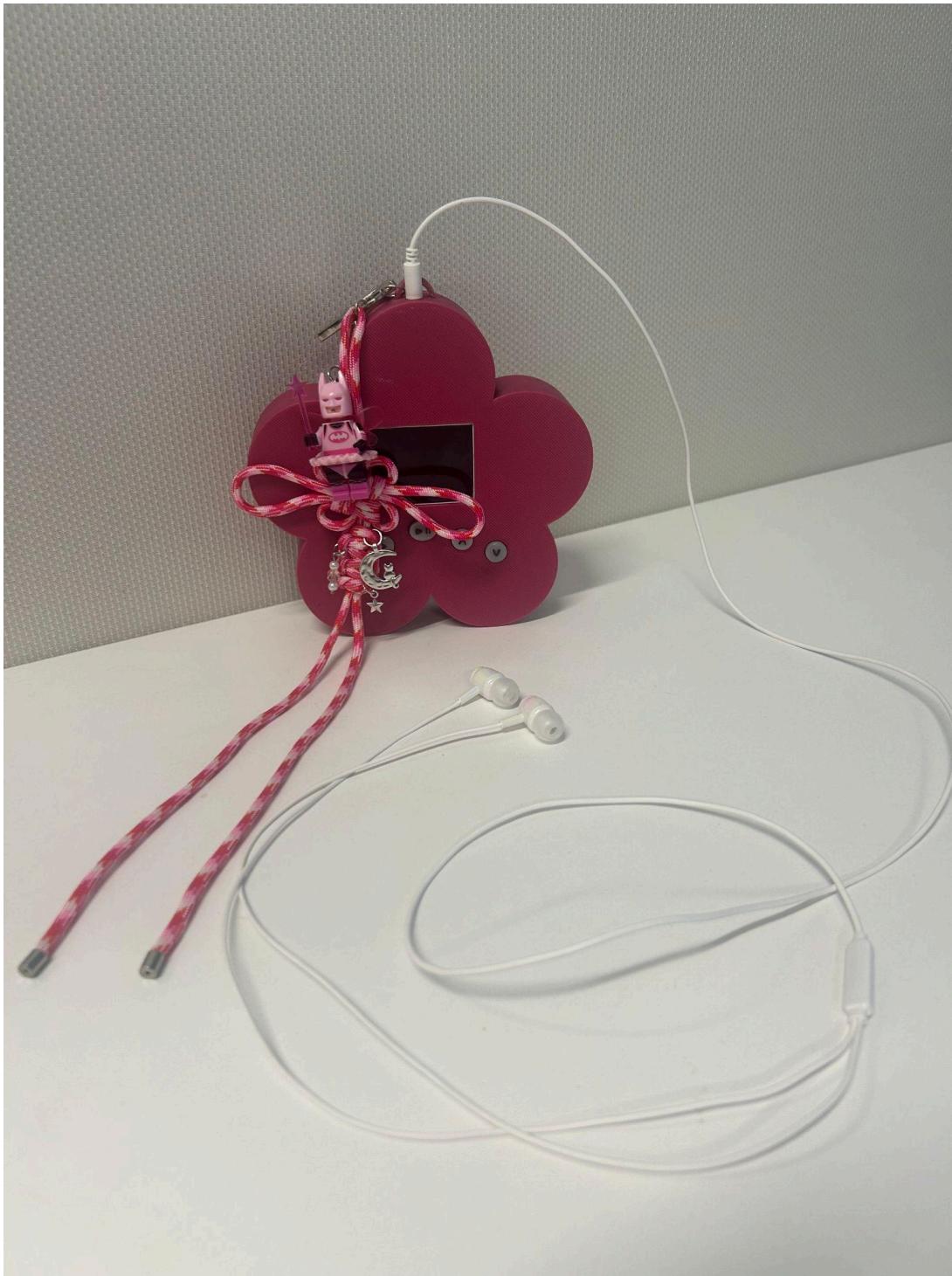


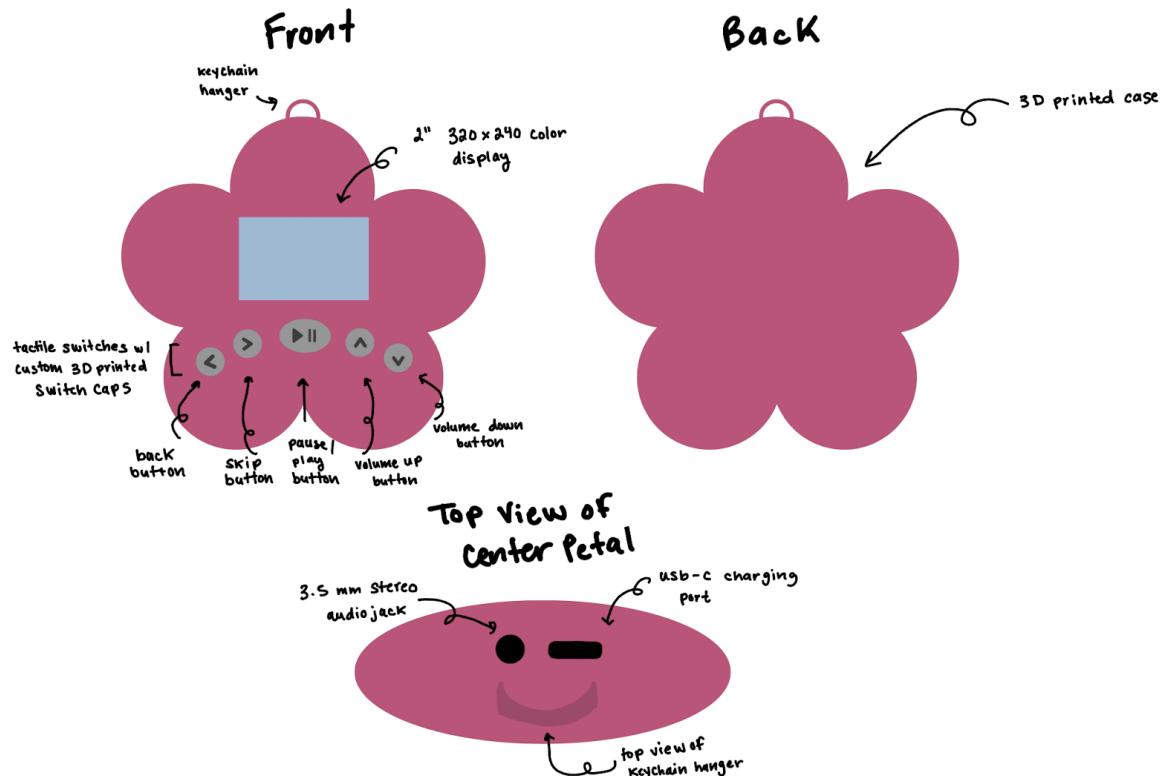
Personalized MP3 Player

Emily Mizuki



Overview

The MP3 player has all the basic functionalities of a typical MP3 player. It features a 3.5mm audio jack stereo output that connects with wired earphones and a 320x240 TFT display. There are 5 buttons: a pause/play button, a skip button, a back button, a volume up button, and a volume down button. It has a pink flower shape with a keychain hanger. The image below shows the initial mockup of the MP3 player design.



* Rough Sketch to provide an idea, final design will likely change based on component fit and size

Figure 1. Initial case design

Iterative Process

The major hardware components used would be an RP2040 Feather, DFRobot MP3 Player Pro, 320x240 Color TFT display. There are also 5 buttons, a 3.5mm audio jack, a 3.7V battery, and a slide switch. All components are connected to the RP2040 Feather that controls the display and the audio module.

Originally, I intended to store the music on an SD card that the TFT display can use to store complex images/backgrounds that can also be used as additional memory by the RP2040 and only use the DFRobot module as the audio driver. The issue that came about was that the DFRobot module can only play MP3 files stored in the on-chip storage and can't read external files. This meant that the number of songs I could save decreased dramatically (from 128 GB of storage to only 128 MB) and that I needed to add another USB-C access point on the case.

