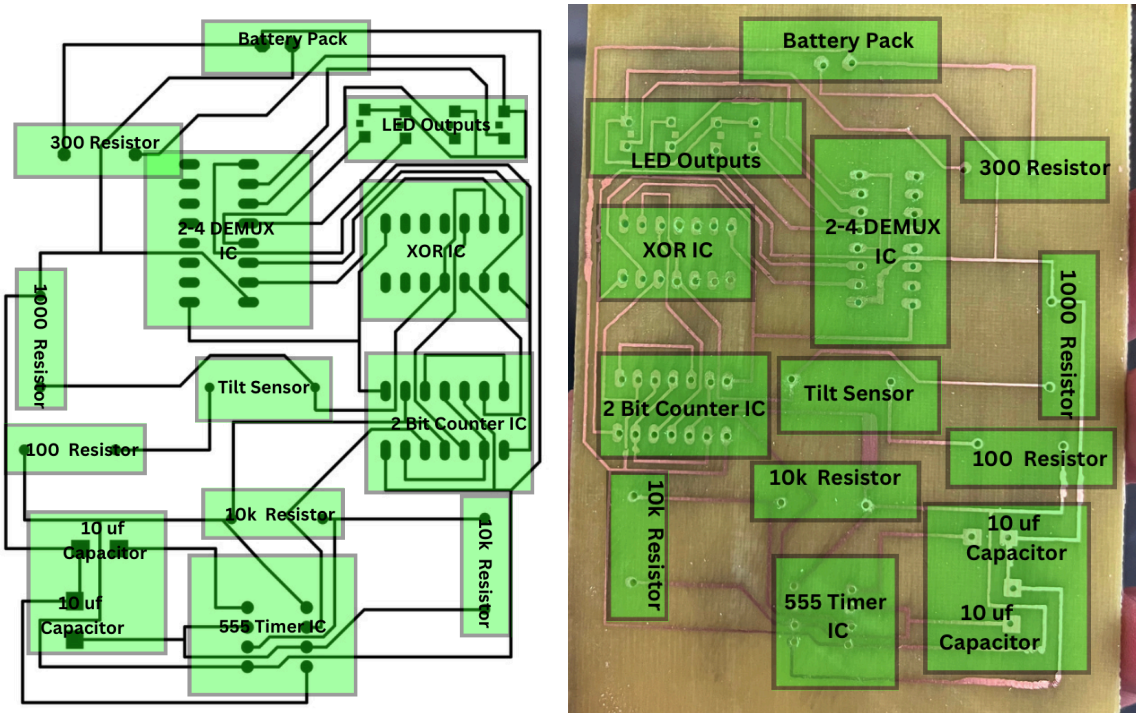
Final Schematic Design



Final PCB Layout



Final Copper Mask Labeled

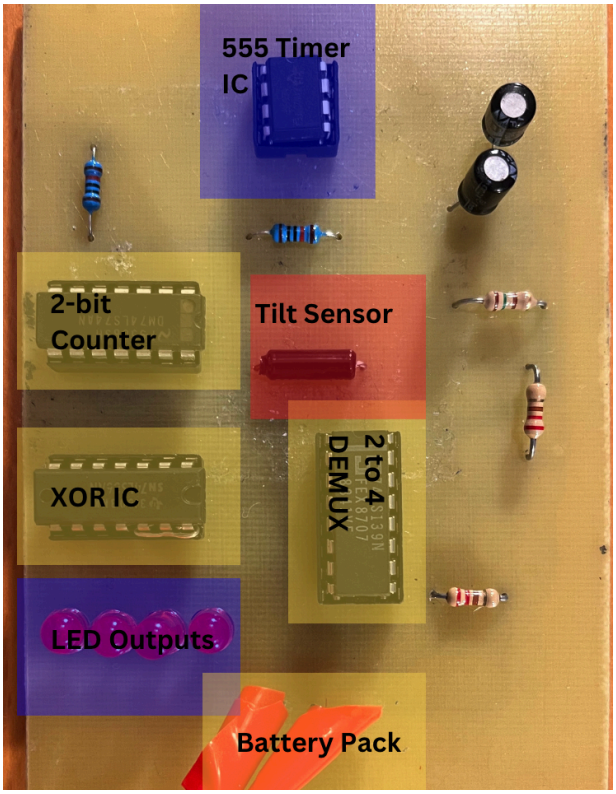


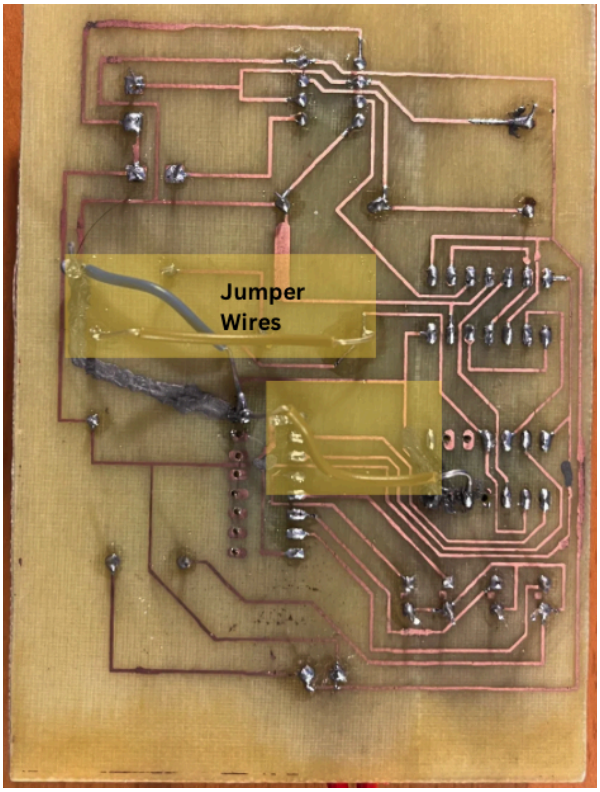
Final Solution

Final Solution Description

I created a PCB that had one LS7485 XOR IC, a SW-200D tilt ball sensor, one 74LS139 2-4 DEMUX IC, one 74LS74 D Flip Flop IC, a LM555 timer IC, 4 LEDs as outputs, 5 resistors varying from 100 ohms to 10k ohms, and 2 10uf electrolytic capacitors. There were also various jumper wires to jump across the board as the board was only single-layered and not double-layered. The 555 timer powers the clock component of the 2-bit counter, allowing it to count from 0-3. Then I paired the outputs of the bit counters with the tilt sensor voltage in the XOR gates to determine if the bits needed to be inverted. The outputs of the XOR IC were connected to the inputs of the 2-4 DEMUX, which in turn outputted a low to the corresponding LED, allowing them to light up sequentially. The direction depended on which way the board was tilted. My PCB worked exactly as my breadboard and simulation did and the functionality was all correct. When completing the project, I ran into multiple problems that I had to solve. One of them was the copper traces being broken when soldering because the soldering iron was too hot. To solve this, I had to use jumper wires and also trace out some of the copper back on with a nickel pen. Also, in the Fusion schematic and PCB layout, the software did not automatically connect ground and power to my XOR IC, and there was no IC version of the component in the library. This led to my XOR IC not having any power or ground, and I had to manually draw out traces after etching and use jumper wires to connect the VCC pin and Ground pin to their correct sources.

Final PCB Top and Bottom View





[Video of the PCB operating]