University of Colorado - Boulder

ECEN 5730

Practical PCB Design Manufacture — Fall 2024

Board 3 Report - Golden Arduino

Sam Walker

Samuel Walker

Monday, November 11, 2024

College of Engineering & Applied Science

University of Colorado Boulder

Introduction

This report presents the design, testing, and analysis results for **Board 3**, referred to as the "Golden Arduino." This project aims to compare my custom board against a commercial Arduino UNO. The focus includes bootloading, quiet high/low noise analysis, switching noise, rail compression, and power consumption. Notable test methods involve measuring near-field emissions, inrush current, and slammer circuit triggering for noise testing.

Plan of Record (POR)

The Plan of Record (POR) for **Board 3** includes verifying the performance of the custom "Golden Arduino" by testing specific metrics:

- **Bootloader Verification**: Confirm successful bootloader installation by testing basic code upload and functionality.
- **Signal and Power Integrity**: Measure quiet high/low noise and assess the performance of power rails under load.
- **Switching Noise Analysis**: Compare switching noise levels between my board and the commercial Arduino.
- **Current Measurements**: Analyze inrush current and power consumption in steady-state and duty-cycle-based operation.
- **Near-Field Emissions**: Test the electromagnetic emissions of my board and compare with the commercial Arduino.

Risk Reduction

Ensuring signal and power integrity, especially in custom microcontroller boards, is essential for reliable operation. By analyzing noise sources, emissions, and inrush current, I aimed to reduce the risks associated with unintentional noise coupling and signal instability.

Project Overview

Block Diagram

The block diagram shows the power input and regulation through a DC barrel jack and USB input, directed into power selection, sensor resistor for current measurement, ESD protection, and critical oscillators supporting the ATmega328 microcontroller.

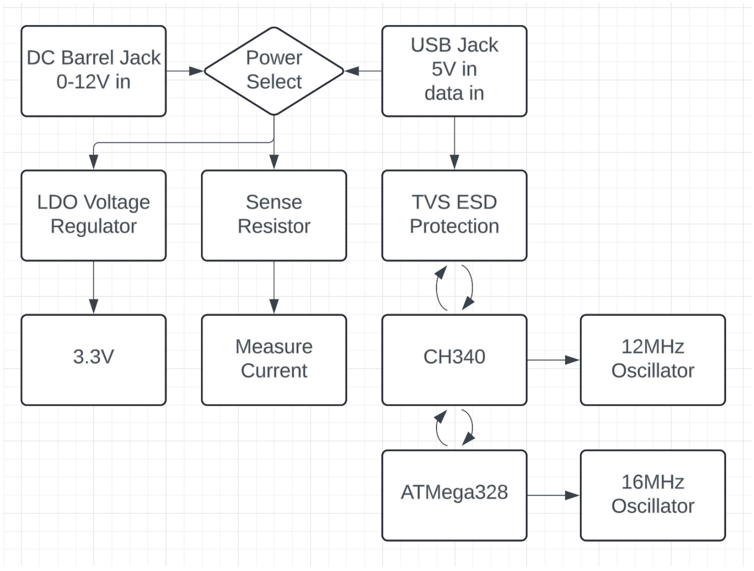


Figure 1: Block Diagram of Golden Arduino

Schematic in Altium Designer

This Altium schematic for the Golden Arduino includes the DC barrel jack, USB, CH340, 16 MHz oscillator, 3.3V LDO, and all necessary connections for the ATmega328.

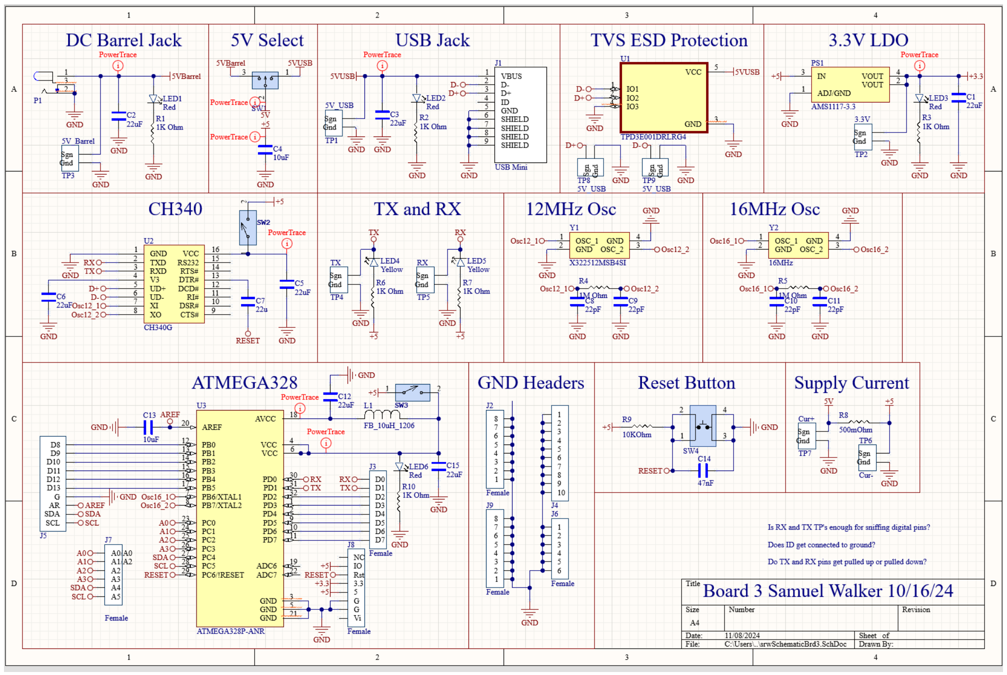

DC Barrel Jack
5V Select
USB Jack
TVS ESD Protection
3.3V LDO
CH340
TX and RX
12MHz Osc
16MHz Osc
ATMEGA328
GND Headers
Reset Button
Supply Current
Is RX and TX TP's enough for sniffing digital pins?
Does ID get connected to ground?
Do TX and RX pins get pulled up or pulled down?
Board 3 Samuel Walker 10/16/24

Figure 2: Schematic in Altium Designer

PCB Layout

The layout image showcases component placement and trace routing. Emphasis is placed on minimizing switching noise and ensuring consistent power delivery.

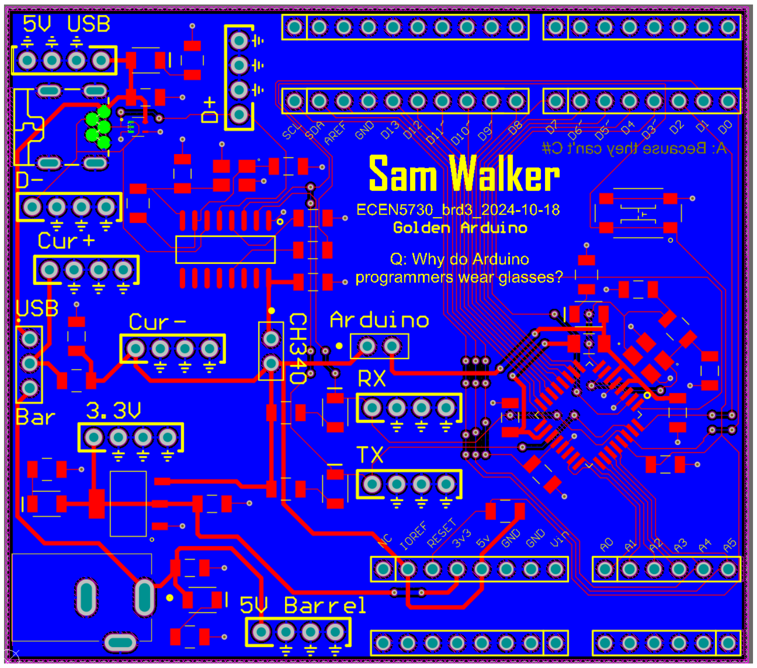


Figure 3: PCB Layout in Altium Designer

Unassembled and Assembled PCB

The unassembled board, freshly manufactured by JLCPCB, and the fully assembled Golden Arduino board are shown here. Key components include the ATmega328 microcontroller, CH340 USB-to-serial, and the LDO regulator.

