Board 4 Report

Zane McMorris*
Practical PCB Design and Manufacture
University of Colorado Boulder

May 2024

Contents

*ID: 109839099

1 Project Overview

For the final board of the class, we added an instrument to the Arduino design, as well as a buzzer and a few smart LEDs that we could play with. The instrument uses an ADC, DAC, opamp, and some other parts we explored in a previous lab, so I will not go into much depth about it. The instrument takes an input power supply and tests it under load to calculate its Thevenin resistance. The usual test points of R+/- for the sense resistor, D+/-, TX/RX are present on this board like the previous Golden Arduino one. I shrunk the footprint slightly compared to the Golden Arduino and moved some sections around because I was unhappy routing it the first time. The general layout of the board is similar to the Golden Arduino with some minor layout changes. One of the biggest changes was the location of the reset button being placed closer to the CH340G to reduce the size of the DTR net. I also figured out a topology for the clock modules that allowed me to place them relatively freely with the passive components surrounding the oscillator. I added a 2-terminal screw input for the system power in addition to the barrel jack we've been using because I often want to use a random power supply to power my board, and I don't have any adapters for the barrel connector. I added both a switch and LED to the buzzer circuit to disable the buzzer, but still have an LED indicating when the test is being performed. I programmed the smart LEDs to light up red, green, and blue with the intensity related to the current going through the test circuit. It gets brighter and brighter as the test continues. Overall, I am happy with how the board came out and worked.

This was the first and last four-layer board that we designed in class. I chose the SIG-GND-GND-SIG stack-up because it provides two nearly uninterrupted return planes for the signals on the top and bottom layers to latch onto. A property of multi-layer boards that we learned about in high-speed was that vias that connect to a layer midway through its path leave a stub on the rest of the via, causing increased near-field emissions. By using this stackup we eliminate that risk because signals always go from the top to the bottom, leaving no extra via to spew out unwanted emissions. A practice that we employed during this design was the use of stitching vias to provide a low-impedance return path for our signal vias. Each and every signal via has an accompanying stitching via to reduce noise. Because this board had a pretty low density compared to what was possible, the additional vias came at practically no expense.

1.1 What it means to work

1. 5V is delivered to the board through the standard barrel-jack connector and the Mini USB connector
2. The 5V source between the barrel jack and the USB connector can be chosen. This 5V source is then passed through the current resistor and indicates the 5V power LED
3. 3.3V is generated from the LDO and illuminates an LED
4. Both the 16MHz and 12MHz clocks generate steady clock signals when power is applied
5. Assuming no code is uploaded to the board, all other lights should not be illuminated
6. Plugging a USB into the board and my laptop causes Windows to detect a COM port. This tells us that CH340G is functional.
7. Uploading code without touching the board works. Reset needing to be pressed indicates bad component selection for the reset circuit.
8. TX and RX LEDS illuminate during the upload process and periodically if there are Serial.printX statements.

9. GPIOs can be driven from the ATMEGA using code uploaded to the board
10. Plugging a power supply into the instrument input illuminates an LED
11. While a power supply is being tested, the smart LEDs increase in brightness as the test current increases.
12. While the buzzer switch is enabled, the buzzer sounds at increasing frequency correlated to the current through the test circuit
13. The current through the circuit varies from 0-200mA with nearly equally sized steps
14. Viewing the serial monitor, the R_{th} is always greater than zero while a test is ongoing. A negative R_{th} indicates that something is broken in the test instrument

My board passed all of these tests with little effort, and all subsystems of the board were functional. No hard errors other than the width of the header pins were found.

