LED Prop Module

Electronics Final Report – Olivia Jo Bradley & Nabih Estefan

Introduction

For this project, we wanted to create a LED Prop Module by creating a battery powered PCB with a microcontroller, LEDs, an accelerometer, and buttons ,which would be enclosed in a clear capsule.

Both of us are members of the Olin Fire Arts Club, and enjoy springing and performing with fire. Spinning fire is dangerous, and fire props are delicate, so we try to avoid practicing and trying new tricks with them. The solution to this is an LED version of the prop that is more durable and less dangerous than fire props, but is still exciting and aesthetic, so it can be used to perform with if desired.

This project requires a board with a battery management system, bright RGBW LEDs, an accelerometer (to react to the movements of spinning), a button to change modes, and a microcontroller that can run the code to make all of these work. As a stretch goal, we also wanted to add a bluetooth module to the board so we could connect and sync the colors and patterns of multiple modules.

Selecting Parts

The process for speccing out the parts for our board was, unsurprisingly, very easy. The reason we say unsurprising is because most of the parts we needed for the board we chose through different projects we or Brad had done before. We based our project off of Mini-Project 3, including all of its components, so we focused on designing the 5 new sections we needed for the board:

1. Battery + Battery Management System
2. Wireless Module
3. Accelerometer
4. LEDs
5. Color Change Button

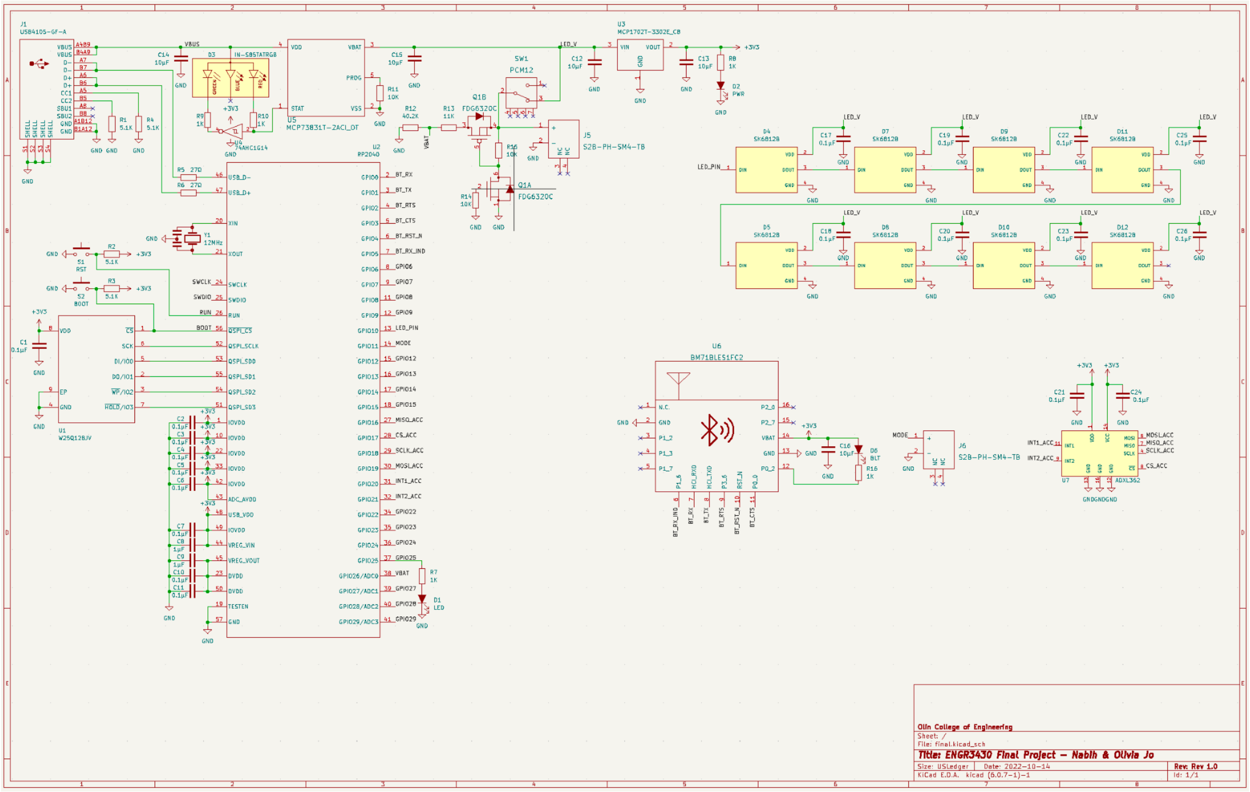
The first three were very easy because Brad showed us one of his old projects that also included these components, and were able to pull parts from that. The color change button was very easy to technically spec, so we focused on choosing a button that would fit the final aesthetic of the module, and went with an arcade style clear button. Finally, the LEDs took us the longest because we had to find LEDs that were bright enough but didn't require an insane amount of voltage and/or current to run. We decided Neopixels were the best bet and eventually we found some side-lit ones that would allow us to shine lights on two sides of our board.