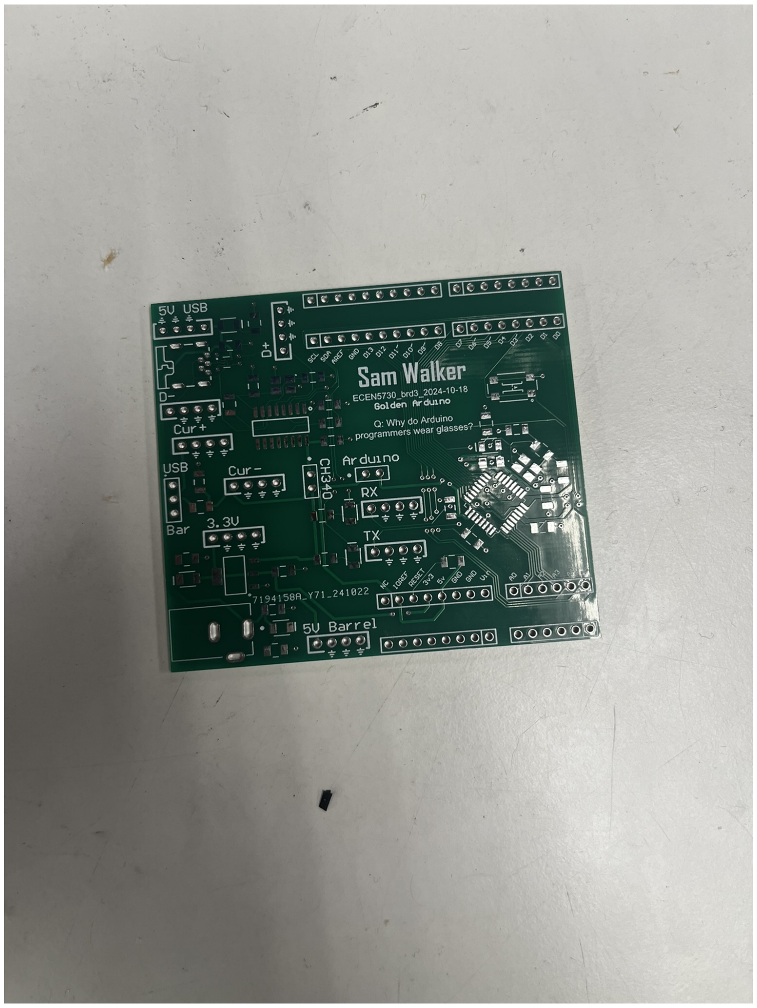


Figure 4: Unassembled PCB from JLCPCB

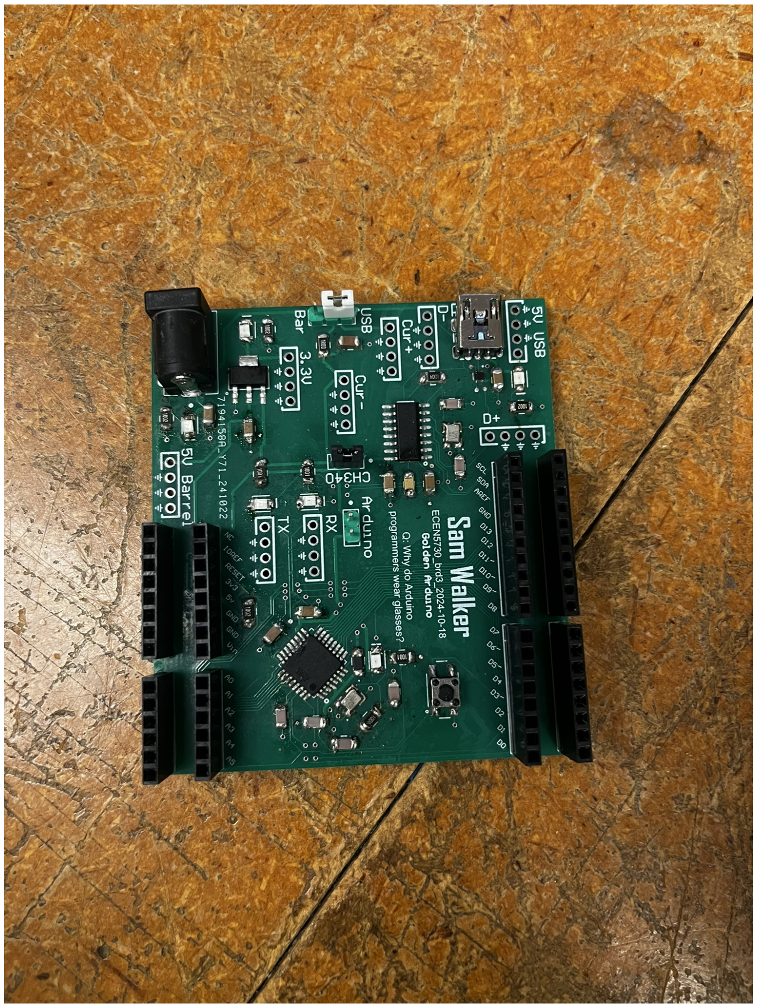


Figure 5: Assembled Golden Arduino PCB

Note here the additional row of ground headers outside of the normal arduino pins to easily probe these pins with a spring tip probe.

Testing and Measurements

Bootloading ATMega328

Using the Arduino UNO, I installed the bootloader onto the ATmega328 on my custom board, confirming successful upload via the blinking code.

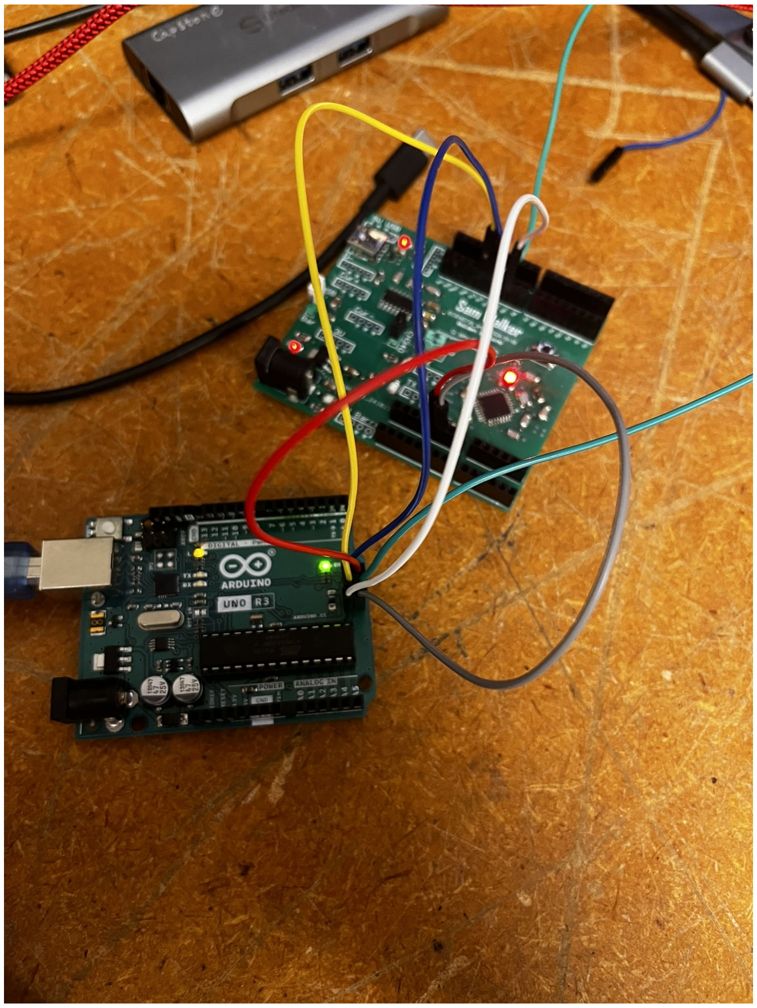
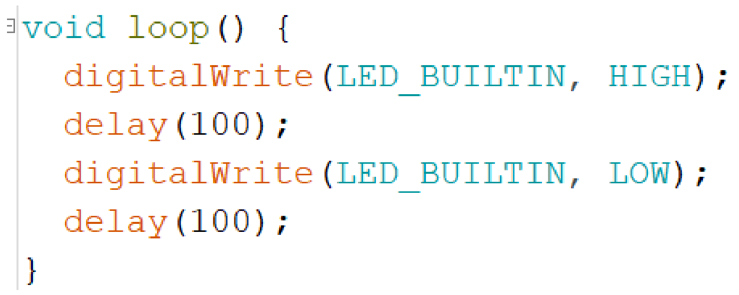