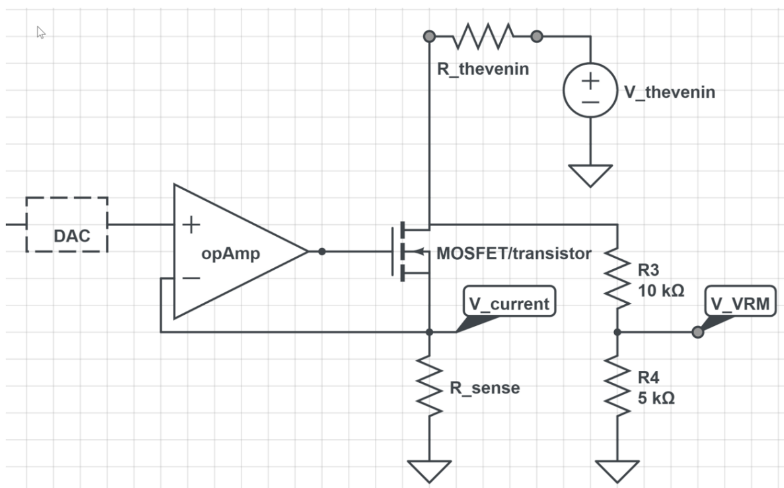

Figure 1: Instrument circuit

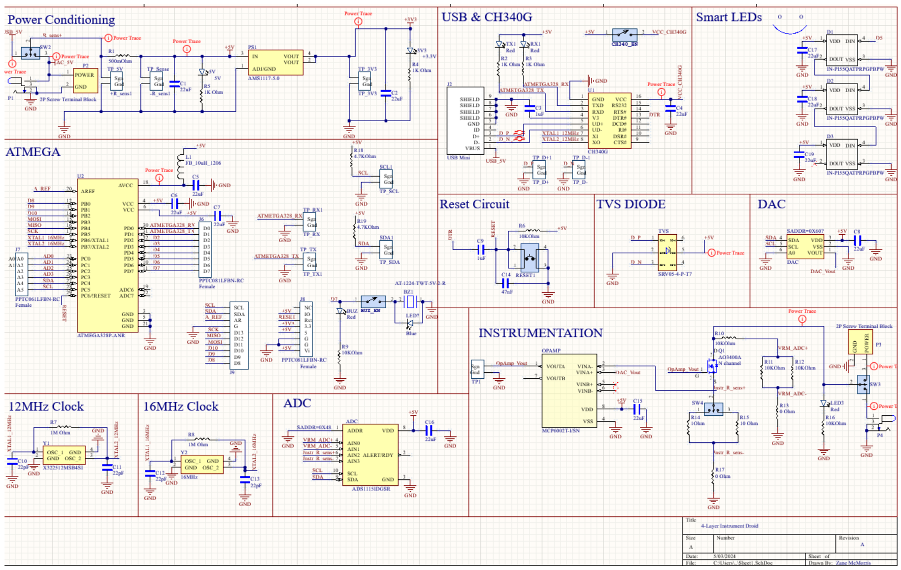

Figure 2: Board Schematic Designed in Altium

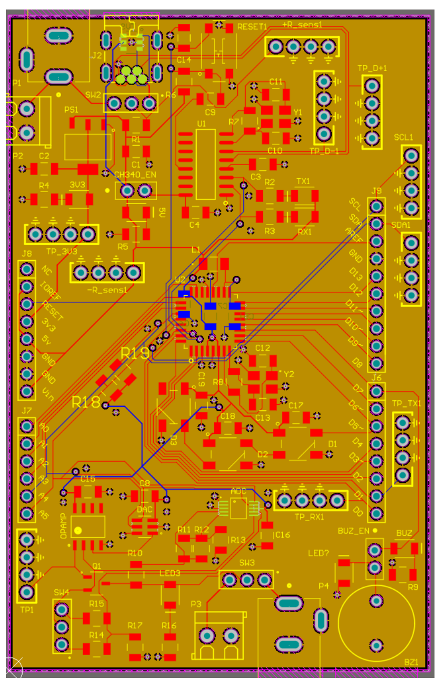

Figure 3: Board Layout Designed in Altium

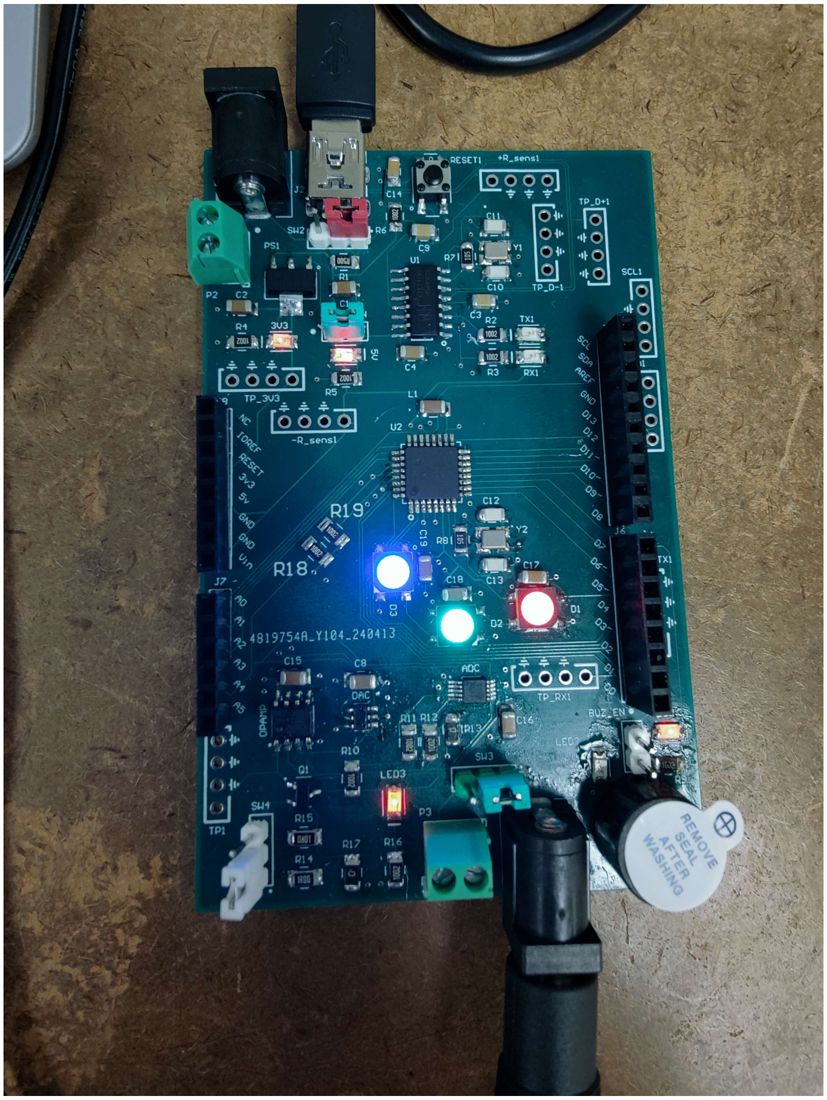

Figure 4: Final PCBA Captured Mid-Measurement

2 Board Measurements

2.1 USB Data Lines

One important metric that I forgot to measure on my last board was the quality of the D+/D- signals as the board was being programmed. I paid special attention to the routing of this differential pair and I was curious to see if my effort was for naught. To test this connection, I programmed the board while probing the test points and used the math function on the oscilloscope to subtract the D- from the D+ to give me the value of the differential signal.

