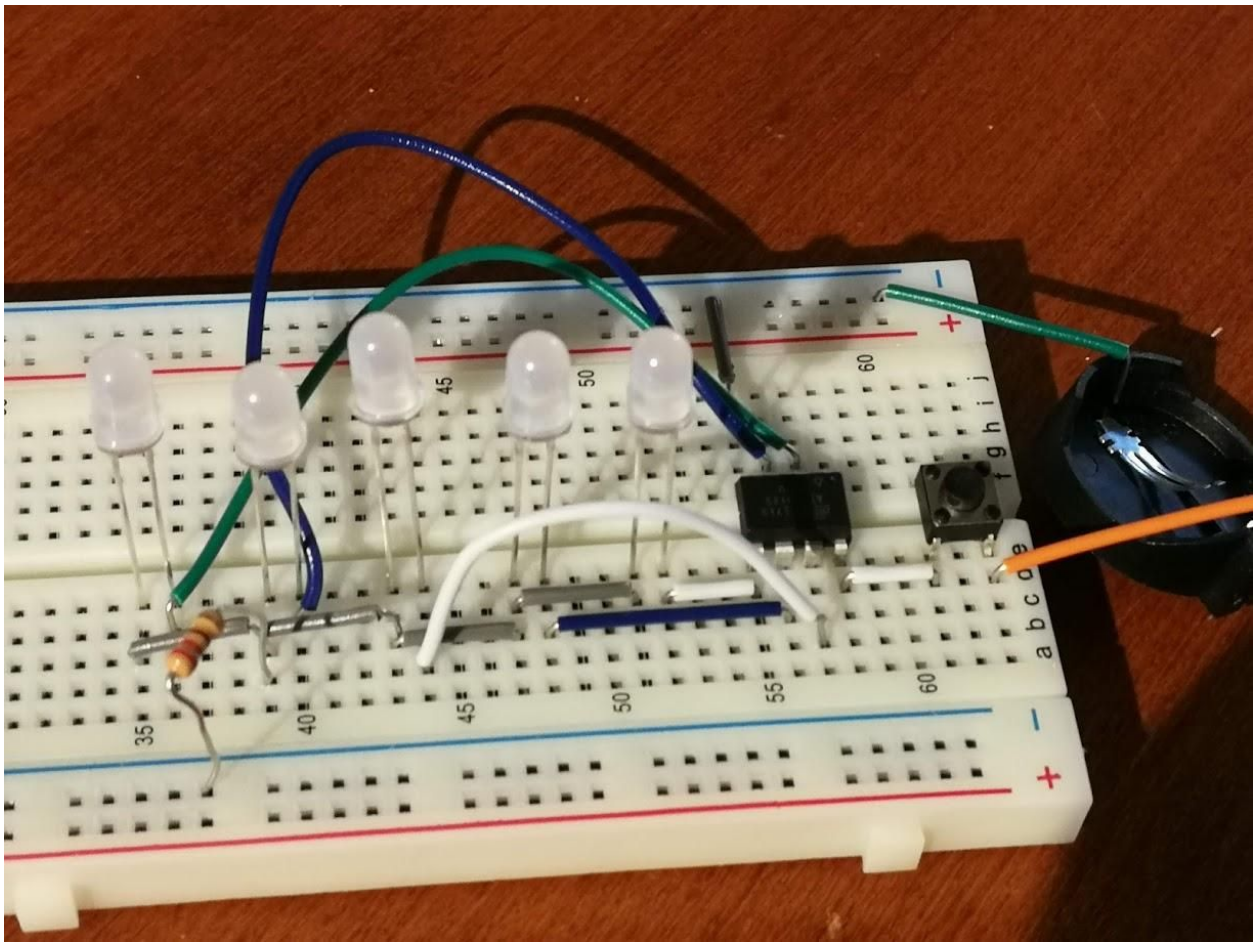
CART 360 - Etude 2

Jennifer Powroznyk

Part One

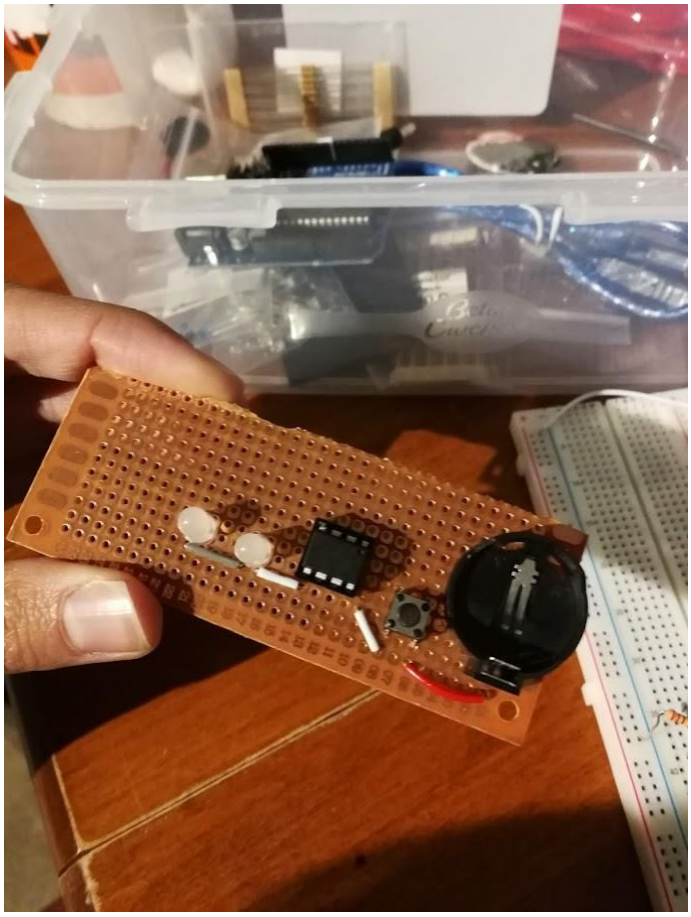
Understanding my objective with Etude 2 was a little daunting because I had missed the class where Etude 2 was assigned. Looking at my schedule, the Sensor Lab's opening hours just happened to align perfectly with my other classes and work. This was really concerning as I knew I needed a soldering iron and to program my ATtiny85. I watched several videos on how to solder Monday night and prepared