Figure 6: Bootloading Setup with Arduino UNO

Uploading Code - Example Blink Sketch

The blink sketch, toggling pin 13 with a 100 ms delay, demonstrated successful bootloader installation. Oscilloscope measurements confirm a 200 ms period with 5.5V peak-to-peak signal.



```
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(100);
  digitalWrite(LED_BUILTIN, LOW);
  delay(100);
}
```

Figure 7: Blink Sketch Code Example and Resulting Signal

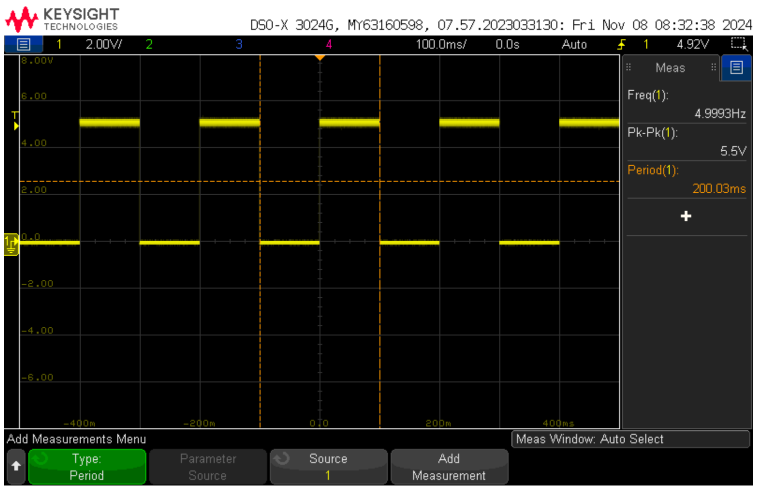


Figure 8: Digital Pin 13 'Blinking'

Testing Setup

The testing setup includes my Golden Arduino and a commercial Arduino side by side, each with shields containing identical test points (quiet high, quiet low, 5V rail, and slammer circuit points) for noise comparison.

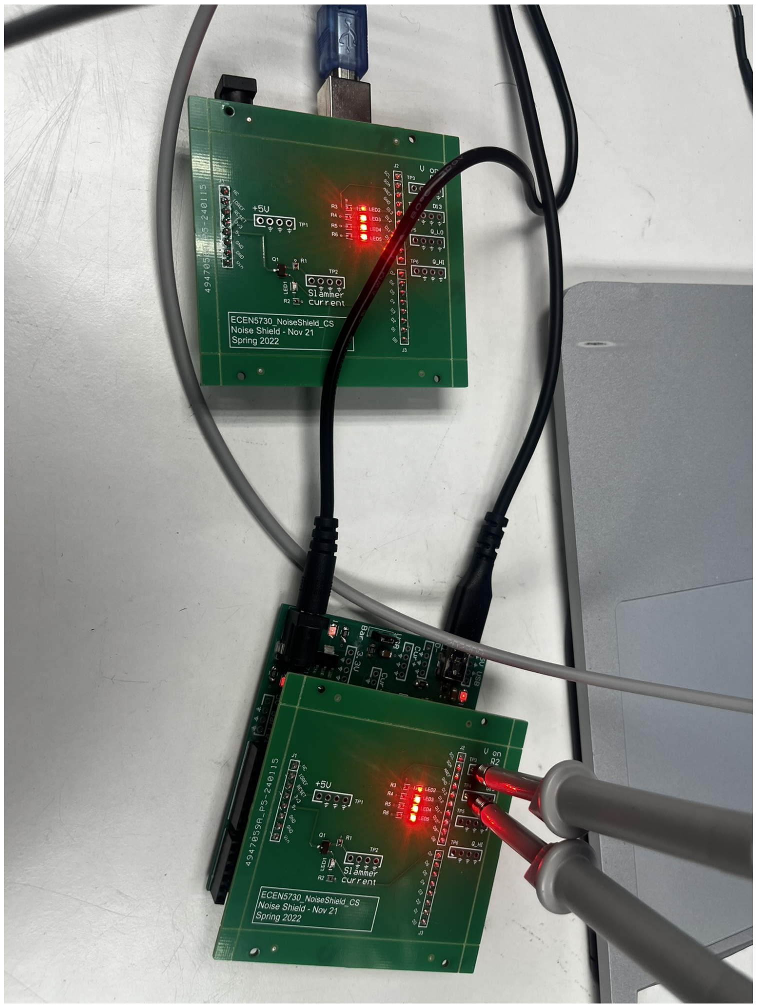
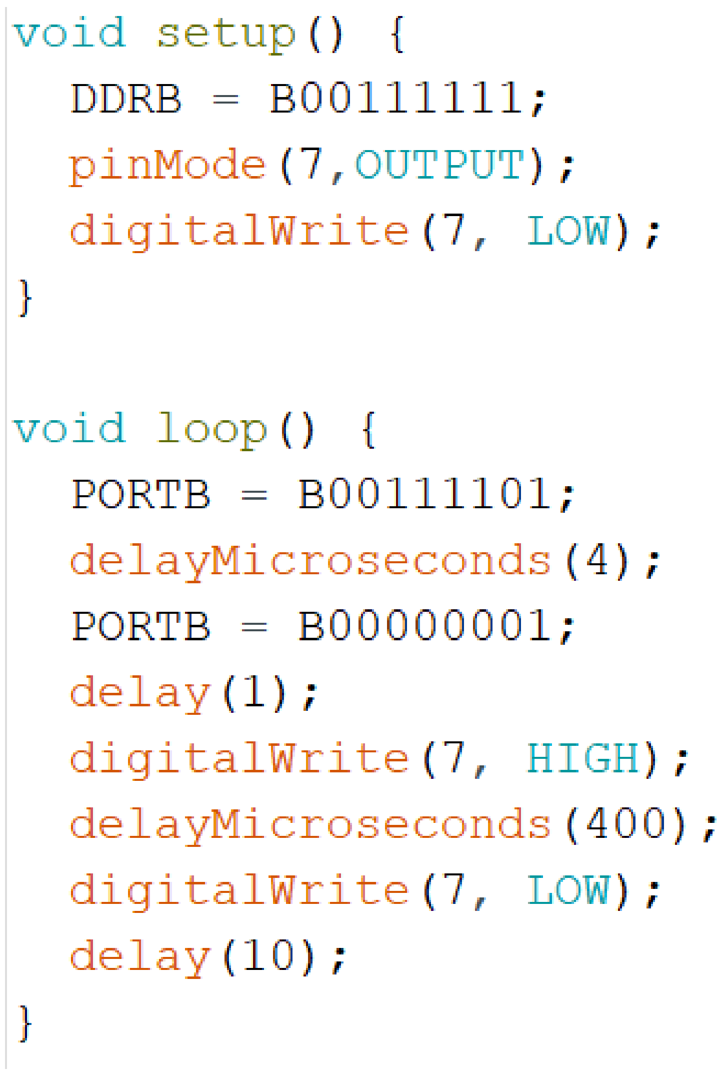


Figure 9: Testing Setup with Golden and Commercial Arduino Boards

The testing setup also used the same script for both the commercial arduino board and my board, shown below. This script oscillates four digital pins, as well as maintaining a high value for quiet high, a low value for quiet low and a switching signal for the slammer pin.



```
void setup() {
  DDRB = B00111111;
  pinMode(7,OUTPUT);
  digitalWrite(7, LOW);
}

void loop() {
  PORTB = B00111101;
  delayMicroseconds(4);
  PORTB = B00000001;
  delay(1);
  digitalWrite(7, HIGH);
  delayMicroseconds(400);
  digitalWrite(7, LOW);
  delay(10);
}
```

Figure 10: Testing Script

Quiet High and Quiet Low Analysis - Rising Edge

The figures outlined below compare my board to the commercial Arduino's Quiet High and Quiet Low switching noise on the rising edge of the digital pin 13 switching signal.