Beyond this issue, everything else hardware-wise went fairly smoothly. The only other design change that was made from the initial mock-up was that an on/off switch was added to the back of the case. I considered getting a new audio module, but that would've required an entire design overhaul, both hardware and software-wise, so I decided to continue with the DFRobot MP3 Player Pro since it was sufficient for this class project. I can easily upload music, and it comes with a library, DFRobot_DF1201S, that already includes core functions like play, pause, skip, back, etc. If I were to switch to a DAC, I would need to write all of those core functions and the MP3 file management system myself. This is something I'd like to continue and look into as a personal project, but I didn't have enough time to do it for this project, unfortunately. The following images are a picture of the circuit schematic and breadboard.

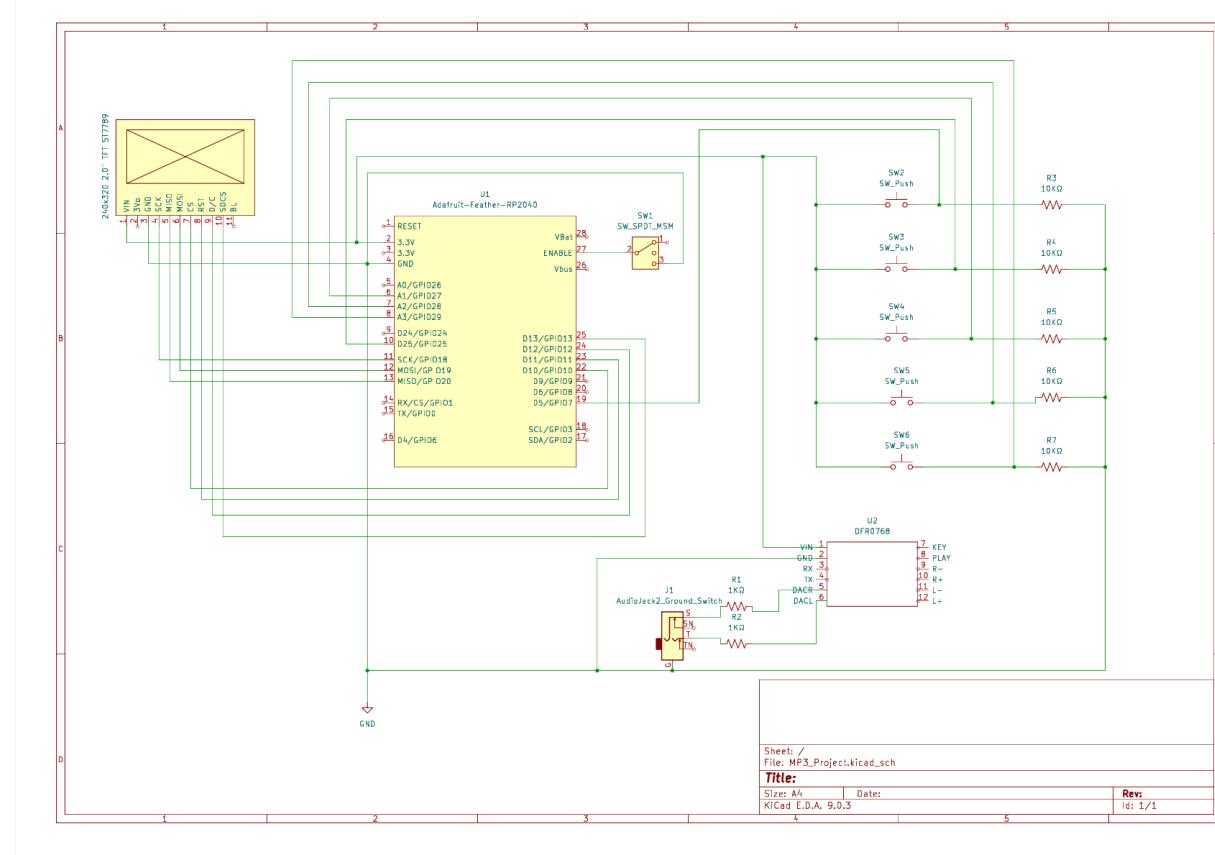


Figure 2. Hardware schematic for the MP3 player

