ECEN 3730 Board 3: Golden Arduino

Christopher Sponza

November 12, 2024

1 Plan of Record

1.0.1 Introduction

The purpose of the board is to design and fabricate a PCB using the ATMega328p to be boot loaded an Arduino Uno with the goal of improved switching noise performance of the microcontroller pins.

1.0.2 Engineering Specifications

The following were the specifications that were required to be met in order for the board to ”work”:

- ATMega328p should be able to be bootloaded as an Arduino Uno and run Arduino Sketches
- A CH340G USB to UART Chip should establish communication between the microcontroller and a computer
- Appropriate pins are available to jumper the board to a commercial Arduino Uno for bootloading
- A TVS chip is included to protect the data pins from ESD and overvoltage events
- The board should be able to be powered either from a Micro USB or a Barrel Jack with a 3-Pin header and shorting flag as the selector
- A reset switch is available with a debounce capacitor
- 3.3V from an LDO is available on a header pin
- The following should be able to be probed for debugging and testing
  - 5V from the barrel jack
  - 3.3V from the LDO
  - V_{in} from the 3-pin header switch
  - RX and TX signals from the ATmega
  - D+ and D- pins from the USB port to the CH340G
- In rush current should be able to be measured using a 47Ω Ohm sense resistor and the 5V and V_{in} test points as two single ended measurements

1.1 Bill of Materials

Part	Value	Quantity
ATMega328p	-	1
CH340G	-	1
AMS1117-3.3 (LDO)	-	1
TPD3E001 (TVS)	-	1
Resistor	47Ω	1
-	$1K\Omega$	5
-	$10K\Omega$	1
-	$1M\Omega$	2
Capacitor	$1\mu F$	1
-	$22\mu F$	7
-	$22pF$	4
LED (Red)	-	5
SMD Crystal Resonators	16MHz	1
-	12MHz	1
Position Header Connector	6 Pin	1
-	8 Pin	2
-	10 Pin	1
10x Probe TP	-	8
Power Jack (Barrel)	-	1
Mini USB Header	-	1
Pushbutton	-	1
2-Pin Header (Switch)	-	1
3-Pin Header (Switch)	-	1

1.2 Datasheets

- **ATMega328p**: Microcontroller Datasheet
- **CH340G**: CH340G Datasheet
- **LDO**: LDO Datasheet
- **TVS**: TVS Datasheet

1.3 Milestones

The following are the main design milestones to reach the final product in chronological order:

- Define the engineering specifications
- Complete schematic and PCB design in Altium

- Critical Design Review (CDR)
- Board Assembly and Bring-Up
- Testing and accumulation of results

1.4 Testing Plan

The following test plan is intended to be completed after the bootloader is burned and a modified blink script is running to blink an LED off boad with a GPIO pin:

- With shorting flag removed from the 3-Pin header switch plug in the 5V barrel jack and verify 5V at the test point and the LED is lit.
- Apply 5V from the barrel jack to the microcontroller and verify the reset switch LED is Lit
- Remove the barrel jack, plug in a micro USB, and move the shorting flag to supply power from the USB. Verify the Reset LED is lit.
- Verify the 16MHz and 12MHz crystals are oscillating correctly.
- Successfully upload code such that a GPIO signal from the ATMega blinks an LED at a set period.
- In the middle of the blinking sequence depress the reset button and verify both the reset and off-board LEDs are extinguished. Release the button and verify the reset LED is lit and the off-board LED blinking sequence has reset.
- Measure the in-rush current when power is supplied from the barrel jack.

1.5 Final Product

The following schematic and PCB were designed in Altium:

