Julia DiTomas
Board 3 Report
ECEN 3730
Spring 2025

1. Plan of Record
    a. Block Diagram

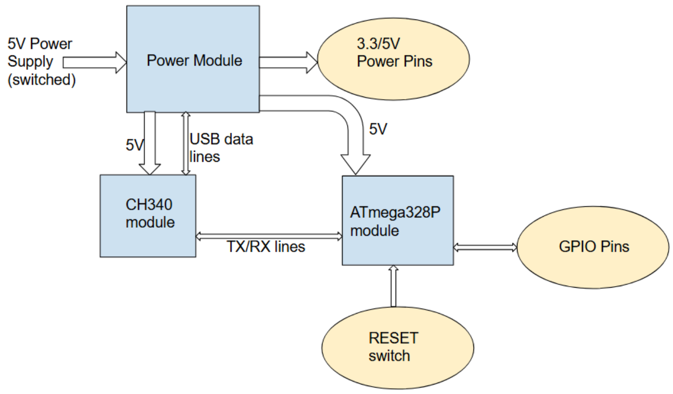


Figure 1: Board 3 simplified block diagram. The objective of this board is to utilize lessons from this class thus far to design an Arduino PCB that performs better (less noisy) than a commercial Arduino Uno. Ovals represent inputs/outputs and rectangles represent modules. Wide arrows are power paths and narrow arrows represent data flow.

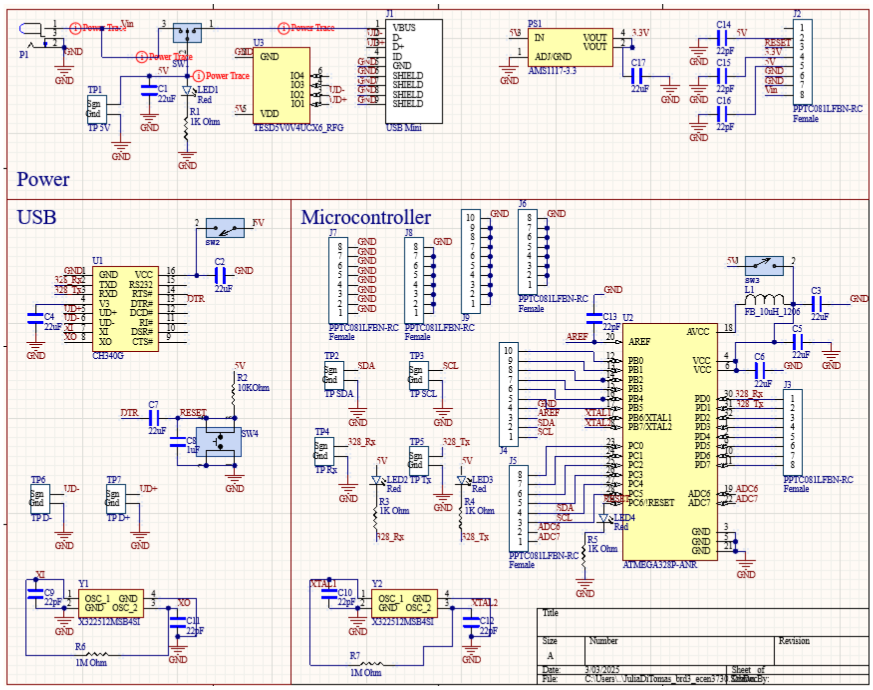
    b. Parts
        i. ATmega328P: LCSC Part #C14877 (https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf)
        ii. CH340 USB-to-UART: LSCS Part #C14267 (https://www.mpja.com/download/35227cpdata.pdf)
        iii. LDO AMS1117: LCSC Part #C6186 (https://www.lcsc.com/datasheet/lcsc_datasheet_2410121508_Advanced-Monolithic-Systems-AMS1117-3-3_C6186.pdf)
        iv. USB mini connector: LCSC Part #C46398 (https://www.lcsc.com/datasheet/lcsc_datasheet_2103081233_Jing-Extension-of-the-Electronic-Co-920-462A2021D10102_C46398.pdf)

    c. Requirements
        i. The board can be powered by either a USB mini cable or a 5V barrel jack.
        ii. The board can be bootloaded using a separate Arduino Uno.
        iii. Once bootloaded, the board can be programmed as an Arduino Uno using the Arduino IDE through USB.
        iv. The board provides the GPIO pins in the same layout as on the commercial Uno.