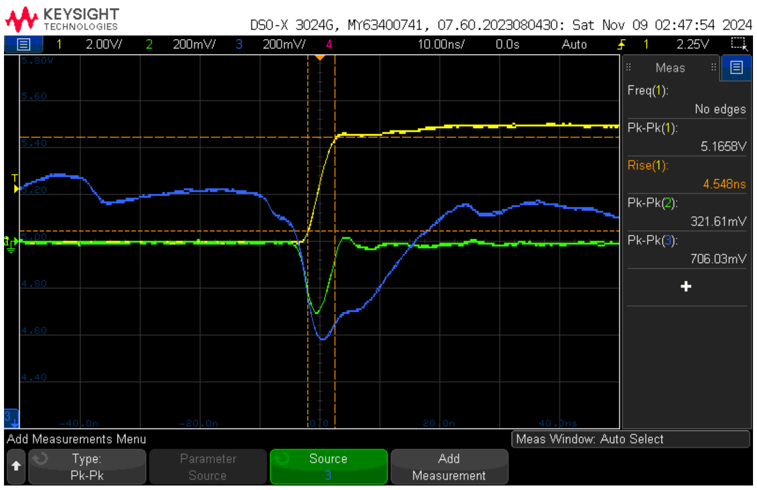


Figure 11: Quiet High and Low Signals on My Board - Rising Edge

As shown above my board experiences quiet low noise (channel two) to be about 321 mV with quiet high noise measuring 706mV on channel three.

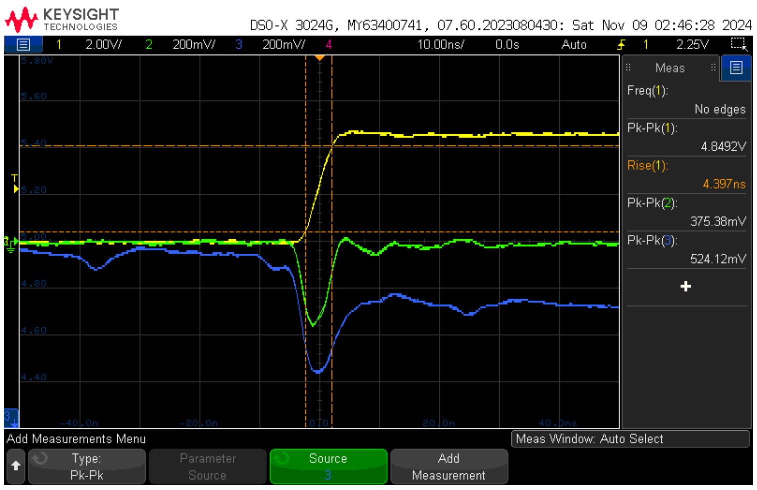


Figure 12: Quiet High and Low Signals on Commercial Arduino - Rising Edge

As shown above the commercial arduino experiences quiet low noise (channel two) to be about 325 mV with quiet high noise measuring 524mV on channel three.

Quiet High and Quiet Low Analysis - Falling Edge

The figures outlined below compare my board to the commercial Arduino's Quiet High and Quiet Low switching noise on the falling edge of the digital pin 13 switching signal.

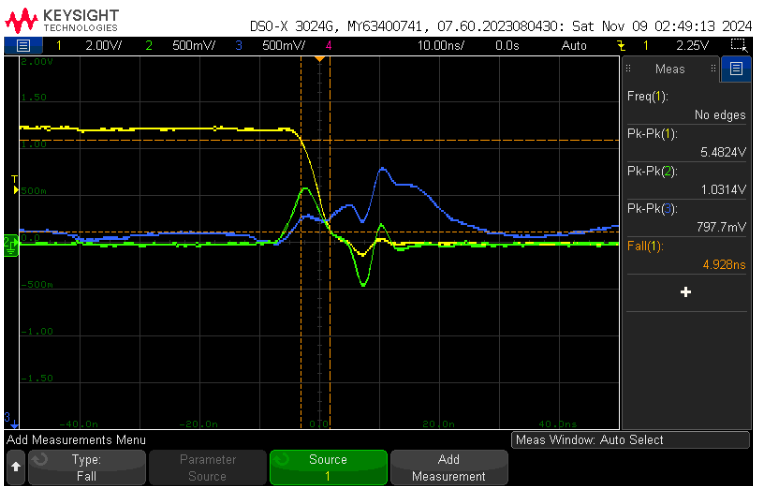


Figure 13: Quiet High and Low Signals on My Board - Falling Edge

As shown above my board experiences quiet low noise (channel two) to be about 1.03 V with quiet high noise measuring 797 mV on channel three.

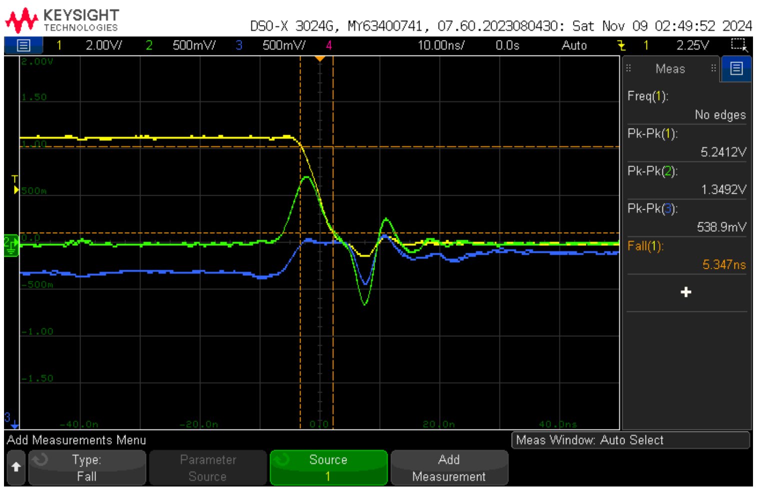


Figure 14: Quiet High and Low Signals on Commercial Arduino - Falling Edge

As shown above my board experiences quiet low noise (channel two) to be about 1.35 V with quiet high noise measuring 538 mV on channel three.

Takeaways of Quiet High and Quiet Low Analysis

The four figures above show that my board compared very similarly to the commercial arduino, beating its noise in some categories such as the quiet high falling edge and generating slightly worse noise in others such as quiet low falling edge measurements. All in all these characteristics show that my board performs at a very capable level with commercially available products.

Quiet High and 5V Rail Analysis - Rising Edge

The figures below compare the switching noise observed on the quiet high signal and 5V rail during the rising edge of the digital pin 13 switching signal, focusing on differences between my custom board and the commercial Arduino.

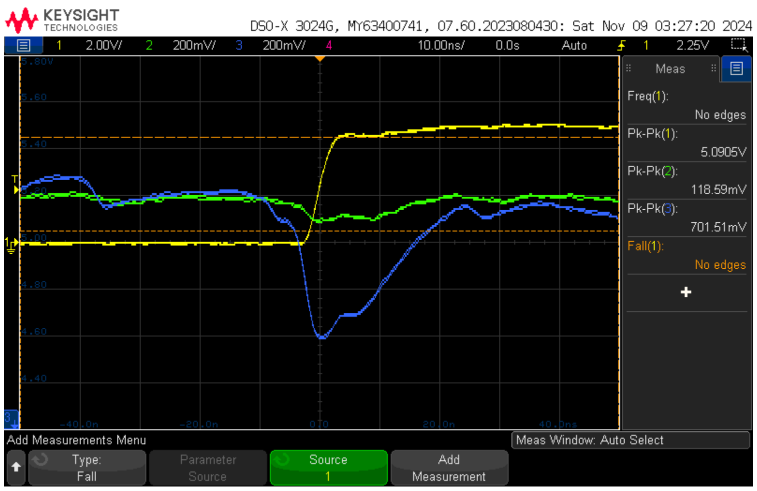


Figure 15: 5V Rail and Quiet High Noise on My Board - Rising Edge

In the figure above, my board exhibits a quiet high noise level (channel three) of approximately 701 mV and a 5V rail noise level (channel two) measuring 118 mV. This combination indicates that my board experiences moderate noise fluctuations in the quiet high signal, while the 5V rail shows relatively low noise, suggesting effective power delivery control.