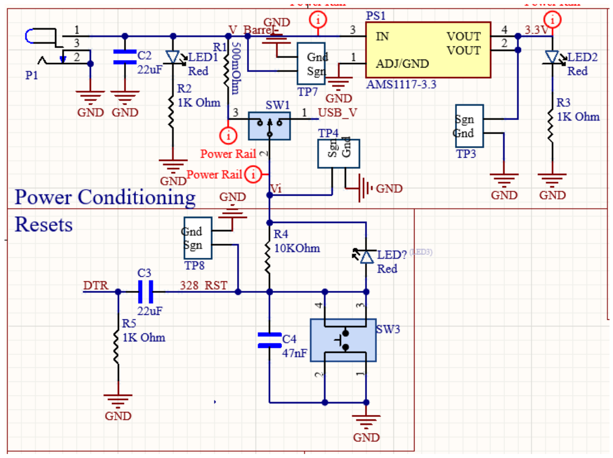

Figure 1: Power and Reset Sections of the Schematic

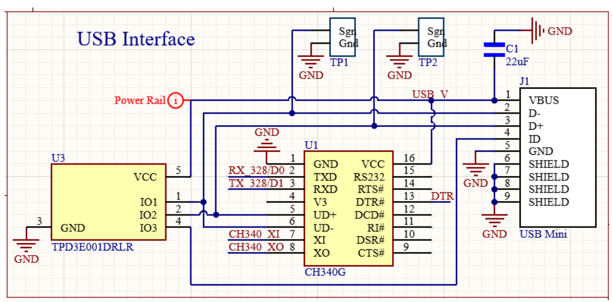

Figure 2: USB Interface Section of the Schematic

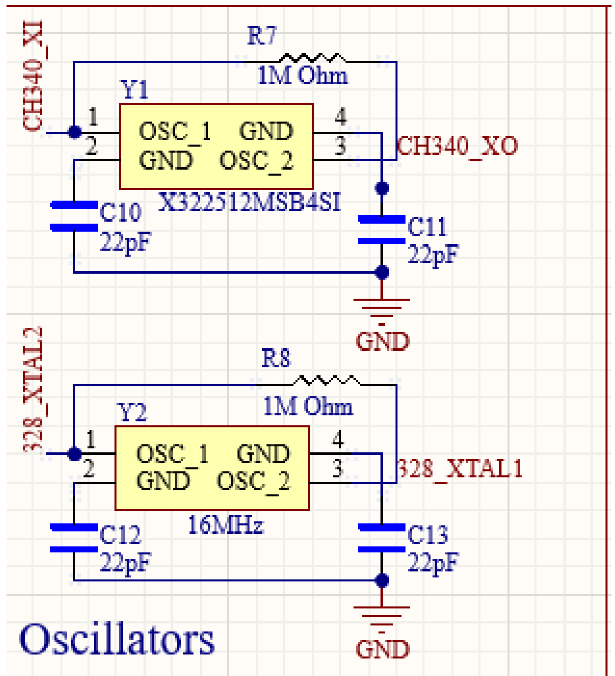

Figure 3: Crystal Oscillator Portion of the Schematic

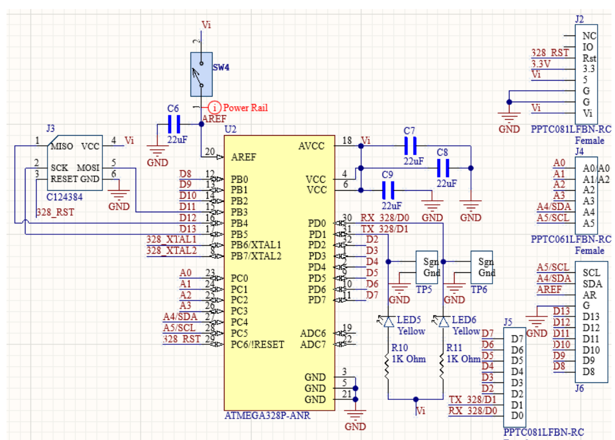

Figure 4: ATMega Section of the Schematic

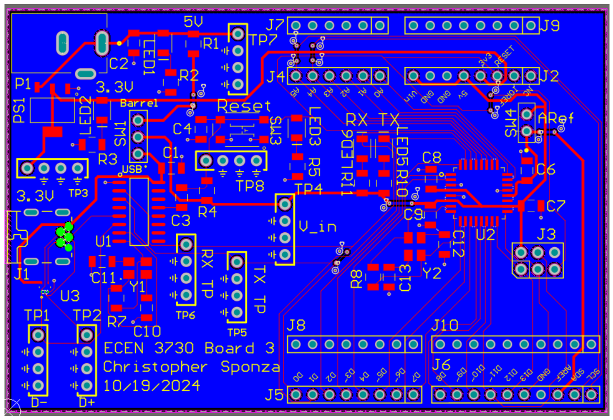

Figure 5: Board Designed in Altium

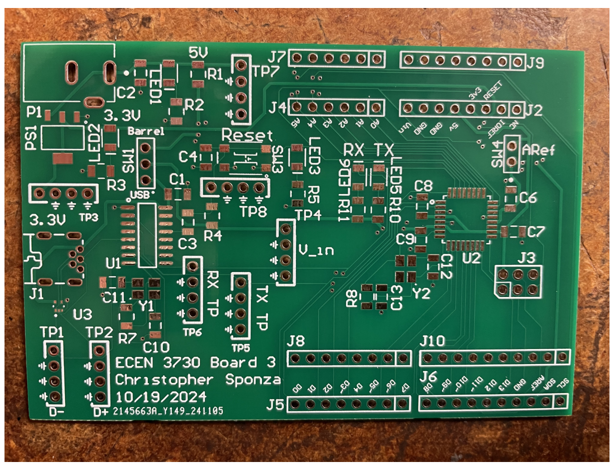

Figure 6: Board Received From the Manufacturer

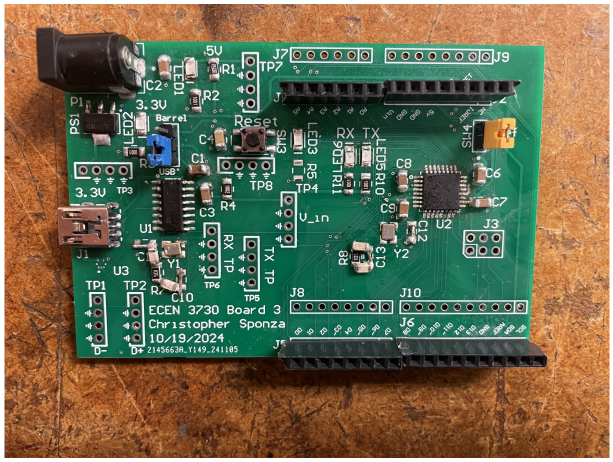

Figure 7: Constructed Board

2 Project Analysis and Measurement

2.1 What Worked

The board was able to be powered by both the USB port and the barrel jack and the selection of power was able to be made with a 3-pin header and shorting flag. The ATMega was successfully bootloaded as an Arduino Uno and subsequent communication was able to be made with a comuputer via the CH340G chip. Additionally the reset button successfully restarted the Microcontroller.

2.2 Testing

2.2.1 Results of Testing from PoR

The following are the results of testing listed in the plan of record from section 1.4:

- With the shorting flag removed from the power source selection switch, 5V was supplied to the barrel jack and the LED was lit. 5V was read at the test point.
