ECE09402 Project 2D - PCB Assembly & Test

Aidan Sharpe

Karl Dyer

Michelle Frolio

May 1st, 2025

Introduction

The goal of this portion of the project is to assemble, test, and debug the digital clock. This stage of the project included assembling the components on the board, testing that the board fit in the enclosure, and fixing any issues with the functionality of the clock. Finally, the clock's oscillator was calibrated to have a period as close as possible to one second.

Requirements

- 1. Clock operates with two AA batteries
- 2. Colons visually separate the time into HH:MM:SS format
- 3. The clock must count up using an internal oscillator
 - 1. The oscillator must have a period within 10% error of one second
 - 2. Clock source must be selectable between internal and external oscillator
 - 3. The internal oscillator must have a test point for debugging
- 4. The clock must be enclosed
 - 1. The enclosure must have at least two parts
 - 2. The enclosure parts must join mechanically
 - 3. The enclosure shall achieve an IP30 rating
- 5. There shall be two user-accessible buttons
 - 1. One button shall increment the minute count when pressed
 - 2. The other button shall increment the hour count when pressed
 - 3. Both buttons shall be debounced

Data

When the boards arrived, the first step was to solder the surface mount components on. This involved setting up the automatic pick and place machine for pasting and part placement. Unfortunately, when by board was in the pick and place, the paste job failed since the solder paste had become runny. Eventually, we were able to get the paste job working properly running smoothly. With the surface mount parts on the board, we then headed to the reflow oven. After a quick visual inspection, it was time to solder the through hole parts. This soldering step, while tedious, did not have any issues. Once complete, the board was ready for testing.



Figure 1: Fully assembled PCB

Once assembled, we began testing. This involved hooking up a DC bench power supply to the battery terminals. The output was set to 3.2V with a current limit of 300mA. The clock began counting, but the seconds did not cause the minutes to increment. Using the increment minutes button, we found the minutes could count up but could not increment the hours.

After much troubleshooting, we found that our debouncing circuits were acting as low pass filters, so the short duration clock pulse was attenuated and unable to increment the counter. We resolved this by adding a diode between the debouncing circuit and the clock input to isolate the signal. I'm not sure how the orignal designer got around this as his RC debouncing circuits are directly connected to the clock input without a diode (Neutrino-1, n.d.). I wouldn't be surprised if he did not run a full functionality test of his project, because this is far from the only mistake in his provided schematic drawings.

Once we confirmed full functionality, we calibrated the oscillator to get the period as close as possible to one second. Unfortunately, the longest period we could achieve before hitting the limit of the potentiometer was about 900ms as seen in figure 2.



Figure 2: Oscilliscope view of on-board oscillator

Reflection

General Design Considerations

One thing I wish the design included was a reset button. I often found myself trying to reset the time to zero, especially when recording the clock for submission. This button would also allow for some degree of control over the seconds count. Since the counters all have a reset pin, I'm sure adding this feature would be quite simple.

Another change I would make would be to the oscillator circuit. Not that a 555 timer will ever be an awesome oscillator, but our choice of potentiometer, in many cases including with my clock, made it impossible to get the clock period to one second. In my case, the best I could do was about 900ms as seen in figure 2. If I were to do this project over (or just pick different parts), I would do some uncertainty analysis to ensure that a period of 1s was well within the range of possible periods given the tolerances of my parts.

Mechanical Considerations

Mechanically speaking, there were a couple issues I ran into. As mentioned previously, rushed bed-levelling left the surface finish of of my case in rough shape. I later discovered during assembly for testing that since the board was not supported opposite the battery, my PCB and especially my enclosure would flex whenever I tried to put the batteries in. In a future revision, I would definitely add some kind of support structure behind the board there to bear the load.

Board Layout Considerations

On the topic of dealing with batteries, I also found that my battery connectors were quite close together. This lead to several issues. Since the 3D models in the part library did not show an issue, with the parts intersecting, I figured the placement of the two battery connectors was fine. Only after ordering the board did I notice that, while the battery holder models did not intersect in the 3D view, the tabs would definitely collide with each other when the batteries were inserted. This actually caused several issues, including a short circuit between adjacent connectors that had to be insulated with Kapton tape (Wikipedia contributors 2025). Unfortunately, the connectors had not been de-burred, so the sharp edges made quick work of the Kapton tape, which had to be reinforced with electrical tape.



Figure 3: Battery connector Altium 3D viewer

Additionally, since the battery connectors interfered with each other, it became nearly impossible to pop the batteries out by hand when the board was in the enclosure. Either the case had to be disassembled, or a small pry tool had to be inserted to remove the batteries. This is obviously not ideal and would be another thing I would change in the next revision.

To prevent similar issues in the future, I would recommend modifying the 3D model to show the shape of the battery connectors flexed outward instead of passing through the battery as it is currently.

Debugging Considerations

I think I managed to create more issues on my board while debugging than I had before I started testing. Almost all of these issues were preventable if I had just taken the time to slow down, think, and do it the right way. Moving quickly is a sure way to break things.

For example, I had thought that I was going to have to completely de-solder and replace one of my battery connectors, but it turns out that I probably did not have to. In the process of removing it, I managed to accidentally knock off a capacitor and an inductor. While putting the parts back on, I scratched through the solder mask with my tweezers by accident, and also heated the inductor to the point of mechanical failure. Luckily, these issues were fixable. I could have avoided additional damage by taking my time soldering and being very deliberate with where I applied heat.

Another example of improper debugging occurred while trying to understand why my minutes and hours were not being incremented by the oscillator. I initially thought that some of my diodes were dropping the voltage below the threshold required to trigger the clock input, but later found out that my button debounce circuit was actually the culprit. Unfortunately, I had already de-soldered what I had thought to be the problematic diodes, which meant I had to re-solder them. In the future, I want to be more thorough with my debugging, and ensure that I am actually fixing the root cause of an issue rather than a symptom.

Appendix



References

- Neutrino-1. n.d. "Digital Clock but Without a Microcontroller [Hardcore Electronics]." https://www.instruct ables.com/Digital-Clock-But-Without-a-Microcontroller-Hardco/.
- Wikipedia contributors. 2025. "Kapton Wikipedia, the Free Encyclopedia." https://en.wikipedia.org/w/i ndex.php?title=Kapton&oldid=1281764953.