to purchase an iron Tuesday. I also prepared my circuit on a breadboard Monday night.



Close up of planned circuit.

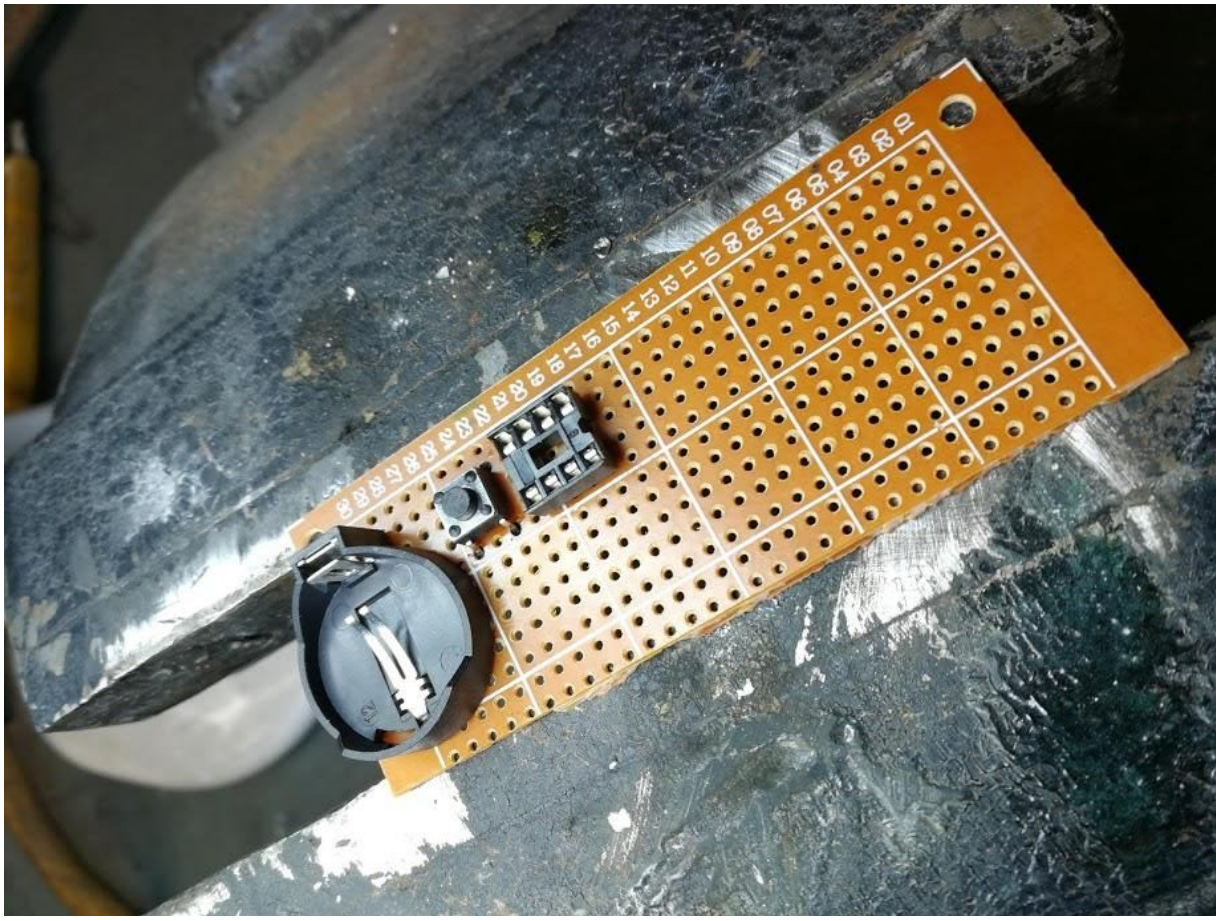
My set up on the bread board was almost identical to the Fritzing diagram provided in the Etude 2 document. It initially made it a little easier for me to translate to the diagram into a physical circuit by copying the exact representation. A major problem I had was an unprogrammed ATtiny 85. Because the chip wasn't programmed, I wasn't able to trust that my circuit was