- When the shorting flag was inserted to supply power from the barrel jack the reset indicator LED did not illuminate. Further explanation of this can be found in Section 3.1
- The barrel jack was removed and a micro USB was inserted into the board. The shorting flag was moved to supply power from the USB port. 5V was read at the V_{in} test point. As with the barrel jack, the reset LED was not illuminated.

- Scope outputs of the crystal oscillators can be found in Section 2.2.2 Figures 8 and 9. These were used to verify proper operation of the CH340G and successful bootloading of the ATMega.
- Code was successfully uploaded to blink an LED with a 50% duty cycle and a period of 200ms. This signal can be seen in Section 2.2.2 Figure 10 and the code can be found in Appendix A.
- While blinking the reset button was depressed and the blinking sequenced stopped. After release the blinking sequence restarted indicating a proper reset.
- Using two single ended measurements on each side of the sense resistor, the in rush current was successfully measured. Figure 11 shows the voltage on both sides when power is initially applied to the barrel jack resulting in a peak voltage drop of 2.25V. With a 500$m\Omega$ Resistor this results in a max current of 4.5A and a power transient of 10.13W.

2.2.2 Measurements

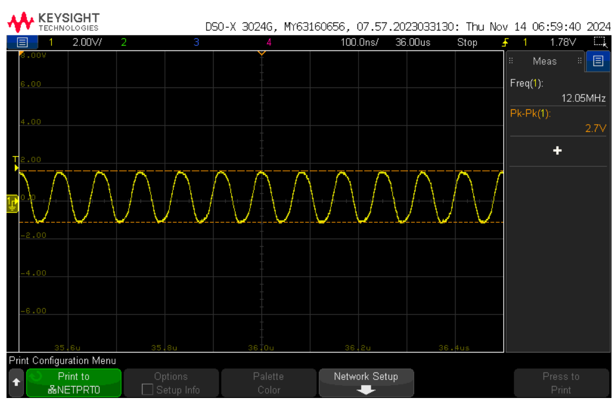

Figure 8: Scope Output of the 12MHz Crystal Oscillator to Verify Proper Operation of the CH340G