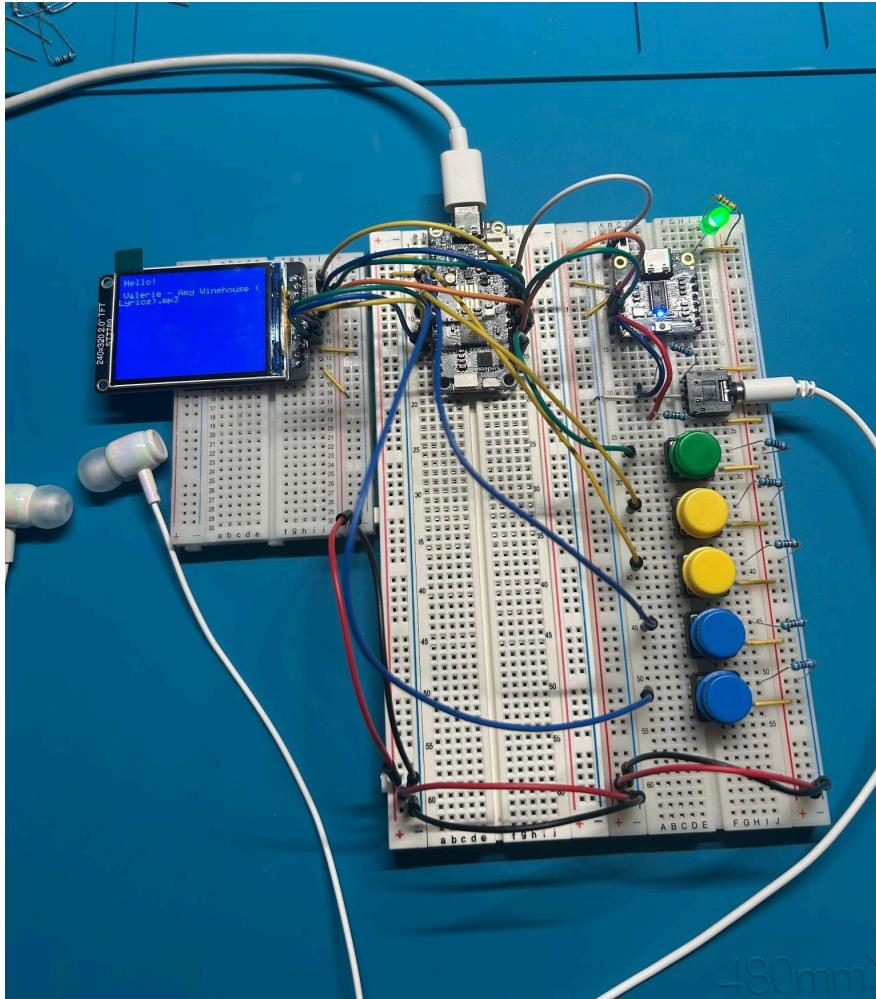


Figure 3. Breadboard layout of MP3 player. Music streaming is fully functional, UI is still being implemented

The tutorial that was most helpful to me would be this [Walkmp3rson: MP3 Player](#) tutorial I found. I ended up only using it as inspiration since I varied the implementation quite a bit from the original tutorial. I used it as a jumping-off point in terms of what hardware to select, so that's where I got the TFT display and the RP2040 Feather. I liked the idea of the on/off switch from this tutorial, so I also implemented it in my design. From there is where the two products diverge further as the Walkmp3rson is implemented with CircuitPython and I implemented my MP3 player with C++. Another resource I used a lot was the [DFRobot MP3 Player Pro Product Wiki](#), since there weren't many tutorials where that module was used, since it is a newer product compared to the older and more popular DFRobot Mini.