working so easily. Instead of watching the lights flicker once I pushed the switch I saw nothing. I knew this would be the case having not programmed my chip, but it was concerning to be working blindly all the same.

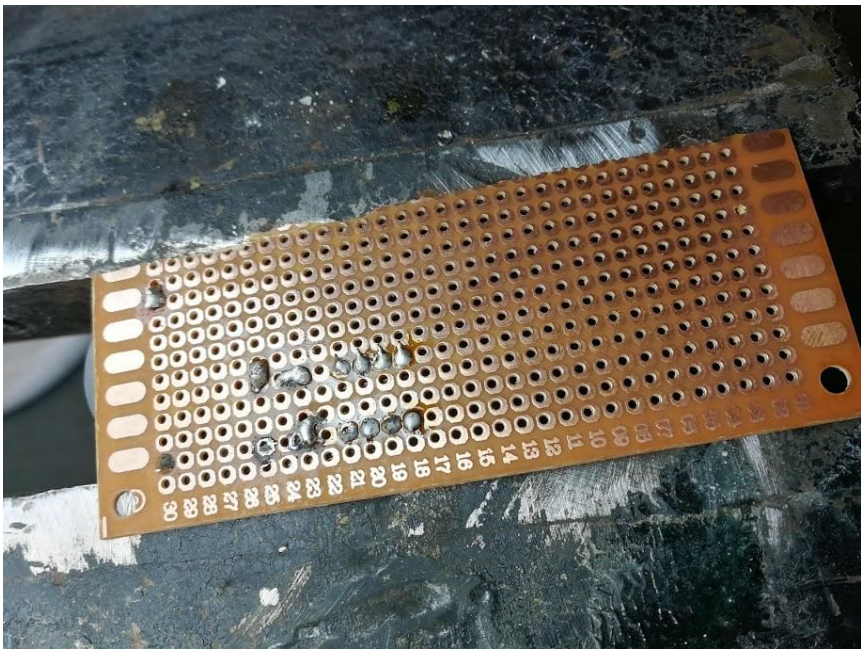


The photo to the left demonstrates some initial planning of the core components of my circuit. This is incomplete and not the design I stuck with but I was mainly just playing around to get a sense of how I would organize my board.

Tuesday morning before work I bought a (seemingly) fair priced soldering iron, solder and a desoldering pump from Rona. After work that day I drove over to my dads house. I called to ask if he had a multimeter that I could use to test for continuity since I hadn't been able to program my chip and I wanted to be sure everything would be connected properly once I began soldering components together.