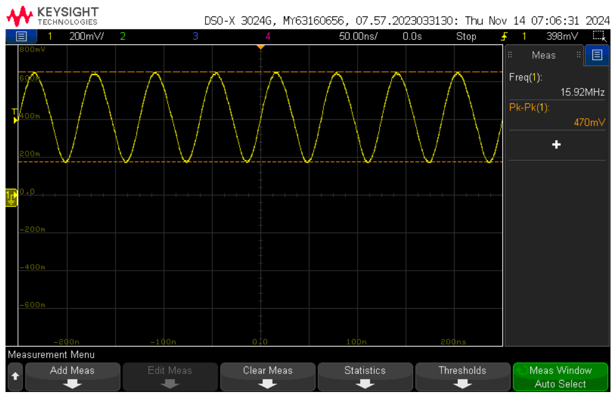

Figure 9: Scope Output of the 16MHz Crystal Oscillator to Verify Proper Bootloading of the ATMega

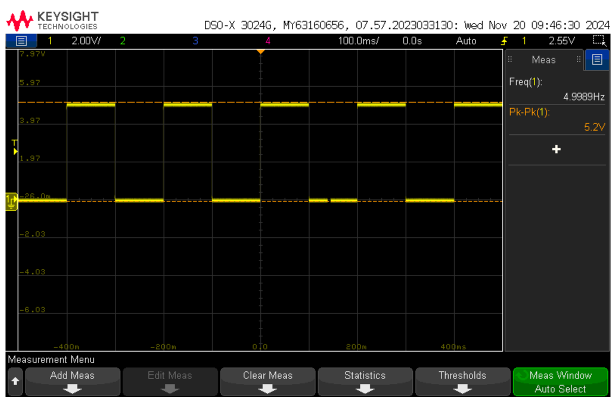

Figure 10: GPIO Digital Signal Out From Arduino to an LED

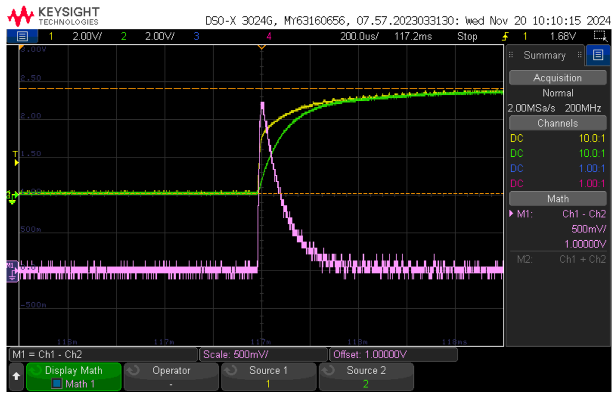

Figure 11: Voltage Across the $500m\Omega$ Sense Resistor

2.2.3 Comparison to a Commercial Arduino

To compare this Arduino board to a commercial board a shield was used. The circuit consisted of a Slammer Circuit driven by digital pin 7, six test points (digital pins 8 and 9 to measure quiet high and low, the 5V Rail of the Arduino, voltage across a resistor for a current measurement, D13 to trigger the scope and the voltage at the source of the NMOS), as well as 4 LEDs in series with 49Ω resistors driven by digital pins 10 through 13. Since the commercial board I am comparing my design to does not have individual return vias for each header pin I expected the switching noise to be greater for the commercial board.

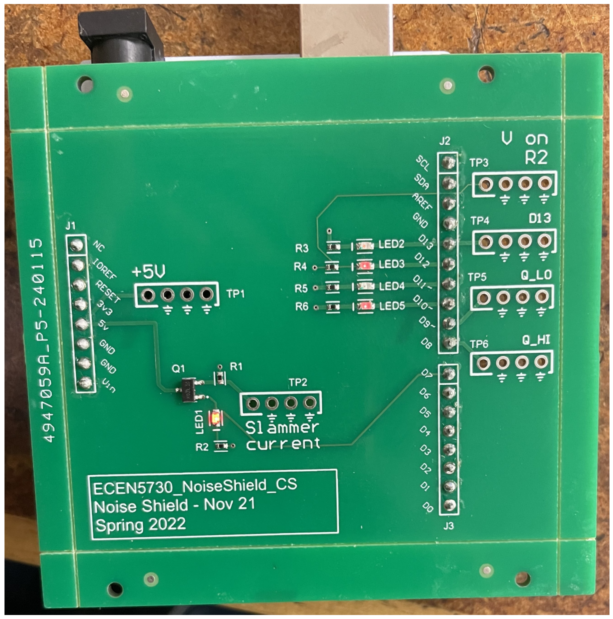

Figure 12: Arduino Shield Used to Measure Switching Noise

Using the code from Appendix B digital pins 10 through 13 were simultaneously pulled high while the noise on the quiet high and low test points were measured for both a commercial Arduino board and my design.