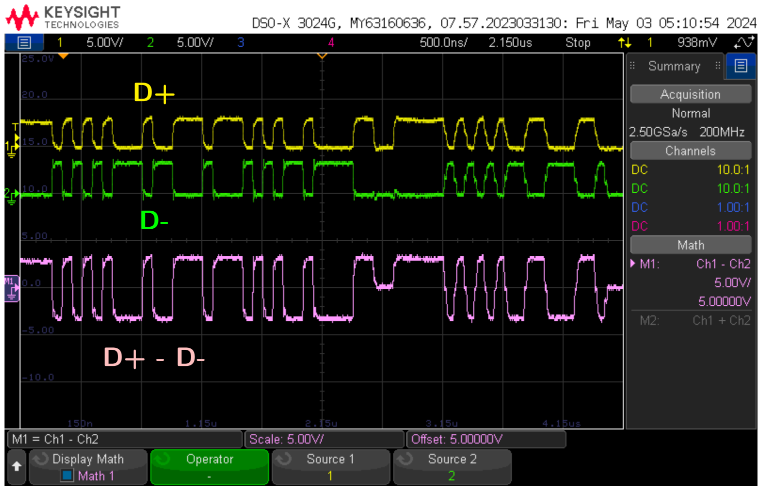

Figure 5: USB D+/D- Lines

2.2 Power Usage

To analyze the power draw of the board, I used the sense resistor populated on this board and the Golden Arduino. I expect the power draw to be pretty small most of the time, with some small increases when the ADC and DAC are actively reading or outputting voltage. The buzzer is another load that I would expect to draw extra power, and because I added a switch to enable or disable the buzzer, I can test various scenarios. I used the same probe setup as the USB test because I needed to test two points and then find the difference in voltage between them. I used a 0.5Ω resistor so the current through this resistor is $2\Delta V$, and the power consumed by the system is $2\Delta V^2$, where ΔV is the difference in voltage between the R+ and R- nodes.

Status	Average Voltage	Power
Idle	33.5mV	2.24mW
Measuring	35.5mV	2.52mW
Measuring w/ buzzer	51mV	5.2mW

The datasheet for the ATMEGA 328 states that the average power consumption at idle is about 2mW, which lines up with my measurements. The extra power draw is likely due to power dissipated in current-limiting resistors on the many status LEDs that I have on board. I wasn't expecting the buzzer to be such a significant power draw, but I think it makes sense that this mechanical device draws a lot of power while operating. During the power measurement with the buzzer on, I could see the buzzer's direct effect on the system's power draw through a regular, periodic draw, as shown below.

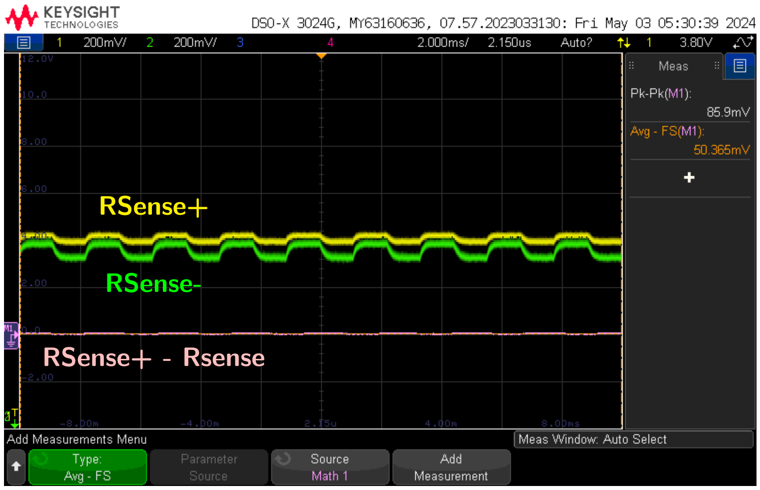

Figure 6: Voltage Across RSense

2.3 Instrumentation Measurements

2.3.1 Opamp Voltage Following

An essential aspect of the instrumentation is the DAC setting an escalating voltage to allow more and more current to be drawn from the DUT. I added a test point to see the opamp voltage because it determines the behavior of the rest of the testing circuit. The anticipated behavior is short pulses with lots of time being off to allow the controlling MOSFET to cool down and not to heat up all the instrumentation parts.