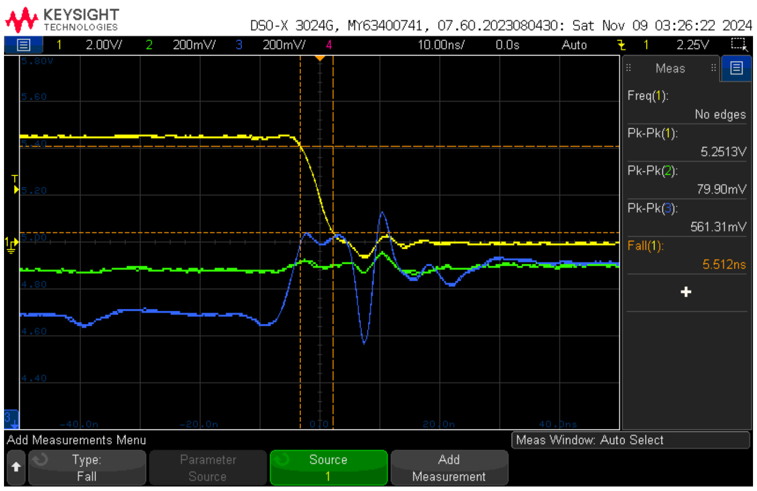


Figure 16: 5V Rail and Quiet High Noise on Commercial Arduino - Rising Edge

In the figure above, the commercial Arduino's quiet high noise level (channel three) measures about 561 mV, while the 5V rail noise level (channel two) is around 79 mV. These measurements indicate that the commercial Arduino maintains lower noise on the quiet high signal but also shows lower 5V rail noise than my custom board. The reduced noise in both channels on the commercial Arduino suggests slightly better stability during the rising edge transition.

Quiet High and 5V Rail Analysis - Falling Edge

The figures below show the noise measurements for the quiet high signal and 5V rail on the falling edge of the digital pin 13 signal, comparing my custom board to the commercial Arduino.

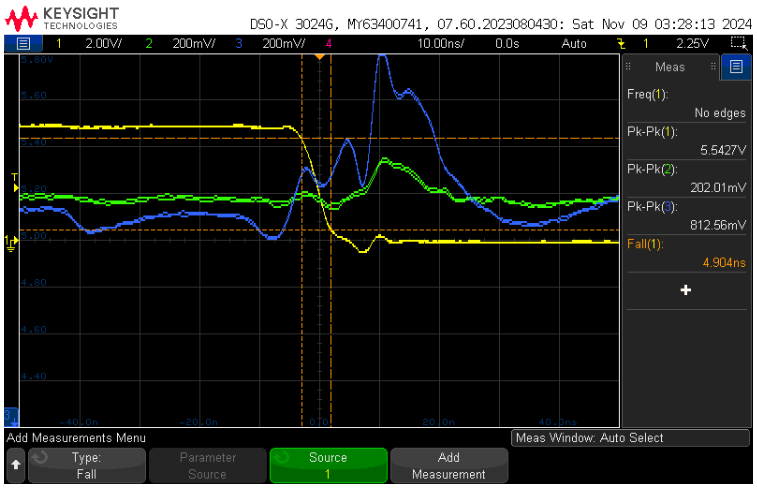


Figure 17: 5V Rail and Quiet High Noise on My Board - Falling Edge

In the figure above, my board exhibits a quiet high noise level of approximately 812 mV and a 5V rail noise level measuring around 202 mV. These values indicate that, during the falling edge transition, my board's quiet high and 5V rail both experience increased noise levels compared to the rising edge. This increase is likely due to greater transient currents associated with the switching event.

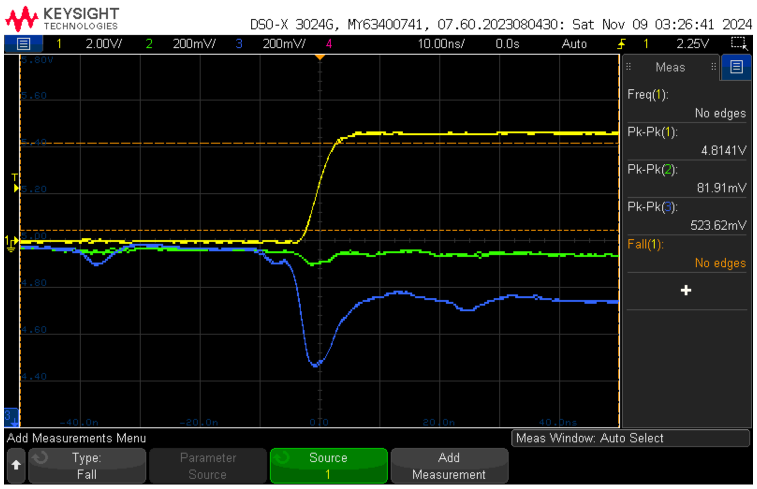


Figure 18: 5V Rail and Quiet High Noise on Commercial Arduino - Falling Edge

In the figure above, the commercial Arduino displays a quiet high noise level of 523 mV and a 5V rail noise level of approximately 81 mV during the falling edge. This indicates that the commercial Arduino consistently maintains lower noise on both the quiet high and 5V rail signals compared to my board.

Takeaways of Quiet High and 5V Rail Analysis

The four figures demonstrate the noise characteristics of both boards in the quiet high and 5V rail signals during rising and falling edges:

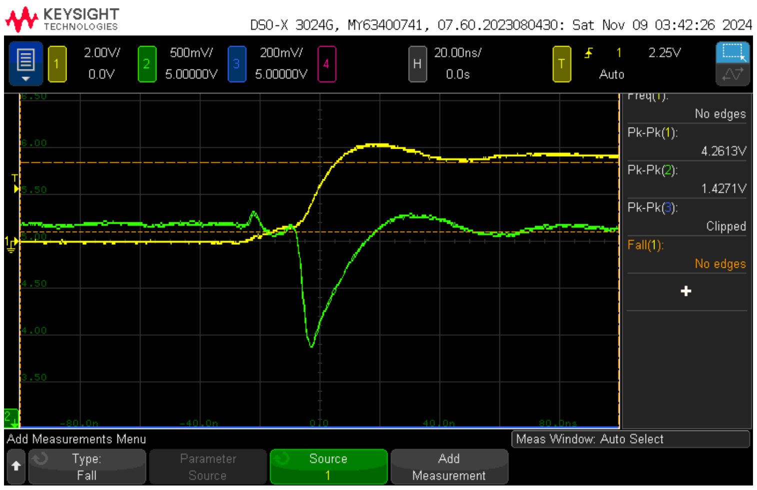
- **Rising Edge**: My board exhibited a quiet high noise level of 701 mV and 5V rail noise of 118 mV, compared to the commercial Arduino's 561 mV (quiet high) and 79 mV (5V rail). The commercial Arduino had slightly better control on both quiet high and 5V rail noise.
- **Falling Edge**: My board showed higher noise levels of 812 mV on quiet high and 202 mV on the

5V rail, while the commercial Arduino measured 523 mV on quiet high and 81 mV on the 5V rail, demonstrating lower noise on both channels.

In summary, my board exhibited slightly higher noise on both the quiet high and 5V rail signals across both transitions. This increased noise may be attributed to minor layout differences and power delivery optimizations that can be further improved in future designs. Nonetheless, these results indicate that my board performs competitively with the commercial Arduino, handling transient switching noise within acceptable ranges for typical operation.

Slammer Circuit Analysis

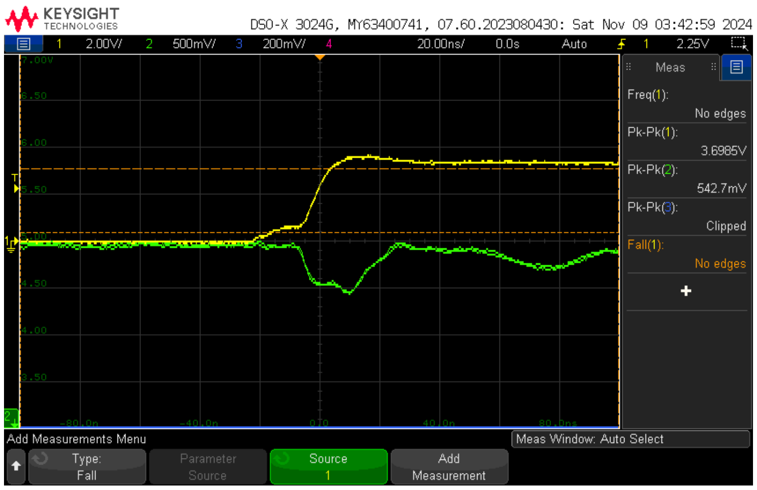
The slammer circuit was tested to assess the noise levels generated during rapid switching events on both my custom board and the commercial Arduino. This circuit's design is intended to evaluate noise control by introducing sharp transitions that challenge the power delivery and grounding network of the board, amplifying any deficiencies in layout or component placement.