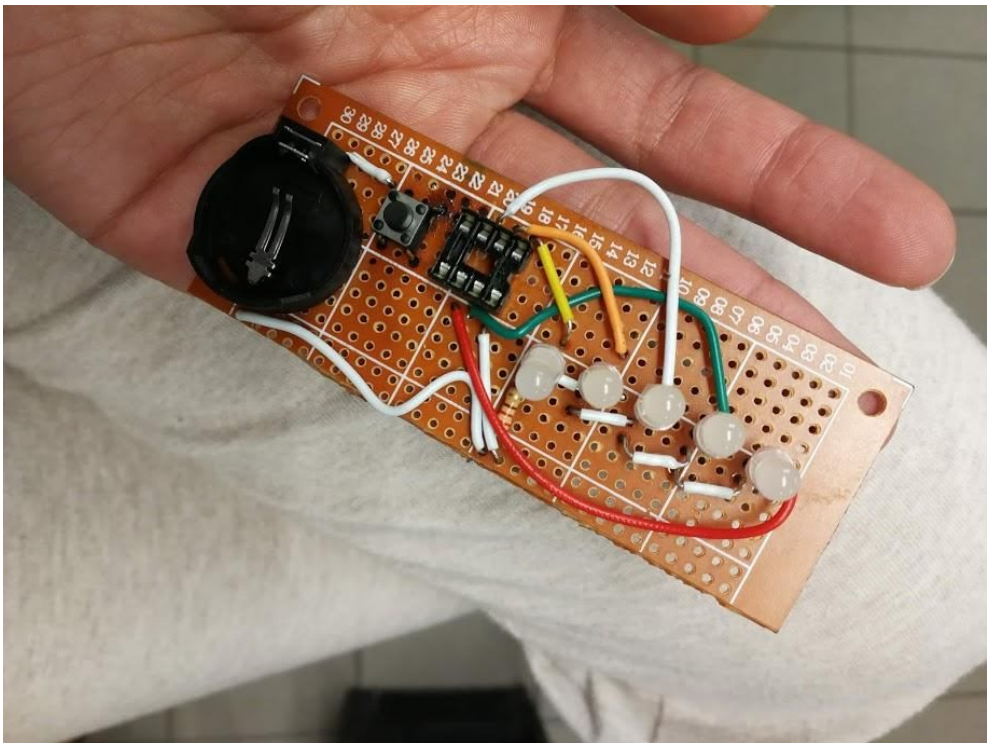


I began placing my core components on to the perforated board. I was pretty intimidated and scared to mess up. My plan was to get all the main components on and then wire them together in the necessary paths. Soldering on the cell clip, switch and chip connector were pretty scary as I wasn't super comfortable with soldering or my design yet.



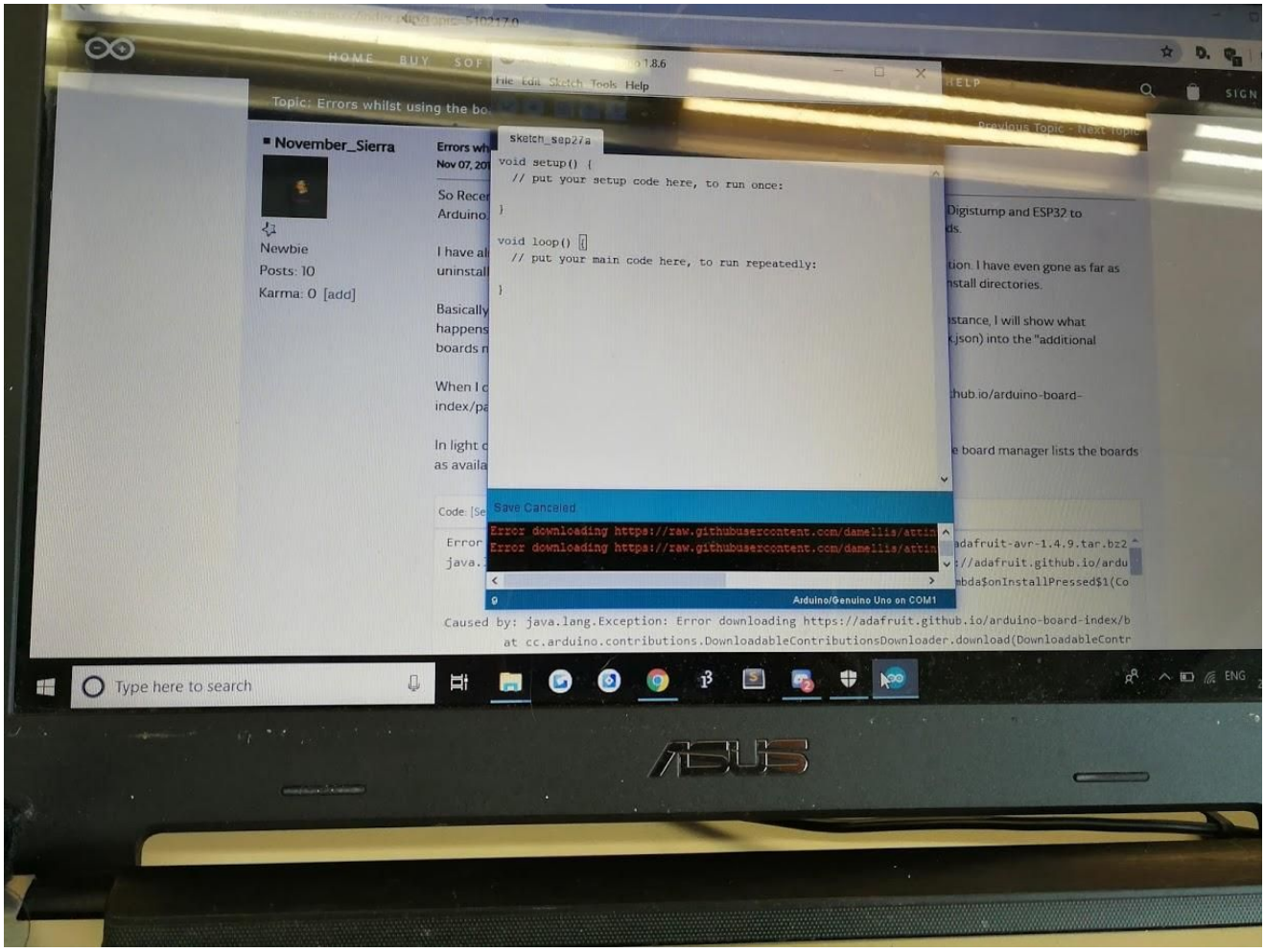
To the left are my first attempts at soldering ever. Here I attached the ATtiny 85 connector, the switch and the cell clip. I didn't connect any of the components to each other at this point. They're all pretty ugly looking and weren't visibly close to any of the professional work I had seen online, but it was the best I could do.