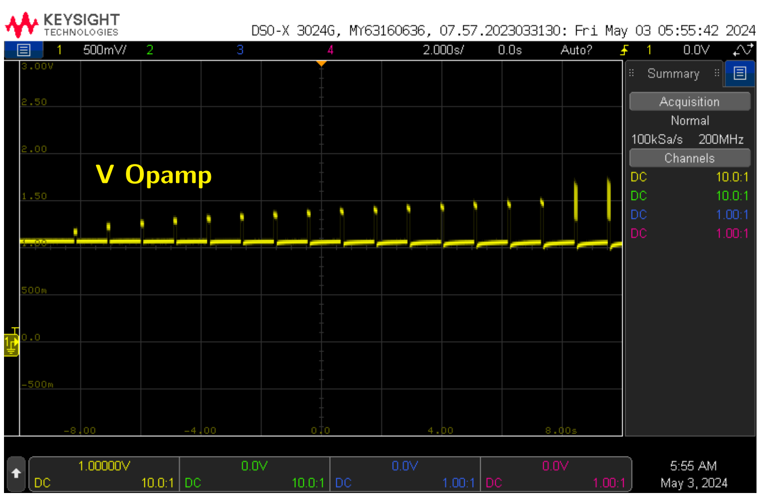

Figure 7: Voltage Across RSense

2.3.2 Results

To test the capabilities of the instrument, I tested a small batch of different power supplies. I used a 9V wall wort, a 5V wall wort, and a function generator of theoretically known impedance.

1	7.539	5.1687	5.1655	0.4214
2	18.853	5.1687	5.1609	0.4117
3	30.031	5.1687	5.1566	0.4024
4	41.385	5.1687	5.1522	0.3993
5	53.78	5.1687	5.1472	0.4
6	65.093	5.1687	5.1428	0.3991
7	76.855	5.1687	5.1382	0.3977
8	88.875	5.1687	5.1334	0.3981
9	101.915	5.1688	5.128	0.4003
10	113.863	5.1687	5.1231	0.401
11	125.746	5.1688	5.1184	0.4004
12	137.708	5.1688	5.1136	0.4009
13	150.745	5.1688	5.1082	0.4025
14	162.33	5.1688	5.1034	0.4033
15	173.91	5.1688	5.0988	0.4028
16	189.243	5.1689	5.0913	0.41
17	202.787	5.1689	5.087	0.4037
18	215.104	5.1689	5.0823	0.4029
19	227.576	5.1689	5.0766	0.4057
20	240.337	5.1689	5.0685	0.4178

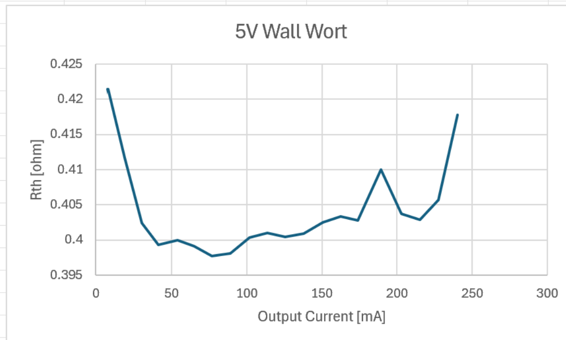

Figure 8: 5V Wall Wort

1	7.616	5.5339	5.5305	0.4465
2	18.83	5.5352	5.5342	0.0486
3	30.026	5.5354	5.5104	0.8324
4	41.281	5.534	5.4954	0.9353
5	53.661	5.5341	5.4848	0.9187
6	64.99	5.5349	5.4833	0.793
7	76.768	5.5342	5.4719	0.8122
8	88.736	5.5353	5.4687	0.7498
9	101.889	5.5356	5.4577	0.765
10	113.749	5.5344	5.4518	0.7261
11	125.708	5.536	5.4467	0.7103
12	137.54	5.5345	5.4396	0.6899
13	150.505	5.5346	5.4304	0.6922
14	162.261	5.5346	5.4115	0.7591
15	173.678	5.5346	5.4019	0.7643
16	185.295	5.5347	5.4058	0.6958
17	198.098	5.5349	5.3861	0.7509
18	209.685	5.5358	5.3899	0.6962
19	221.404	5.535	5.3747	0.7243
20	233.306	5.5352	5.3735	0.6932