Making the Schematic

Below, you can see the schematic we ended up designing for our LED Prop Module. The schematic work was surprisingly easy because of a couple of reasons, very similar to the reasons choosing parts was easy. For starters, we had a pretty big baseline with the Mini-Project 3. The other 5 big components that needed to be added to the board were:

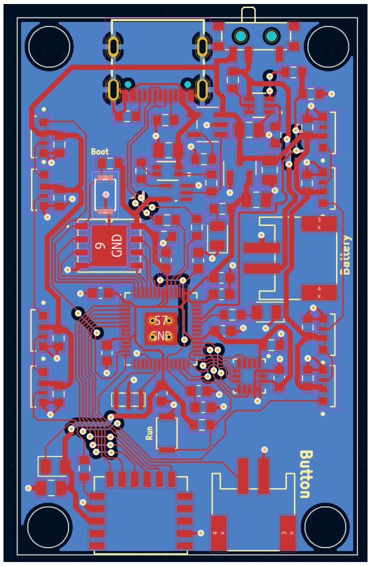
1. Battery + Battery Management System
2. Wireless Module
3. Accelerometer
4. LEDs
5. Color Change Button

The first three were very easy to add because Brad helped us understand how they worked with one of his older schematics for a separate project that used these components, so we were able to just bring those over. The LEDs were also very easy to add because we only needed a single pin from the RP2040, and then we followed the schematic from the datasheet. Finally, the color change button was probably the easiest, since it's just a button connected to the RP2040, and we have implemented surface mount versions in the past few mini projects. This meant that we were able to get a schematic done very fast, and were able to move onto the layout quickly.



Making the Layout

To the right you can see the final layout of our board. Once again, a lot of the general knowledge from Mini-Project 3 was brought onto this board, which resulted in it only really needing 2 revisions. Our first revision was getting the components down and fitting into the tiny board we gave ourselves. This board is very small because we didn't want to have to handle a giant board for a prop module, since those are supposed to be nimble and small. Below the board you can see the diameter of our battery for context. While we were trying to make a smaller board, we did limit our minimum component size to 0603 components due to our experiences with Mini-Project 3.



Building the Board

Once we got the physical boards and parts after thanksgiving, we set to building our board. We used the stencil that Brad provided, and created component sheets to organize our strips of parts. Our first iteration of the board had an unattached RP2040, and even though we tried to hand-solder it, we weren't able to get it to work. However,our second version worked perfectly, and we were able to run the LED code on it immediately!

Code

Once we had a working board, there were 3 steps to the code development for the board. The first step was to get the LEDs to light up a single color which was extremely non-trivial since there is a micropython neopixel library which makes this extremely easy. The second step was getting the button to work to change colors. Theoretically this should have been non-trivial, and in code it was, but we had bought the wrong JST PH 2.0 connector cables for the button so we stalled here for a while. Finally, the third part was to implement the code to read data off of the accelerometer. This was extremely hard because the accelerometer works on a SPI bus, something neither of us had ever done. Eventually, we figured out the code we needed for it, but turns out that trying to compile the code needed through Thonny (the IDE we had been using to flash the RP2040 after it was the only working option in MP3) is basically impossible because there is no support for external libraries. Eventually we found an ADXL362 library that was perfect for what we needed, and it worked with Micropython! Sadly, we couldn't install all the needed dependencies on the RP2040, so we had to turn back and look into the built in SPI library and try to get that one to work. We eventually got code that theoretically should work, but couldn't figure out why it wasn't and we ran out of time. Currently, the code in the github repo has the implementation for mode changes with the button and has theoretically working code for the accelerometer changing the blinking speed, we believe that the error with the accelerometer comes from the physical part on the board but we didn't have time to assemble a new one. This means our Software folder currently has 2 versions: main.py and main_accel.py. The first one is the one we're currently using that has no accelerometer code, and the second one contains the accelerometer code.

Building the Enclosure

The enclosure for our LED Poi module needed to be semi transparent and contain all the necessary components, and hopefully easy to interact with (charge, press the button to change modes), connect to a rope handle of some sort, have a comparable weight to a fire prop, and be robust. To create the module, we instantly turned to 3d printing as it would allow us to use transparent materials, create unique shapes, and iterate quickly. We chose to make the enclosure out of clear resin, as it is a mostly transparent material, diffuses light nicely, and prints solid, providing more weight. We started with a traditional cylinder, making sure the button, board, and battery all fit. Once this was confirmed, we iterated and created a form with a more natural, rounded shape, as well as some modifications to make it print easier. The final enclosure can be seen below.

Final State

This all means that the final state of our project is a self-encased board with battery and button that has different blinking modes for spinning as a prop. Sadly, it has no accelerometer reaction, but assembling a new board would most likely solve this problem.

Project Experience

We both had a lot of fun with this project, as we were both personally motivated and excited about the prop module we were making. We both came into this project wanting to become better at board assembly, as Mini Project 3 took far too many attempts. We were able to mostly complete this, as we were much better at connecting the RP2040 and only had to use 2 boards to get our first working board. We also got much better at selecting components, as we had to select and design for our LEDs using datasheets, something Olivia Jo was interested in doing, and she was able to complete this learning goal. While we were not able to complete all of our goals and stretch goals for the module, we were able to complete our learning goals, expand our knowledge, and create a working prototype.

The link to our github repository that contains all of the elements we created in this project is: https://github.com/nabihestefan/LEDPoi

This includes a schematic of our system, CAD renderings, our code, a bill of materials, photographs of our system, and our photographs and videos of our prototype.