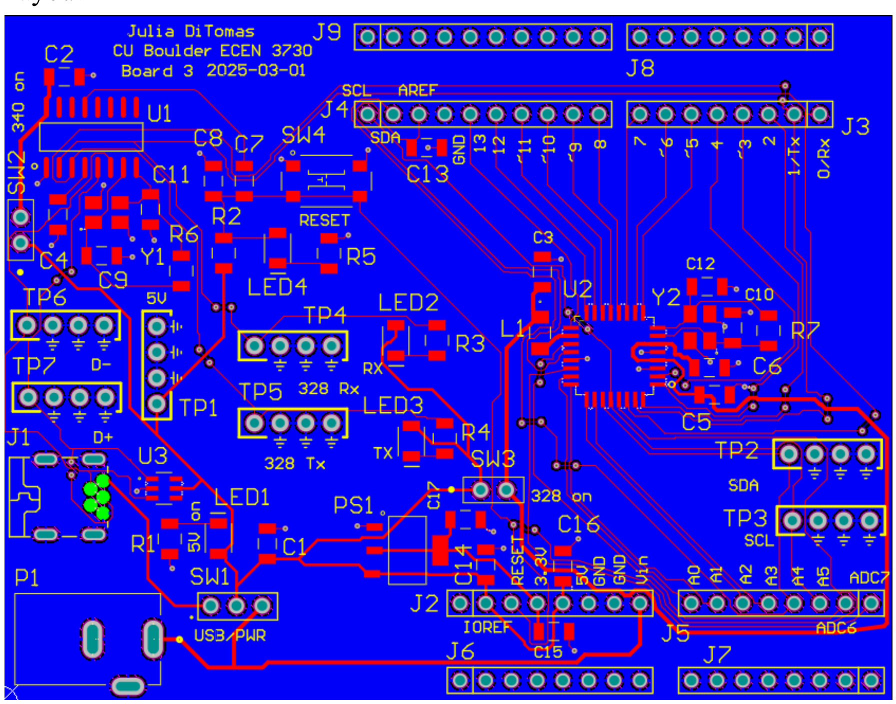
    d. Risk Mitigation
        i. Risk: Components are too close together for easy labelling/soldering leading to mistakes.
            1. Run Design Rule Check and then manually look at spacing between components flagged. Shrink labels as needed and move around on the silk screen so that it is unambiguous which corresponds to which component.
        ii. Risk: Electrically/thermally damaging ICs.
            1. Include test points on data lines to verify signals sent.

iii. Pins not matched correctly with commercial Arduino.
   1. Double check using Arduino Uno schematic, and label all GPIO as they are commercially. Measure distances in Altium.