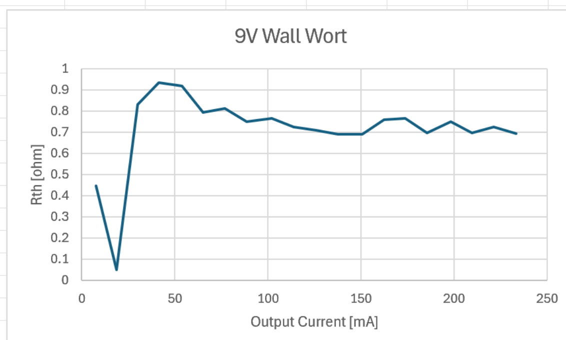

Figure 9: 5V Wall Wort 2

1	7.592	9.9003	9.544	46.9301
2	18.843	9.9003	9.0146	47.0012
3	30.08	9.9003	8.4875	46.9671
4	41.343	9.9003	7.956	47.027
5	53.77	9.9003	7.3728	47.0048
6	65.1	9.9004	6.8409	46.9973
7	76.934	9.9004	6.285	46.9933
8	88.943	9.9003	5.7215	46.9837
9	101.928	9.9004	5.107	47.0277
10	113.953	9.9004	4.545	46.997
11	125.712	9.9003	3.9909	47.0074
12	137.671	9.9004	3.4295	47.0025
13	150.698	9.9004	2.8164	47.008
14	162.375	9.9004	2.2708	46.9875
15	173.928	9.9004	1.726	46.9985
16	185.505	9.9004	1.1807	47.0054
17	198.261	9.9003	0.5786	47.0176
18	205.83	9.9003	0.2195	47.0328
19	205.853	9.9003	0.2195	47.0278
20	205.868	9.9003	0.2195	47.0245

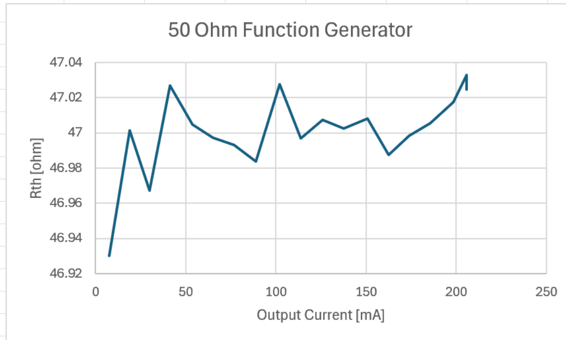

Figure 10: Function Generator

3 Conclusion

I am very happy with my fourth and final board for this class. I learned a lot about PCB design and assembly, as well as about general circuit knowledge. I learned what parts are good choices for what applications and how to find the parts that I need. It definitely rounded out my circuit knowledge, and I feel much more comfortable with everything I've learned over the past four years.

This board had one hard fault: the header pins were 100 miles further apart than the Uno design dictates. This means that a standard Uno shield will only connect to one side at a time and will not be fully supported mechanically or electrically. Overall, this is acceptable because I rarely insert shields into my Arduinos, and testing with a shield was not required in this lab.

Assembly went well overall, with only one scare in the boatload process, which prompted me to reflow both of my crystal oscillators, which may or may not have been the solution. The problem that I recognized was that I wasn't connecting the reset pin during the boatload process, which caused it to fail consistently. Following the proper procedure solved the problem and allowed me to bootload without problems. Both of my I2C devices were detected immediately, and both the buzzer and smart LEDs worked immediately. The SBB lab definitely helped with my understanding of this system and made the debugging process a lot more straight forward.