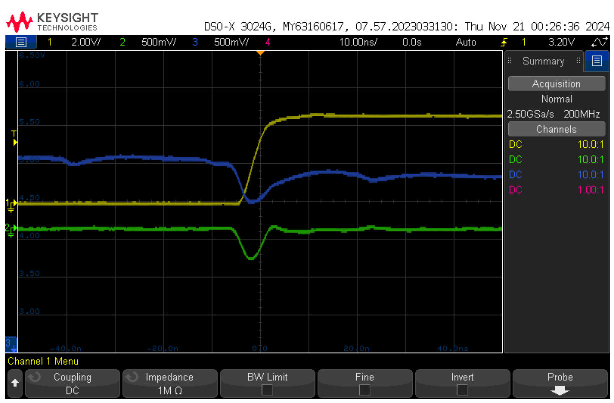

Figure 13: Switching Noise on the Rising Edge of Four Switching Digital Pins on the Commercial Arduino (Yellow: D13, Green: Quiet Low, Blue: Quiet High)

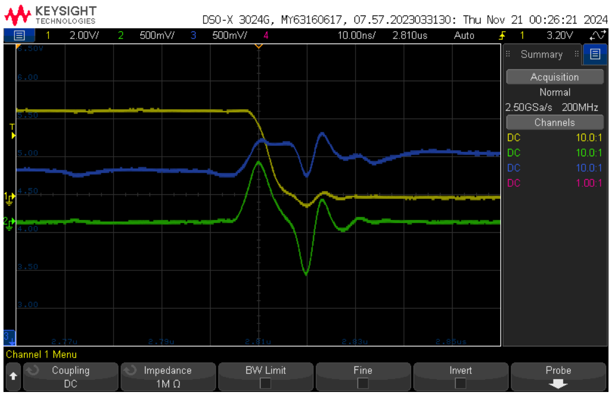

Figure 14: Switching Noise on the Falling Edge of Four Switching Digital Pins on the Commercial Arduino (Yellow: D13, Green: Quiet Low, Blue: Quiet High)

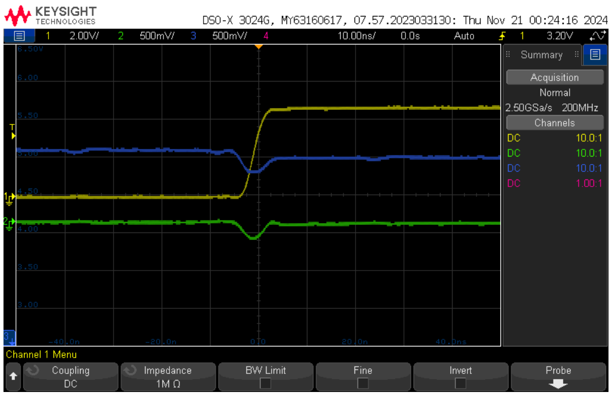

Figure 15: Switching Noise on the Rising Edge of Four Switching Digital Pins on the New Arduino (Yellow: D13, Green: Quiet Low, Blue: Quiet High)

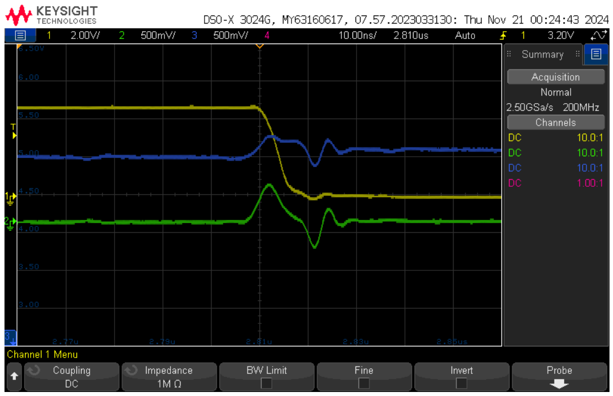

Figure 16: Switching Noise on the Falling Edge of Four Switching Digital Pins on the New Arduino (Yellow: D13, Green: Quiet Low, Blue: Quiet High)

Figures 13, 14, 15, and 16 show the oscilloscope outputs of the rising and falling edges as a result of the code and the following data was collected:

Measurement	**Commercial Board Noise (mV)**	**New Design Noise (mV)**
Quiet Low Rising	400	200
Quiet Low Falling	800	500
Quiet High Rising	300	200
Quiet High Falling	400	300

Based on this data my design performed better than the commercial design for each switching scenario. The greater switching noise on the falling edges was likely due to the slightly shorter switching time for the falling edge resuling in at higher $\frac{dI}{dt}$.