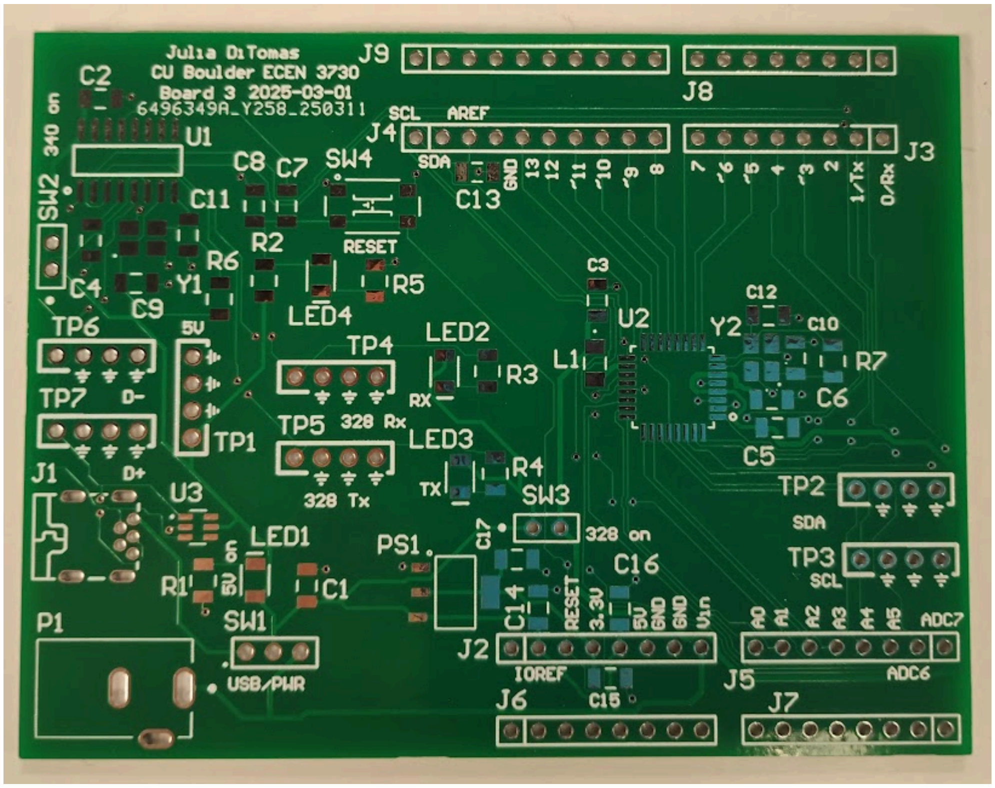
2. Board Progression
   a. Schematic



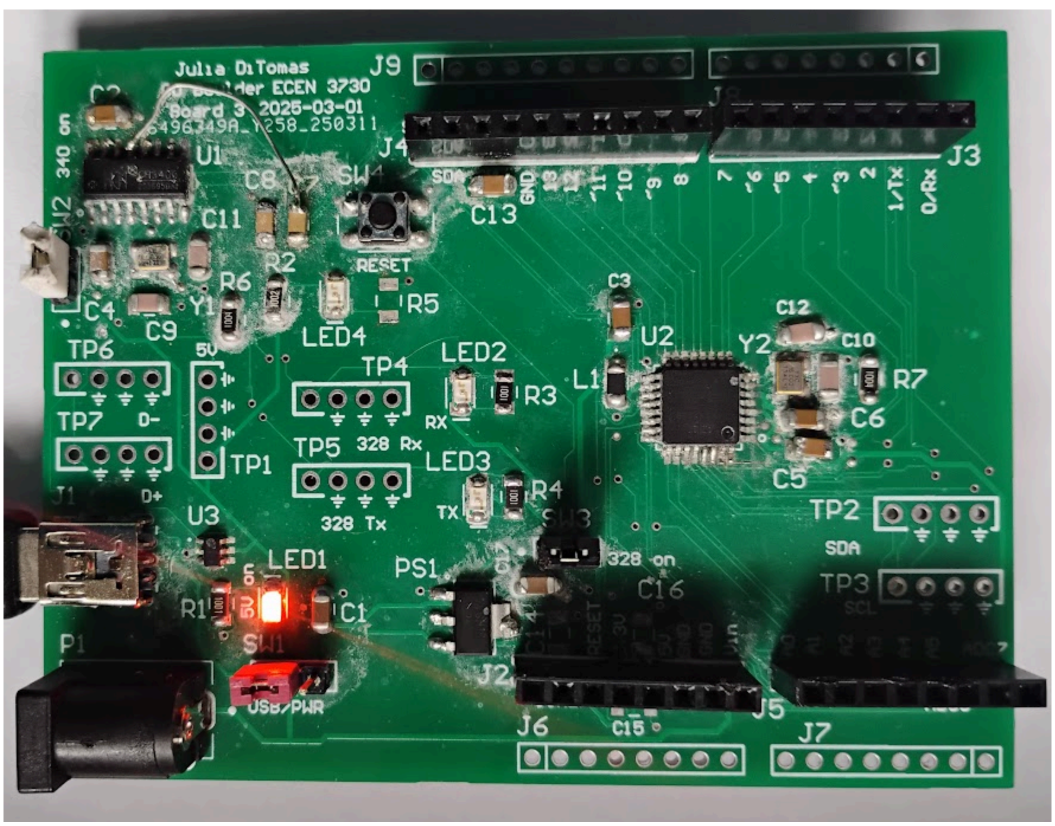
   b. Layout



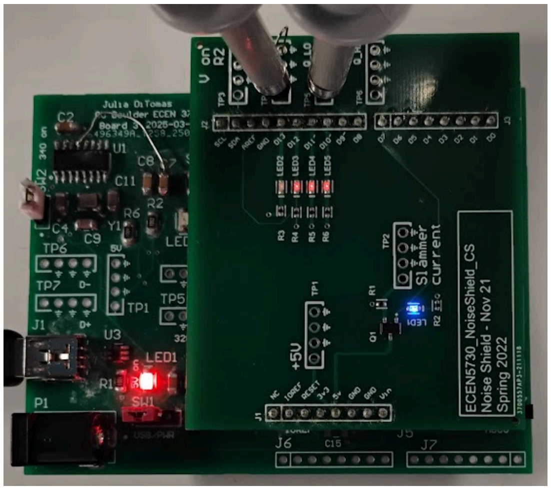
c. Bare Board

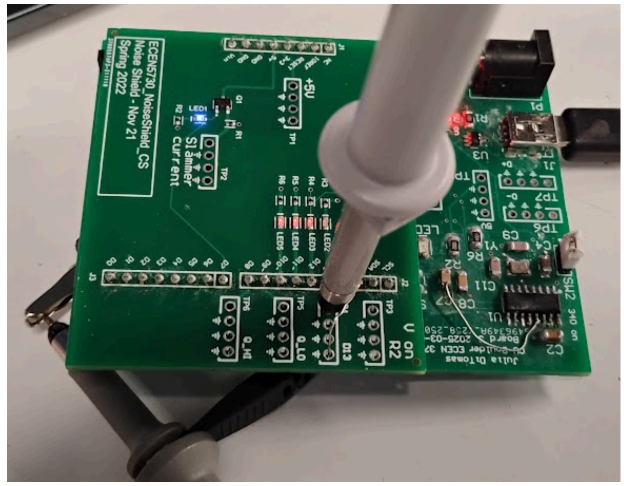


d. Assembled Board



e. Board in Testing





3. Board Power-On

After assembling my board, I connected it to my Arduino Uno and burned the bootloader. This task was easy because 1) I have experience doing it from an earlier lab and 2) I labelled the pins clearly. I then attempted to upload the example sketch “Blink” to my board, but started to run into issues with the ATmega328P being unresponsive. I determined that this was not due to the CH340, the 16 MHz crystal, or the TX/RX lines’ connections, but I had a very hard time figuring out what was causing it. Eventually, with lots of much-appreciated assistance, I was able to fix my board and to learn where I had gone wrong.

The first issue I remedied was having an indicator LED on my reset line, which created a voltage divider that never let the RESET line near 5V. The second was that I had thought the 47 nF capacitor in my kit was 47 uF, and had decided to use it for C7 instead of 22 uF because I figured either of those values would work. Since the actual capacitance was much lower than the 1 uF of C8, that voltage divider was backwards and the reset line was not dropping much lower than 5V. My failures in troubleshooting were my lack of understanding of what *should* happen on the RESET line and not considering the possibilities of multiple errors. After removing the LED and switching the capacitors (and fixing the trace I accidentally broke when removing C7), I was able to upload code.

4. Board Testing

Finally, using the provided shield, I tested my board against the commercial Arduino Uno. I first measured the quiet high and low when the IOs driving LEDs 2-5 switched on and off. On both the rising and falling edge, I observed less ground bounce noise with my board compared to the Arduino Uno (up to 30% less). For the on-die power rail (Qhigh), my board achieved lower noise only on the falling edge. However, the higher noise on the rising edge is mostly due to the much faster switching time of my board. It shows only 15% more noise but is 35% faster., implying that the actual rail inductance is less than that in the commercial board.