DSO-X 3024G, MY63400741, 07.60.2023080430: Sat Nov 09 03:42:26 2024

Pk-Pk(1): 4.2613V
Pk-Pk(2): 1.4271V
Pk-Pk(3): Clipped
Fall(1): No edges

Figure 19: Slammer Circuit Noise on My Board

In the figure above, my custom board's slammer circuit exhibited noise reaching approximately **1.4V**. This noise level indicates a significant fluctuation during switching, highlighting areas for improvement in the circuit's grounding and decoupling practices.


DSO-X 3024G, MY63400741, 07.60.2023080430: Sat Nov 09 03:42:59 2024

Freq(1): No edges
Pk-Pk(1): 3.6985V
Pk-Pk(2): 542.7mV
Pk-Pk(3): Clipped
Fall(1): No edges

Figure 20: Slammer Circuit Noise on Commercial Arduino

In the figure above, the commercial Arduino's slammer circuit shows a noise level of **542mV**, which is considerably lower than my board's. This difference suggests that the commercial Arduino employs more effective noise control techniques in its power delivery network, reducing the susceptibility of the slammer circuit to switching noise.

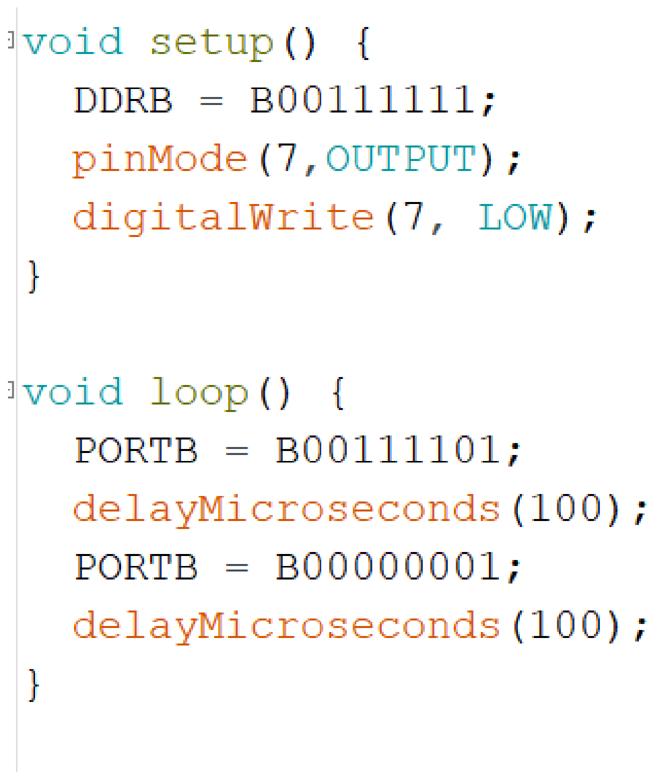
Slammer Circuit Takeaways

The slammer circuit introduces sharp transitions by rapidly switching a digital pin, forcing the board's power supply and ground planes to accommodate sudden changes in current. This test effectively highlights weak points in layout and design. Key areas to consider for reducing noise in the slammer circuit on my custom board include:

- **Ground Plane Optimization**: Ensuring a continuous and low-impedance ground plane is critical for minimizing noise during high-speed switching events. Any interruptions or high-inductance paths in the ground plane can exacerbate noise levels by creating unintended voltage differences. Improvements to the ground plane continuity, particularly around the slammer circuit, could significantly reduce the observed noise on my board.
- **Decoupling Capacitor Placement**: The placement and value of decoupling capacitors play a crucial role in managing noise. Placing capacitors close to the power pins of the components within the slammer circuit can provide localized energy storage, absorbing the transient currents created by rapid switching. My board could benefit from additional or repositioned decoupling capacitors directly adjacent to the slammer circuit's high-current paths to better filter noise.
- **Power Delivery Network (PDN) Design**: A robust PDN design, which includes the careful selection and routing of power and ground connections, helps prevent noise from propagating through the board. The commercial Arduino likely benefits from a PDN that reduces impedance between the power source and slammer circuit, which could be achieved by using wider traces or dedicated power and ground planes. Implementing similar PDN improvements on my board could help reduce noise generated during switching.
- **Component Shielding and Isolation**: Isolating the slammer circuit from other sensitive components can prevent noise from coupling into adjacent circuits. Physical separation or additional ground shielding layers around the slammer circuit can help contain the noise generated during switching events.

Current Measurements

This section examines the current draw on my custom board as compared to the commercial Arduino. Each board was configured to run four LED-resistor combinations on pins switching at a 50% duty cycle. The modified code shown below was used to achieve this consistent duty cycle and is essential to replicating the power measurements.

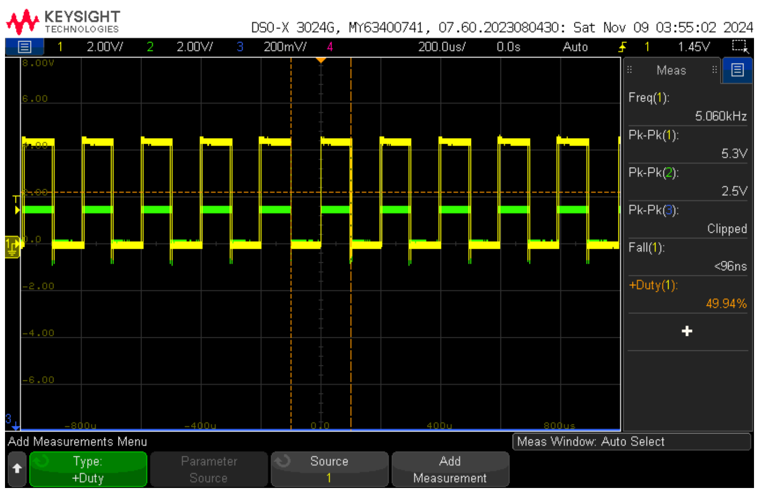


```
void setup() {
  DDRB = B00111111;
  pinMode(7,OUTPUT);
  digitalWrite(7, LOW);
}

void loop() {
  PORTB = B00111101;
  delayMicroseconds(100);
  PORTB = B00000001;
  delayMicroseconds(100);
}
```

Figure 21: Modified Code for 50% Duty Cycle - Current Measurements

The figure above shows the modified code used to generate a stable 50% duty cycle for each of the four LED-resistor combinations. This code sets each output pin high for an equal amount of time as it is set low, ensuring that each LED receives power half of the time on average.



```
DSO-X 3024G, MY63400741, 07.60.2023080430: Sat Nov 09 03:55:02 2024
1 2.00V/  2 2.00V/  3 200mV/  4   200.0us/  0.0s  Auto  1  1.45V

Freq(1): 5.060kHz
Pk-Pk(1): 5.3V
Pk-Pk(2): 2.5V
Pk-Pk(3): Clipped
Fall(1): <96ns
+Duty(1): 49.94%

Meas Window: Auto Select
Type: +Duty
Source 1
```

Figure 22: Current Measurement on Commercial Arduino (2.5V Drop)

In the figure above, the commercial Arduino board shows a voltage drop across the LED resistor of approximately **2.5V** when operating at a 50% duty cycle. This voltage drop indicates the effective load on the Arduino's output and provides a baseline for power consumption comparisons.

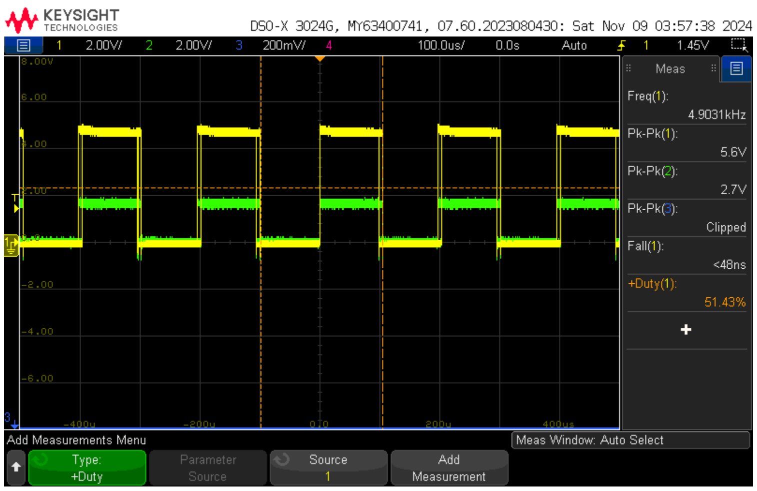


Figure 23: Current Measurement on My Custom Board (2.7V Drop)

In the figure above, my custom board shows a voltage drop of approximately **2.7V** across each LED resistor. The slightly higher voltage drop compared to the commercial Arduino suggests a marginally greater current draw on my board, likely due to slight differences in the board layout or component characteristics.