I next measured the near field emissions (NFE) from the bottom of the board while the slammer circuit was running using the code from Appendix C. The setup for this can be seen in Figure 17.

Figure 17: Measurement Setup for NFE

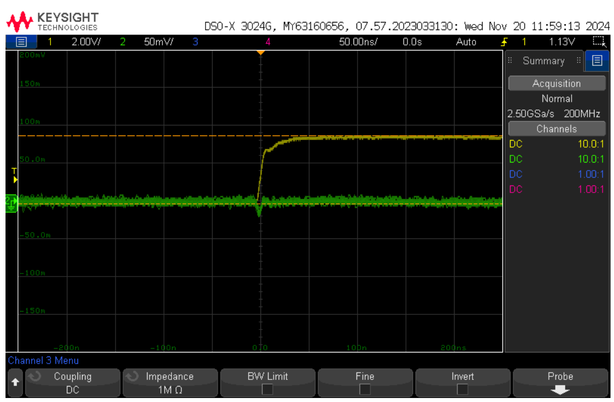

Figure 18: NFE for the Rising Edge of the Slammer Circuit Using the Commercial Arduino

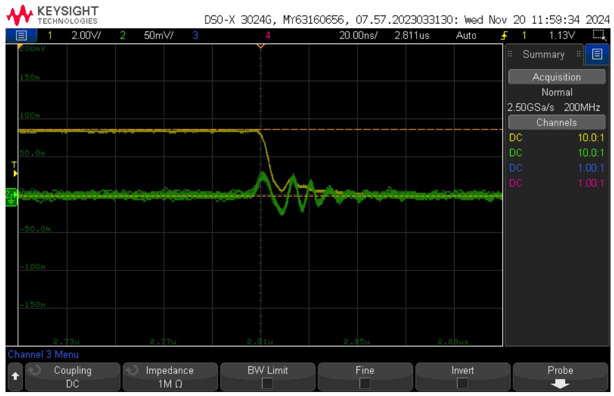

Figure 19: NFE for the Falling Edge of the Slammer Circuit Using the Commercial Arduino

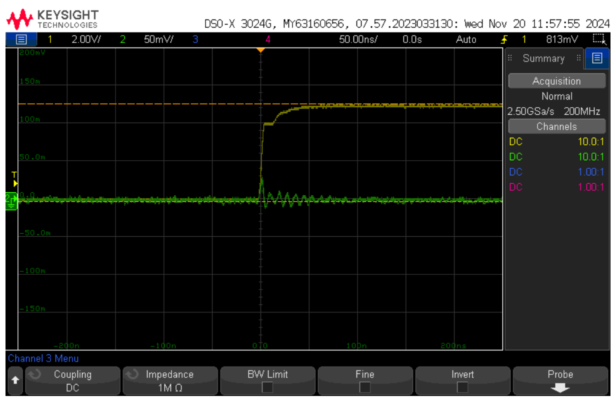

Figure 20: NFE for the Rising Edge of the Slammer Circuit Using the New Arduino Design

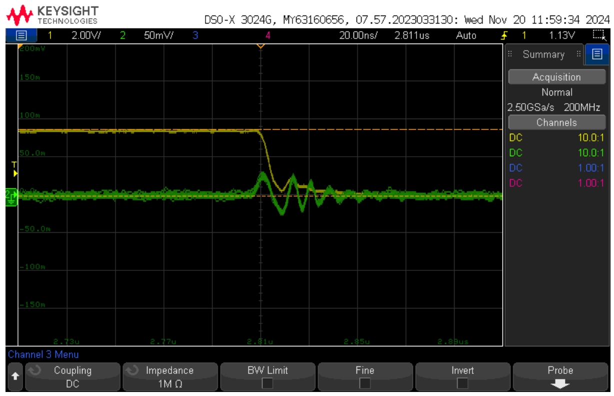

Figure 21: NFM for the Falling Edge of the Slammer Circuit Using the New Arduino Design

From Figures 18, 19, 20, and 21 the following data was collected:

Measurement	**Commercial Board NFE (mV)**	**New Design NFE (mV)**
Rising	20	30
Falling	30	40

The slightly higher measurements on the my design can be explained by the physical location of decoupling capacitors. The closest decoupling capacitor to the 5V header on the commercial board is a good deal closer than on my board. The comparision can be seen in Figure 22.

