

Product

The product uses a push button that when pressed will toggle an LED light to turn on or turn off. Then with human input, by typing in “a” it will change the brightness to 0%, “b” will change it to 20%, “c” to 40%, “d” to 60%, “e” 80%, “f” to 100%, and by typing in “?” you will receive the current brightness level. The product works by using arduino code to tell the button to act as an input and the LED as an output. When the button is toggled, it uses analog write to set its brightness value to 50% and then sets a boolean variable when the button is in the state to true. When the button is pressed again, it will set the brightness value back to 0 and set the same boolean variable to false. Then we can set a char variable since we are using characters and set it so `humaninput = Serial.read`. Now we can make an If statement that will change the LED’s brightness based on whatever user input we put it, but this will only work if the initial if statement uses the boolean value from before when the boolean value is set to true. Now whenever one of those letters is imputed, the brightness will change value and only if the button is toggled on.

Challenges

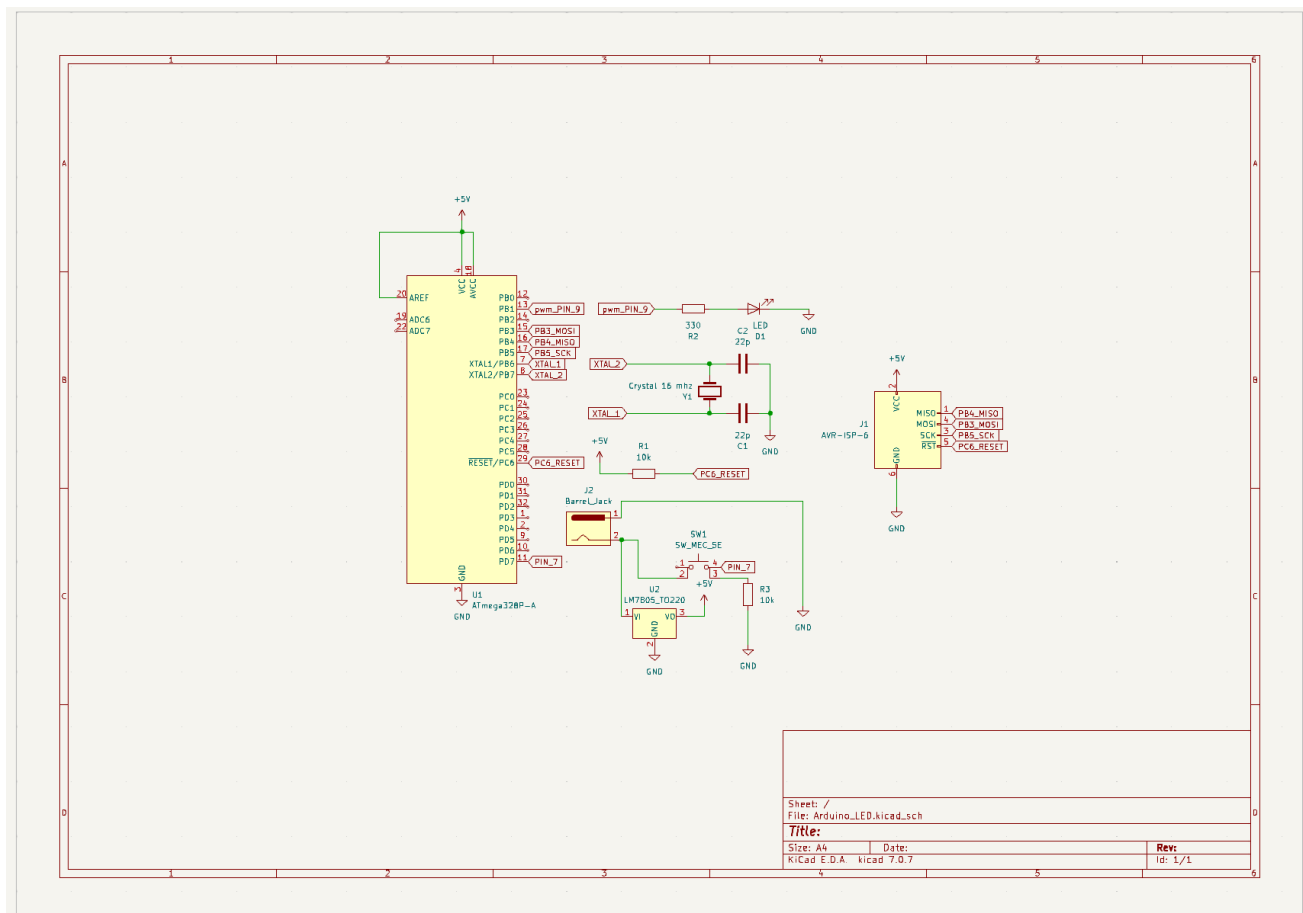
I think the biggest challenge of the entire project is finding all the information on the entire project. While there are a lot of great resources, finding specific information tends to be troublesome. Another factor is that I was going into the project with 0 experience so many things proved to be difficult. My first issue was trying to make the input functions only work when the LED was turned on. There were two problems with this. First, when the button would turn on and off, it would read digitally with values of HIGH and LOW. I am assuming this interfered with the analog values when the brightness changed because when the button had a value between 0 and 1, the button could not tell the difference. After fixing that the brightness was working, but then to actually make it turn on and off started to become difficult for me. Then another recruit Jefferson helped me out (he helped a ton with going in the right direction) by giving me the idea of using a boolean value to add the extra value of the button being true when on and false when off. After that the arduino system then worked. After that I did not face too many challenges, the main thing about the pcb schematic was learning what the global flags did and actually designing the circuit (even though tinkercad gives you one I am very bad at circuits). After the schematic I added the footprints just fine, but finding some of them was extremely annoying. Then the actual pcb tracing was not too bad until I realized that I had to connect the right components to other components which became a problem as I mixed up voltages and grounds. Also the barrel jack was really annoying because I kept getting an edge cut