One of the biggest takeaways I learned was at this step of the project. It was to check the capabilities of every component more carefully. I'm ultimately happy with how the MP3 player came out, but I wish I had known beforehand that the DFRobot could only access internal files.

This was an important lesson to learn, but it also allowed me to practice working with what I had and how to overcome potential limitations.

Once I was able to get the hardware working, I soldered it all together onto various protoboards to be placed in the case. It took me 4+ hours to decide how to lay everything out and how it would work in the case and then solder accordingly, but I got it done. Unfortunately, I used 22 AWG solid core wire, and it ended up not being very flexible. This didn't work well when I tried designing the case, so I had to resolder a majority of it with a 20 AWG flexible wire instead. Luckily, the most time-consuming part of the initial soldering session was figuring out the layout, so this resoldering process only took an hour at most.

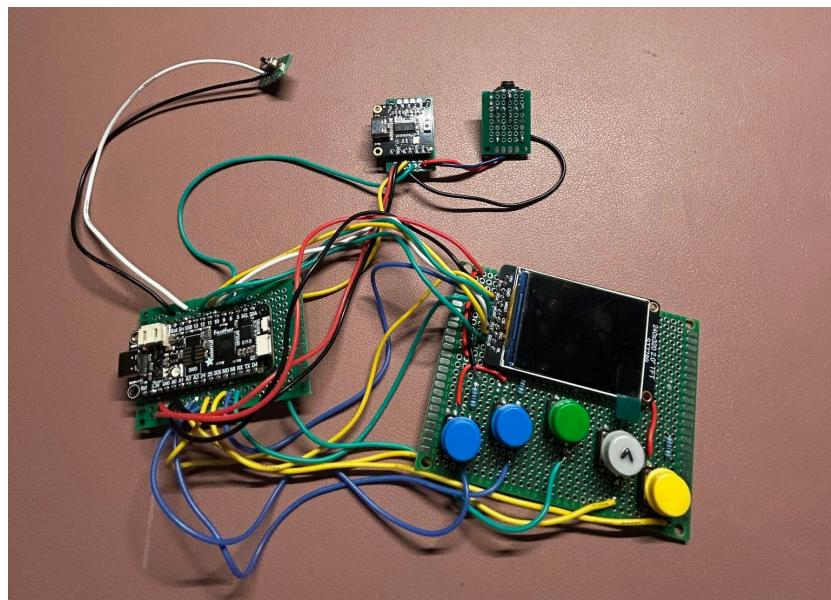


Figure 4. Hardware soldered onto protoboards

The next phase of the project was designing the buttons/ case for the MP3 player. I was going to 3D print these components, but had never done anything with 3D modeling, so this was a challenge initially. I decided to use Tinkercad to create the models, and once I got over the initial learning curve, it was quite fun designing the buttons and case. It took a lot of print attempts to get the case right, especially, but it turned out almost exactly how I envisioned it. I originally wanted a smaller and sleeker design, but was limited by the component sizes being larger than anticipated once soldered onto the protoboards. Since I didn't have access to gray filament at the time, I sanded and painted them how I wanted and used resin to seal the design.