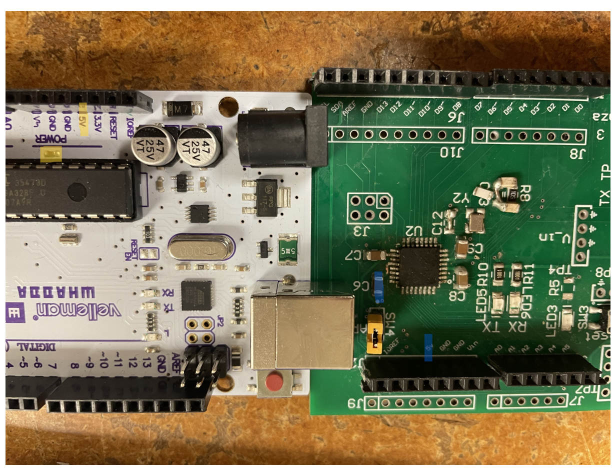

Figure 22: Capacitor Locations on the Commercial Arduino Board (White) and the New Design (Green)

3 Final Remarks

3.1 Errors

- The filter capacitors for the oscillators were incorrectly placed between the ground pins and ground rather than from the oscillator outputs to ground preventing proper operation. This required me to jumper some of the headers the original design of this section of the circuit, the corrected circuit, and the fix on the constructed board can be seen in Figures 23, 24, and 25.