This is a photo of my completed chip. Once I had all my major components in I began connecting them based on the plan provided. I tested the connections for continuity so that I could feel confident everything was properly connected without knowing if it worked for sure. My soldering job was pretty yucky. I nearly melted the switch through the perforated board, but otherwise it looked alright to me!

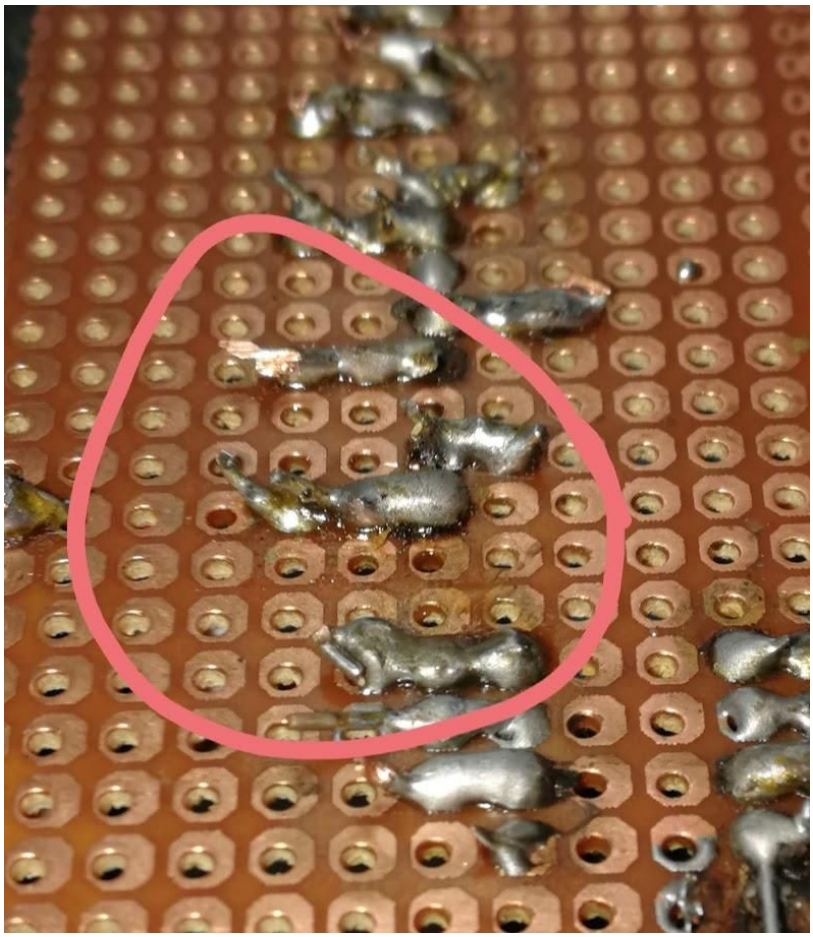


I ended up taking a day off work so I could head to the Sensor Lab on Thursday afternoon. I installed the Arduino IDE Thursday morning and followed the steps provided in the Etude 2 document so that I could upload the sample code to my ATtiny 85. To my frustration I kept getting this error when updating Arduinos preferences:

“Error downloading https://raw.githubusercontent.com/damellis/attiny/ide 1.6.x-boards-manager/package_damellis_attiny_index.json”



I did a bit of research and did not find a way to resolve this issue. I tried installing an older version of the Arduino IDE which doesn't even have the same interfacing for adding additional boards. I then reinstalled the newest version a couple of times which didn't work. Some forums mentioned reinstalling Windows 10 which I was not about to do so I just headed to the Sensor Lab in hope for answers.

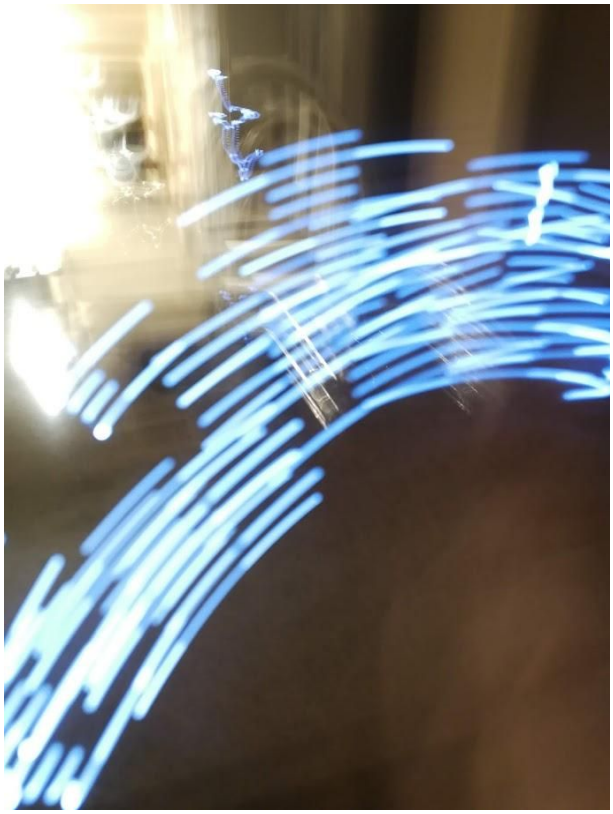


Circled to the left is the bad solder (not that any of them were really that great).

Once I got to the Sensor Lab and consulting other students, no one seemed to have seen my issue before. I was able to program my chip using someone else's laptop and Arduino IDE. Once I programmed it, I immediately tested in in my circuit and it did not work. I asked to test my chip and battery on someone else's functioning circuit just to be sure both were working, and they were. I was horrified.

When I got back to my apartment I the multimeter my dad leant me to check my connections again and found that I had a bad solder between the resistor and the wire connecting all the negative poles of the LED's. I re-soderd it and it worked!

I tried taking a low exposure image of my chip in action (which didn't really work given I didn't have a camera that could take low exposure images). I'm happy with it all the same!



Part 2

The only differences I've found between the Alternate Circuit and the Built circuit are that in the build circuit there is 1 resistor for the 5 LED's whereas the Alternate Circuit has 1 resistor per LED. The Alternate Circuit could be considered the more reliable circuit in that if one resistor fails the whole system wouldn't fail, you would simply have one LED that wouldn't work. Though with the Built Circuit if the resistor fails it would be responsible for every LED and nothing would display. Therefor the Alternate circuit is a little more reliable in that it would help in providing information about debugging your circuit. If all the LED's aren't working in the Alternate Circuit you would know that it is unlikely the fault of the resistors or LED's but another issue and if 1 of the LED's weren't working you would know that it were the fault of that LED or it's resistor.

My change to the alternate circuit isn't very creative, but it would be to use a photocell as a switch in place of a simple switch. Given that the project works best in dark lighting it could be a little more telling of the circuit to have it only function with a lack of light as that's when it would likely be used anyway!