Figure 5. 3D printed pause/play, volume up, volume down, skip, and back buttons

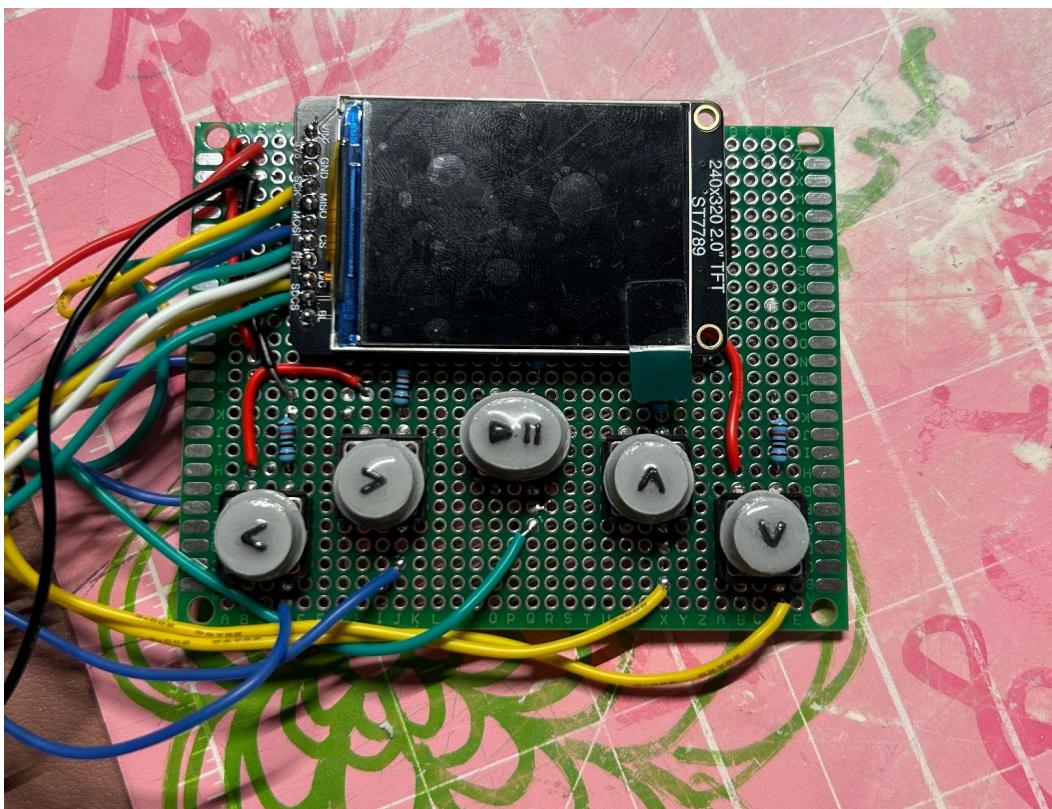


Figure 6. Painted and sealed buttons placed on protoboard.

The case design was one of the more time-intensive tasks, but definitely one of the most rewarding, too. It took 15 different print tests to get everything to a point where it could be fully printed. Once I got the design to where it fit the buttons/screen well and fit the ports, I was able to put the hardware into the case. I was able to use a lot of the skills from the skill demos we did,

but I did have to learn a lot more about audio processing and 3D printing than was covered in our modules.

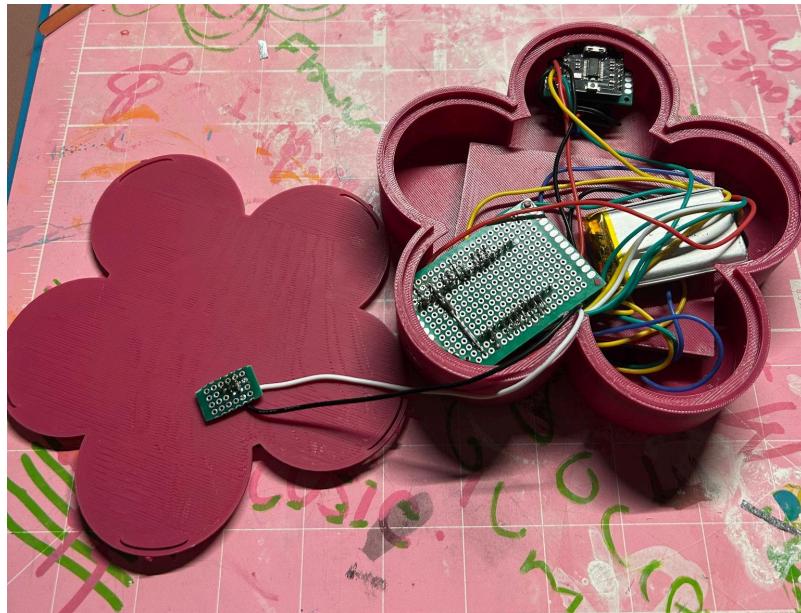


Figure 7. Protoboards being placed into a 3D printed MP3 player case

When designing the MP3 player, I was inspired by 2000s technology aesthetics, specifically Tamagotchi's. I wanted to keep the fun keychain vibes that it had, so I made a paracord Good Luck Knot keychain with a pink fairy Batman keychain for added personality.