error which thanks to Joseph (I am guessing you are reading this) told me how to do it. Then strangely setting up github super confused me, but I eventually figured it out.

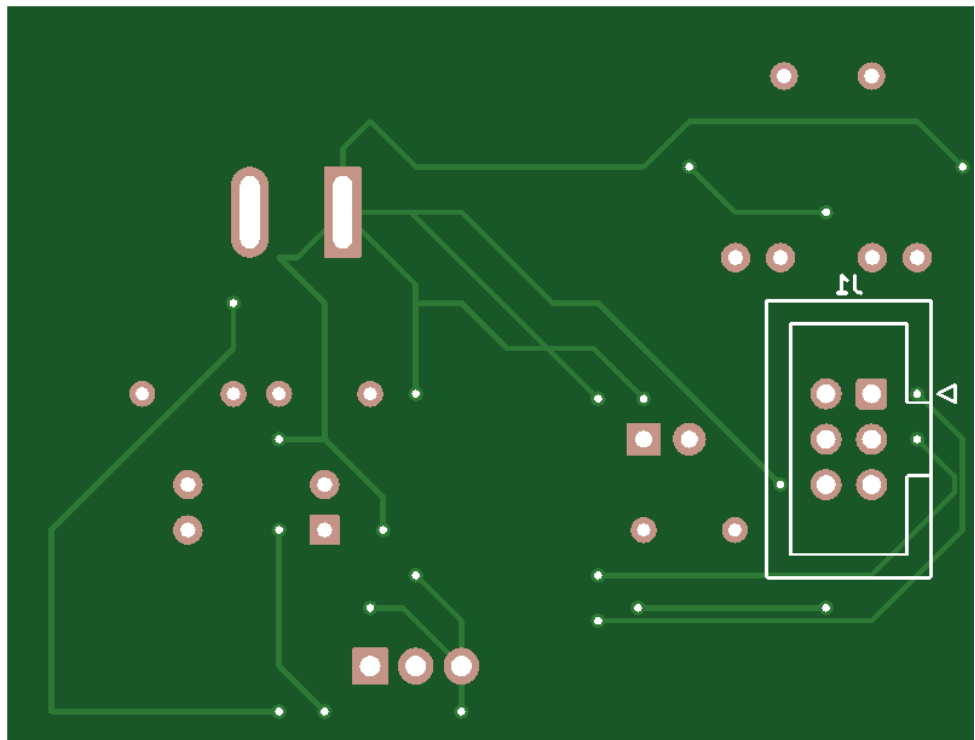
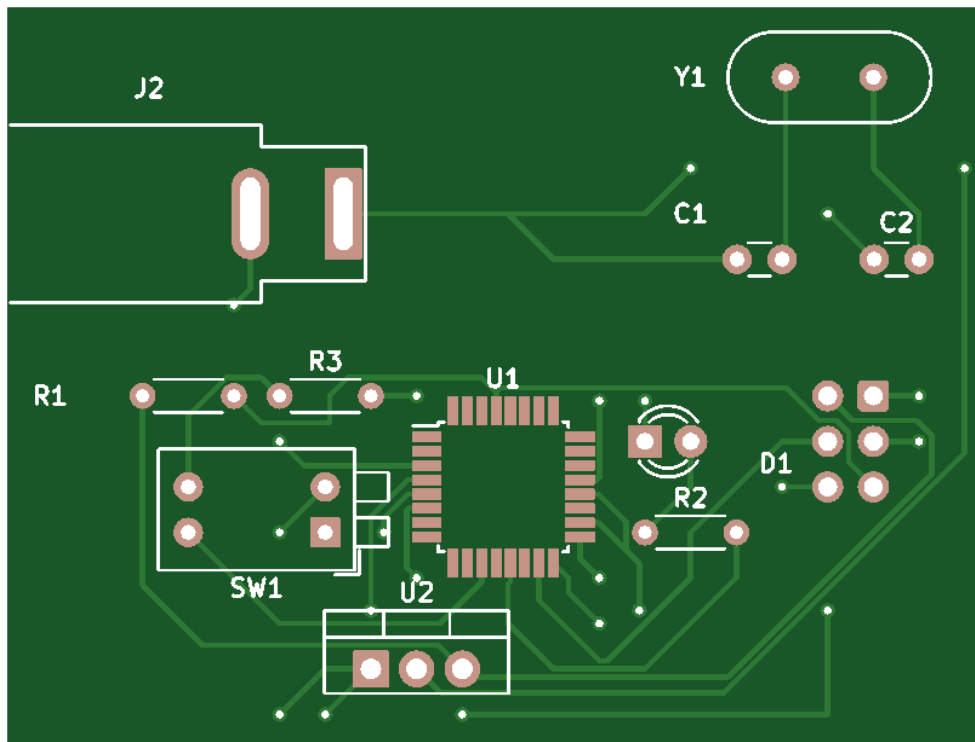
Conclusion

Overall, this project was very fun and I really enjoyed every part of it except the coding part. I dislike coding. This project was a great teaching tool on how to use all my resources to complete the project. There is still so much I have to learn and I will need much more practice with coding and pcb design. The design of pcb's is something I could see myself doing in the future and putting a lot more work into it. I am hoping I can be a good part of the team.