Power Draw Calculation: Using the 2.7V drop observed on my board and assuming each resistor value is $R = 220\,\Omega$, the current I through each LED-resistor pair is calculated as follows:

$$I = \frac{V}{R} = \frac{2.7V}{220\,\Omega} = 12.27\,\text{mA}$$

Since each LED is only on 50% of the time, the effective current for each LED-resistor pair over time is:

$$I_{\text{effective}} = 12.27\,\text{mA} \times 0.5 = 6.14\,\text{mA}$$

With four LED-resistor pairs, the total effective current I_{total} for my board is:

$$I_{\text{total}} = 4 \times 6.14\,\text{mA} = 24.56\,\text{mA}$$

The total power draw P_{total} for my custom board, using the applied 5V supply, is calculated as:

$$P_{\text{total}} = 5V \times 24.56\,\text{mA} = 122.8\,\text{mW}$$

Comparison with Commercial Arduino: For the commercial Arduino, with a voltage drop of 2.5V across each LED-resistor pair, the current calculation is as follows:

$$I = \frac{2.5V}{220\,\Omega} = 11.36\,\text{mA}$$

With the 50% duty cycle, the effective current for each LED-resistor pair is:

$$I_{\text{effective}} = 11.36\,\text{mA} \times 0.5 = 5.68\,\text{mA}$$

Thus, the total effective current for the commercial Arduino is:

$$I_{\text{total}} = 4 \times 5.68\,\text{mA} = 22.72\,\text{mA}$$

The total power draw for the commercial Arduino is therefore:

$$P_{\text{total}} = 5V \times 22.72\,\text{mA} = 113.6\,\text{mW}$$

Summary: The total power draw for my custom board was measured at approximately **122.8 mW**, while the commercial Arduino demonstrated a lower power draw of **113.6 mW**. The slightly higher power consumption on my board is consistent with the increased voltage drop across each LED-resistor pair (2.7V vs. 2.5V) and may be influenced by differences in component tolerances or layout. These measurements underscore the impact of layout and component selection on overall power efficiency. But again my board is very similar to that of a commercial board.

Near Field Emission

The near-field emission test was conducted to measure electromagnetic interference (EMI) levels on both my custom board and the commercial Arduino. This test is essential for understanding how each board manages electromagnetic emissions, which can affect nearby components and signal integrity.

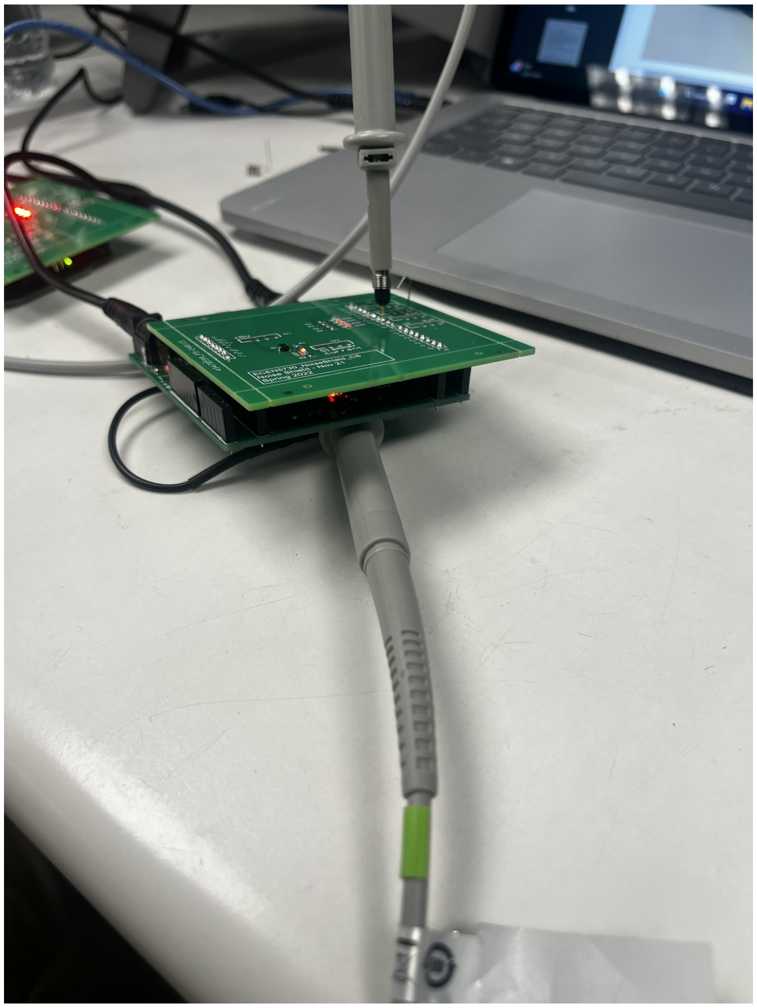
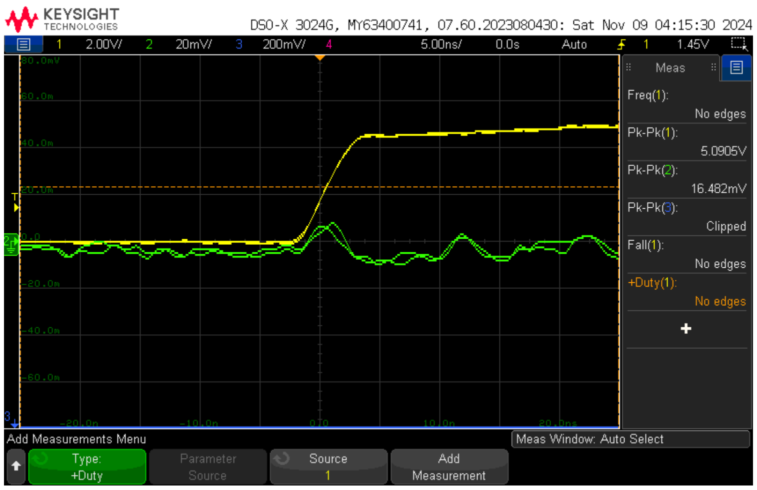


Figure 24: Testing Setup for Near Field Emission Measurement

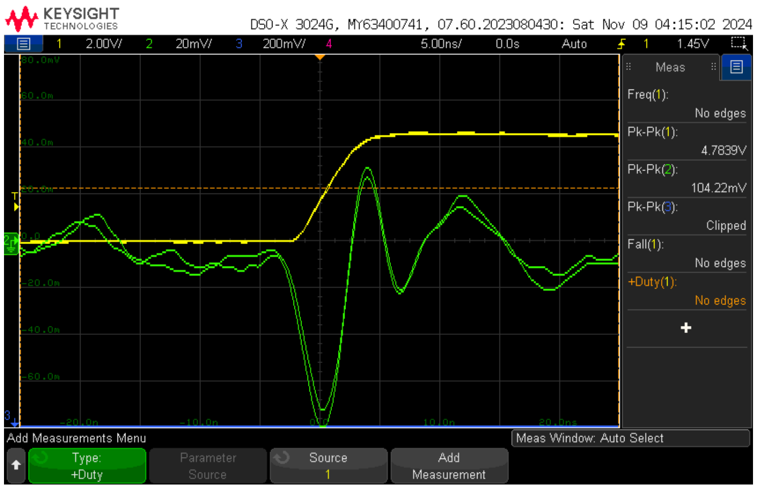
In the figure above, the testing setup for near-field emission measurement is shown. Here, **Channel 2** of the oscilloscope was connected to a probe placed under the board, capturing near-field emissions emanating from the board's ground plane. **Channel 1** was used to measure the switching signal at digital pin 13, allowing us to correlate the emissions with the switching activity on the board.



```
DSO-X 3024G, MY63400741, 07.60.2023080430: Sat Nov 09 04:15:30 2024
1  2.00V/   2  20mV/   3  200mV/   4      5.00ns/   0.0s   Auto   1   1.45V
Meas
Freq(1):      No edges
Pk-Pk(1):     5.0905V
Pk-Pk(2):     16.482mV
Pk-Pk(3):     Clipped
Fall(1):      No edges
+Duty(1):     No edges
Add Measurements Menu                Meas Window: Auto Select
Type: +Duty   Parameter Source   Source 1   Add Measurement
```