Figure 8. Keychain for MP3 player

In terms of programming, the DFRobot_DF1201S and SoftwareSerial libraries were useful for the core functionalities of the music player. I also used the SPI, Adafruit_GFX, and Adafruit_ST7789 libraries for the display. The most difficult part, coding-wise, was trying to get the elements I wanted on the screen properly. Otherwise, it wasn't too bad, just a lot of trial and error, as shown below.



Figure 9. Example of display in the middle of the coding process. The blocks were to help visualize the bounds of where the song name and artist name were going to go

Lastly, as a fun easter-egg, I added an alternative display that can be used by simply entering the Konami code on the MP3 player: volume up, volume up, volume down, volume down, back, skip, back skip. It is shown below in Figure 10. It is also the initial loading screen when the MP3 player is first starting up. You're able to toggle back by entering the Konami code again. All basic functionalities still work when in this alternative display mode. Figure 11 shows the finalized default music playing view.

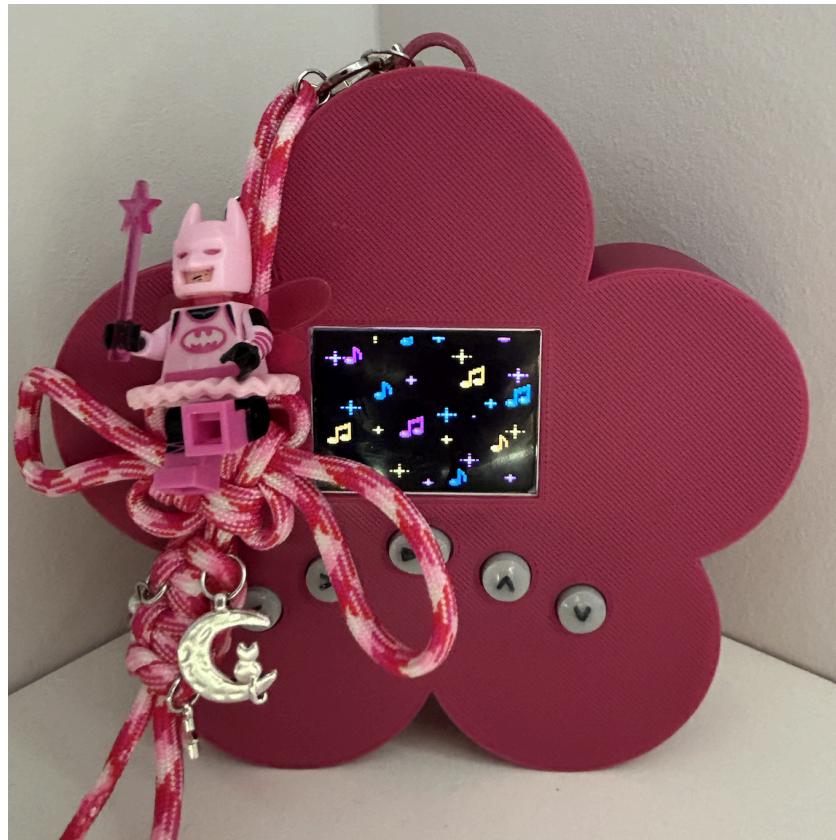


Figure 10. An alternative display screen that is shown when the MP3 player is first starting up or displayed long-term when the Konami code is entered



Figure 11. Default music playing view for the MP3 player

Demo Presentation: [□ CS 3651 MP3 Player Project](#)

Demo Video: <https://youtu.be/JtUGNILLJnY?si=bc0iQjZwPzlIrHHT>