I next looked at the quiet high and the 5V rail when the slammer circuit switched on and off. It drew current from the 5V board rail, so I could look at the inductance of this rail as well as the coupling between this rail and the on-die rail. Again, I measured the switching time as well to be able to make any necessary adjustments, but they were extremely close between my board and the Arduino Uno.

I found that the voltage on my board rail dropped much more than the Uno's (up to twice as much). This is because after mistakenly soldering the female header pins first, I decided to just leave out the decoupling capacitors on the power supply pins, with an assumption that they were not necessary because they could be included on the connected circuit instead. I still measured less noise on my board's Qhigh test point on both edges, indicating less coupling between the two rails compared with those of the Arduino Uno.

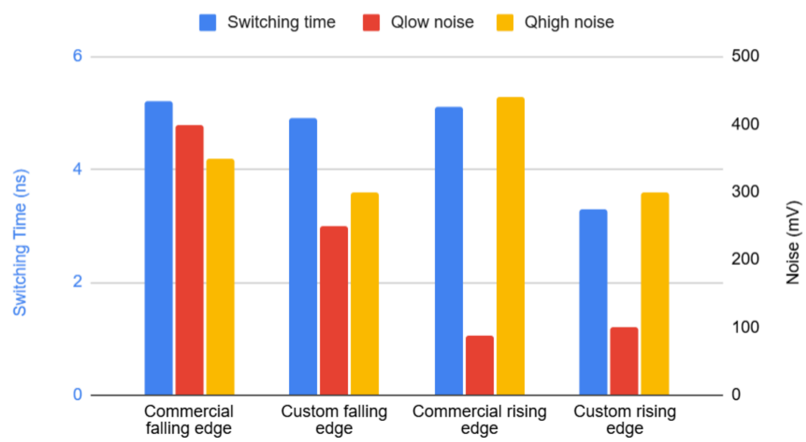


Figure 2: Comparison of noise on quiet high/low with switching IOs drawing current.

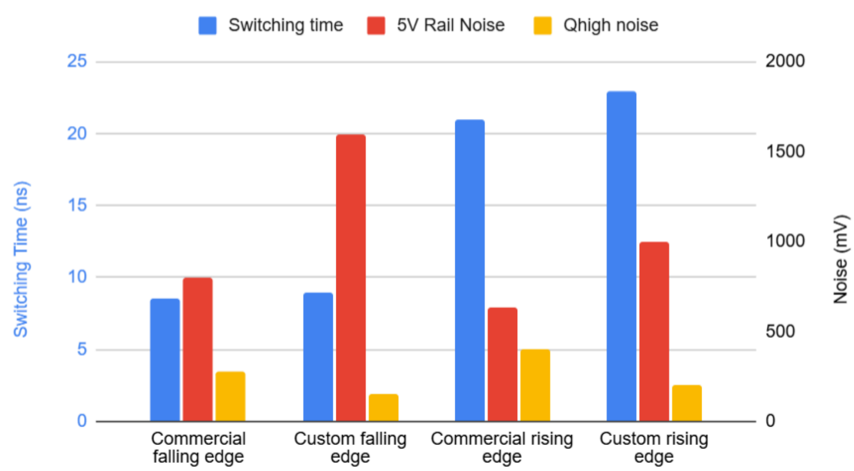


Figure 3: Comparison of noise on quiet high and 5V rail with slammer circuit turning on/off.

I then used a shorted scope probe loop to measure the near-field emissions of the two boards. As expected, the emf induced in the loop changed drastically depending on its configuration with the board; I tried to place it where it was maximized for each board. There was no significant difference between my board and the commercial Arduino in this regard (about 6 mV in each case).

In conclusion, this board presented me with many opportunities to learn about new components and to consider design trade-offs. It also presented many challenges and demonstrated the importance of good planning. Failing to stick to the plan, or even to create one, caused me to make mistakes throughout the process. But on the other hand, these mistakes in turn taught me a lot about troubleshooting and gave me a lot of practical experience.