Figure 25: Near Field Emission on My Board

In the figure above, my custom board shows near-field emissions measuring approximately **16 mV**. This low level of EMI suggests effective EMI reduction strategies were implemented in the layout design, likely benefiting from optimizations in ground plane continuity and component placement, which help confine emissions and minimize radiation.



```
DSO-X 3024G, MY63400741, 07.60.2023080430: Sat Nov 09 04:15:02 2024
1  2.00V/   2  20mV/   3  200mV/   4      5.00ns/   0.0s   Auto   1   1.45V
Meas
Freq(1):      No edges
Pk-Pk(1):     4.7839V
Pk-Pk(2):     104.22mV
Pk-Pk(3):     Clipped
Fall(1):      No edges
+Duty(1):     No edges
Add Measurements Menu                Meas Window: Auto Select
Type: +Duty   Parameter Source   Source 1   Add Measurement
```

Figure 26: Near Field Emission on Commercial Arduino

In contrast, the figure above shows the commercial Arduino's near-field emission measurement, with a significantly higher EMI level of approximately **104 mV**. This higher level of emission may be due to differences in layout, grounding techniques, or less effective isolation of high-frequency switching noise, resulting in greater radiation from the board.

Factors Contributing to EMI Control on My Board: Several design practices likely contributed to the reduced EMI observed on my custom board:

- **Optimized Ground Plane Layout**: A continuous, low-impedance ground plane minimizes return path inductance and contains high-frequency switching currents more effectively. This layout design helps reduce EMI by confining switching noise to specific areas of the board.
- **Strategic Component Placement and Routing**: Careful placement of components, particularly those associated with high-speed switching, reduces the potential for radiated emissions. Routing sensitive traces away from high-frequency circuits also minimizes cross-coupling of noise.

- **Use of Decoupling Capacitors**: Properly placed decoupling capacitors close to high-speed components help filter out transient currents, limiting their spread across the board and reducing overall emissions.

Inrush Current

Inrush current refers to the initial surge of current drawn by a circuit when power is first applied. This transient current spike occurs as capacitors charge and components begin to power up, often resulting in a brief but significant increase in current draw compared to steady-state operation. Controlling inrush current is essential to prevent stress on power supplies and to avoid voltage dips that may affect other components on the board.

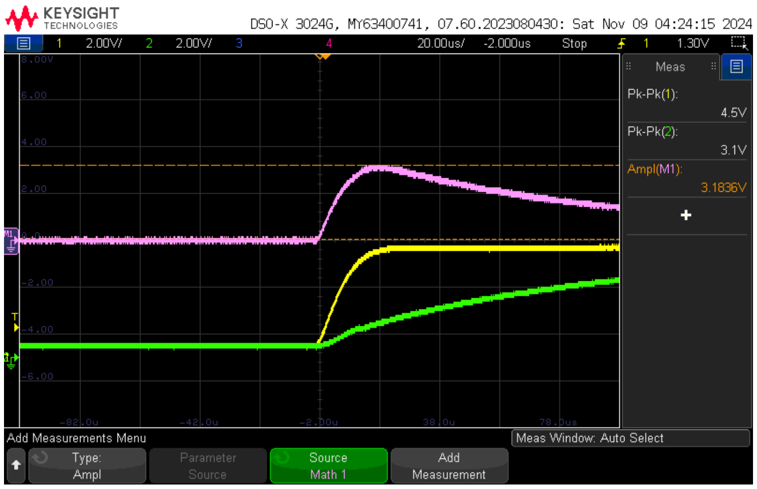


Figure 27: Inrush Current Measurement on My Board

The figure above shows the inrush current measurement taken on my custom board. Due to the limitations of the commercial Arduino's configuration, this measurement was only possible on my board. The oscilloscope captured a peak voltage of approximately **3.18V** across the series measurement resistor during the initial power-on phase.

Inrush Current Calculation: To calculate the inrush current, we need to know the value of the resistor used in the measurement. Assuming a measurement resistor of $R = 10\,\Omega$, the inrush current I_{inrush} can be calculated using Ohm's law:

$$I_{\text{inrush}} = \frac{V}{R} = \frac{3.18V}{10\,\Omega} = 0.318\,\text{A}\,(318\,\text{mA})$$

This calculated inrush current of **318 mA** represents the peak current drawn by the board at the moment of power-up. This transient current quickly stabilizes as capacitors charge and the board reaches a steady state.

Design and Assembly Reflections

What Worked Well?

Several aspects of the Golden Arduino design and assembly were successful:

- **Component Soldering**: Attaching the ATmega328 microcontroller was challenging, as it was difficult to avoid creating shorts between the pins. I had to use a heat gun for the reflow process to achieve stable connections. Although I accidentally tore off one of the pins on the ATmega chip during this process and had to replace it with a new chip, the final soldering was successful and resulted in reliable connectivity.
- **Noise Control on Near-Field Emissions**: My board's near-field emissions were significantly lower than those of the commercial Arduino, indicating effective EMI mitigation in the layout. This improvement highlights the success of the ground plane design and careful component placement, which reduced radiated emissions.
- **Consistent Operation After Initial Bootloading**: Once the ATmega328 was successfully bootloaded, the board consistently allowed for code uploads without needing additional adjustments. This stability demonstrates reliable integration between the microcontroller and the USB interface.

Hard and Soft Errors

Soft Errors:

- **Oscillator Shielding Issue**: The oscillators on the board are shielded by grounded metal enclosures. During assembly, I accidentally shorted one of the oscillator pins to ground, which caused issues when attempting to bootload the ATmega chip. Correcting this resolved the bootloading error.
- **Manual Reset Required for Bootloading and Initial Code Upload**: To bootload the chip, I had to manually press the reset button, which was also required during the first code upload. This behavior was unexpected, and while the reason is unclear, it did not impact subsequent code uploads. The need for manual intervention initially, but not subsequently, suggests a possible issue with the initial synchronization between the microcontroller and the USB interface.
- **ICSP Header Omission**: I forgot to include In-Circuit Serial Programming (ICSP) headers on my board, so I had to use individual pins rather than a convenient header for bootloading. This oversight made the initial setup more cumbersome but did not affect the board's functionality after bootloading.
- **Inconsistent Noise Performance Compared to the Commercial Arduino**: My board demonstrated varied noise performance when compared to the commercial Arduino. In some tests, my board showed improved noise levels, while in others, the commercial board performed better. These fluctuations indicate areas for potential optimization in noise control across different signal paths.
- **Quiet High Signal Noise Fluctuations**: Occasional fluctuations in the quiet high signal noise were observed, though they did not consistently impact performance. This variability suggests the potential for further refinement in grounding and decoupling to stabilize noise levels across all conditions.

Takeaways

The Golden Arduino custom board demonstrated satisfactory performance overall, with notable noise and EMI improvements over the commercial Arduino in specific areas. The assembly process provided valuable insights, particularly in managing small components like the ATmega328, which required careful reflow soldering to avoid shorts. The need for a heat gun and the replacement of a damaged ATmega chip underscore the importance of precise handling in assembly.

Although the board performed well in many areas, the inconsistency in noise reduction compared to the commercial Arduino highlights opportunities for further optimization, especially in the slammer circuit and quiet high signal paths. Future improvements could include adding ICSP headers for easier bootloading, refining the layout to reduce EMI even further, and investigating the initial reset requirement during bootloading. Overall, the project successfully demonstrated the design and assembly of a reliable custom Arduino, with practical lessons that will enhance future PCB designs.

Conclusion

Board 3, the Golden Arduino, successfully achieved the primary goals outlined in the Plan of Record. The board performed comparably to a commercial Arduino with minor variations in noise levels and rail stability. Lessons learned, especially in EMI reduction, will inform future design optimizations.