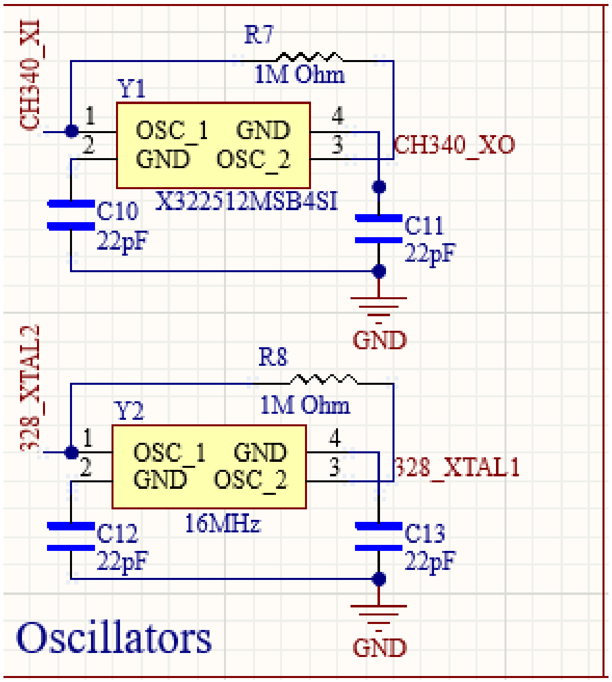

Figure 23: Incorrect Circuit Design for Oscillators

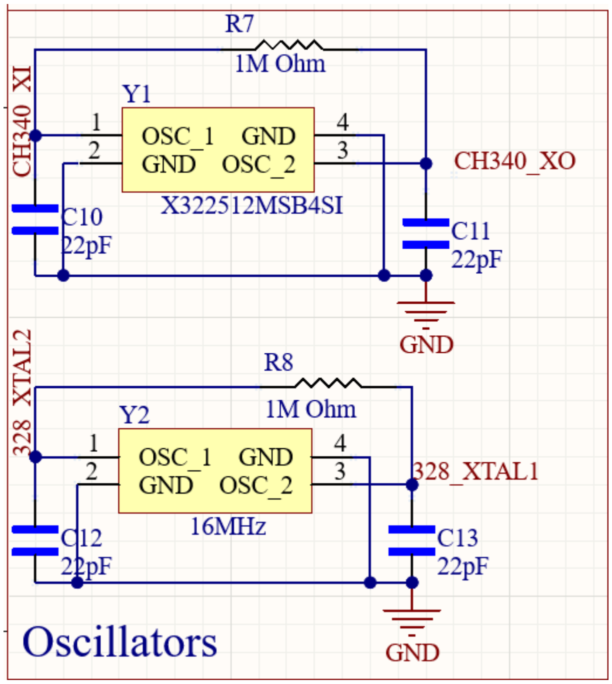

Figure 24: Corrected Circuit Design for Oscillators

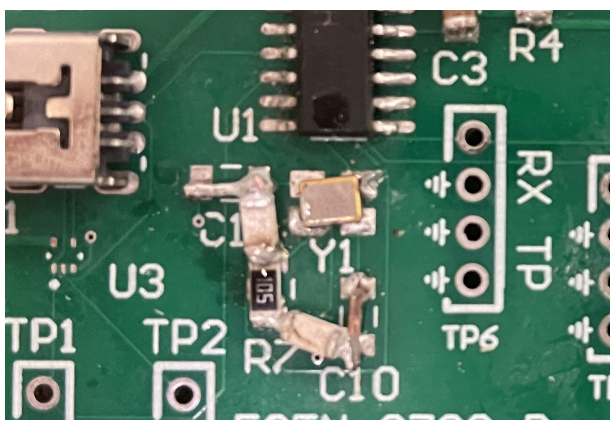

Figure 25: Jumpered Capacitors for the Oscillators

- The indicator LED for my reset button was configured incorrectly. Originally I had an indicator LED in series with a 1k resistor to ground. This caused the Reset signal to be held too low ($\approx 2.2V$) instead of the required 5V. However, when the resistor was removed (effectively removing the indicator LED) the board was able to function properly. The error and correction can be seen in Figures 26 and 27.

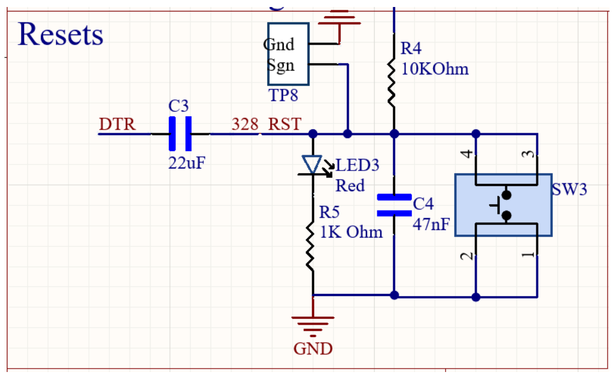

Figure 26: Incorrect Circuit Design for the Reset Button with an Indicator LED

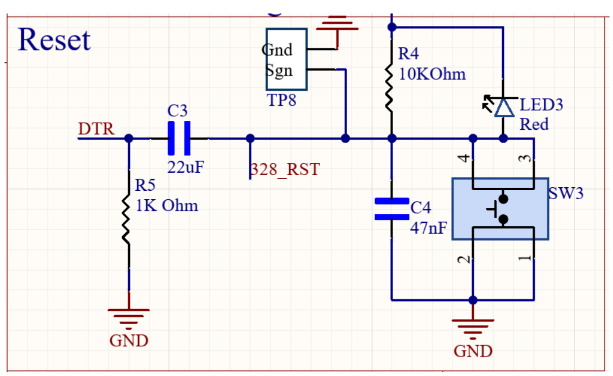

Figure 27: Corrected Circuit Design for the Reset Button with an Indicator LED

3.2 Things I Would Do Differently

- Add isolation switches for the power to the CH340G and ATMega328p for debugging during board bring-up
- Add a decoupling capacitor to the 3.3V pin of the CH340G
- Place the sense resistor after the 3-pin header switch and an additional test point on the USB side of the switch in order to measure both the in-rush current from the USB header and barrel-jack
- Correct the crystal configurations
- Add an indicator LED for power from the USB header

- Supply power to the 3.3V LDO after the supply switch. With the current design it is only powered when the barrel jack is plugged in. This also prevents it from being isolated.
- Place the return vias for the header pins physically closer to further reduce switching noise
- Place decoupling capacitors close to the Vin, 5V, and 3.3V header pins to reduce noise when current is drawn.

Appendix A Blink Code

```
const int ledPin = 8; // LED connected to digital pin 8

void setup() {
  // Initialize the pin as an output
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Turn the LED on
  digitalWrite(ledPin, HIGH);
  delay(100); // Wait for 100 ms

  // Turn the LED off
  digitalWrite(ledPin, LOW);
  delay(100); // Wait for 100 ms
}
```

Appendix B Quiet High and Low Measurement Code

```
void setup() {
  DDRB = B00111111; // Set PORTB pins as output
  pinMode(7, OUTPUT);
  digitalWrite(7, LOW);
  PORTB = B00000001;
}

void loop() {
  PORTB = B00111101; // Pin 13, 12, 11, and 10 HIGH (binary representation: 0b00111110)
  delayMicroseconds(4);

  PORTB = B00000001; // Reset other pins, keep pin 8 High
  delay(1);

}

```

Appendix C Near Field Emission Measurement Code

```
void setup() {
  DDRB = B00111111; // Set PORTB pins as output
  pinMode(7, OUTPUT);
  digitalWrite(7, LOW);
  PORTB = B00000001;
}

void loop() {
  digitalWrite(7, HIGH);
  delayMicroseconds(400);

  digitalWrite(7, LOW);
  delay(10);
}
```