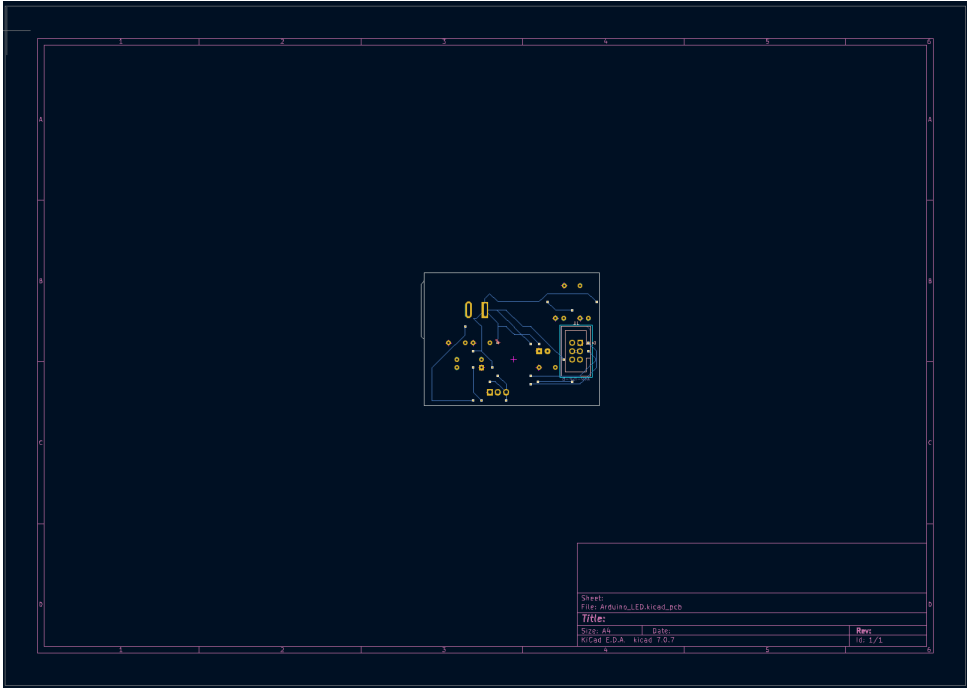
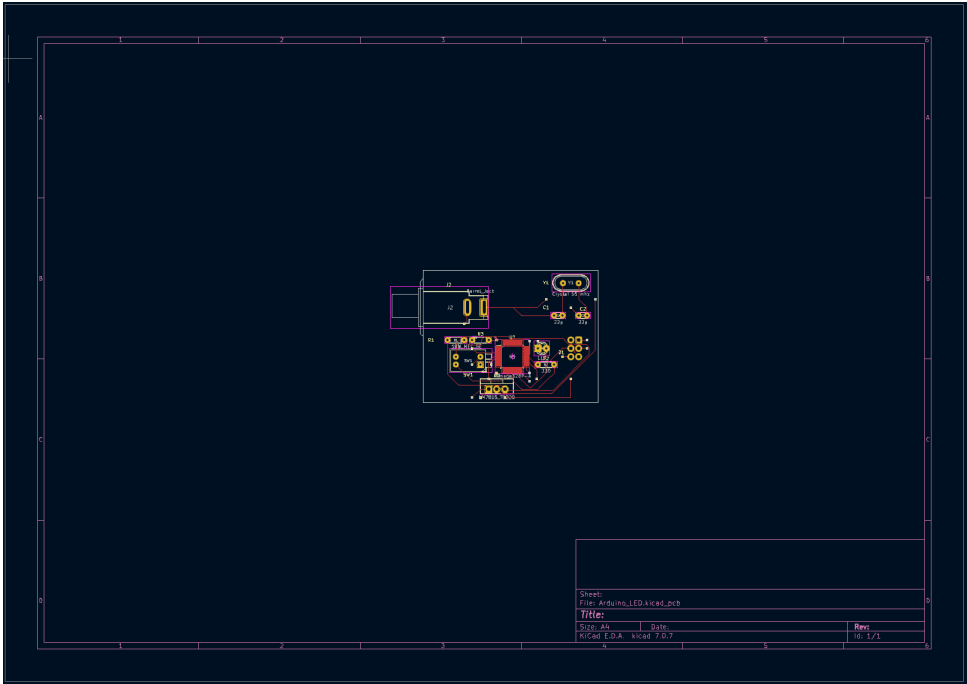
Schematic



Front



Back



BOM Table

- ClearLED
- AVR-ISP-6Header (1)
- Barrel Jack (1)
- Resistor (330 ohms) (1)
- Resistor (10k ohms) (2)
- SPST Piano Button Switch (1)
- ATmega328P-A Package Microcontroller (1)
- HC49-US 16Mhz Crystal (1)
- LM7805 5V Regulator (1)
- 20 pF Disc Ceramic Capacitor (2)