ECEN 4730-5730-001 **Practical PCB Design Manufacture** **Fall 2023**

Board4 Report

Shrinithi Venkatesan

Aim:

The project targets a standalone 4-layer instrument droid for graduate students, featuring a 328 microcontroller and data acquisition system. It prioritizes non-compatibility with Arduino Uno R3 but ensures access to essential pins for bootloading. Emphasizing 4-layer design proficiency, the instrument measures and characterizes the output impedance of diverse voltage sources, including dynamic Thevenin resistance. Underpinned by insights from Lab 21, the solderless breadboard version, the process involves meticulous assembly, bootloading, and rigorous testing. Thorough documentation focuses on design, assembly, testing, and code development, enhancing the graduate educational experience in board design and microcontroller programming.

POR:

1. Incorporate a power plug for an external 5V AC to DC charger to power the board.
2. Implement a voltage regulator to convert 5V to 3.3V for efficient operation of components.
3. Include USB mini support for programming and powering the board.
4. Integrate a power selector switch to choose between USB and the 5V external power supply.
5. Utilize a Low Drop-Out (LDO) regulator to convert 5V to stable 3.3V.
6. Install indicator LEDs for visual confirmation of 5V and 3.3V power status.
7. Implement a power isolator switch to disconnect power to the microcontroller and CH340 chip.
8. Include a reset switch to initiate a reset of the microcontroller in case of undesired behavior.
9. Ensure In-Circuit Serial Programming (ICSP) compatibility for bootloading and programming the microcontroller.
10. Develop functionality to measure in-rush current during board startup.
11. Add Transient Voltage Suppression (TVS) chip to protect against static voltage damage.
12. Provide extra headers for convenient ground connections for sensors or other peripherals.
13. Plan test points for critical signals:
    5V input
    3.3V input
    In-rush current measurement
    USB D+ and D-
    USB VBUS
    Microcontroller RX and TX
    Microcontroller SDA and SCL
14. Place decoupling capacitors in proximity to all ICs to reduce switching noise.

15. Implement a copper poured ground plane for effective signal return and noise reduction.
16. Implement an ADC to accurately measure the differential voltage between the sense resistor and the VRM.
17. Utilize the DAC output to drive the MOSFET through an Op-Amp, establishing a current source system.
18. Establish connections for smart LEDs and a buzzer to the digital pins of the ATmega328.

Risk Reduction Steps:

1. Thoroughly research and validate the chosen components for reliability and compatibility.
2. Prototype and simulate critical sections of the board design to identify and mitigate potential issues.
3. Perform a comprehensive analysis of the board layout to ensure proper signal integrity.
4. Conduct in-depth testing of each feature to validate its functionality and performance.
5. Implement feedback mechanisms to monitor and address potential problems during the testing phase.
6. Standardizing signal lines to 6 mils width and power lines to 20 mils width is a sound practice. This design decision simplifies the detection of signal and power nets, reducing the risk of misconnection or interference.
7. Keeping board size under 3.9” to ensure it sticks to standards.
8. Adhering to established best practices by using as many ground VIAs to ensure there’s no noise due to return currents in the ground plane. Placing decoupling capacitors closer to IC.
9. By placing test points and appropriately labeling them, you mitigate the risk of confusion during testing and debugging. This practice ensures that you can easily access critical points for measurement and verification.
10. Using switches at each stage for isolation and debugging purposes is a proactive measure to minimize risks. These switches allow you to control and isolate specific sections of the circuit, making it easier to identify and address issues without affecting the entire system.

Demonstration Goals:

1. Bootload the Atmega 328 to turn it into an Arduino.
2. Run the Arduino IDE and execute a standard sketch on the board.
3. Maintain compatibility with standard header pin footprints for shields.
4. Achieve noise levels 20% to 50% of a commercial board under identical conditions.
5. Ensure near-field emissions are less than 10% of a commercial Arduino board.
6. Optionally, add test points for digital buses to observe actual bus traffic.
7. Optionally, measure in-rush current and steady-state current with a series resistor for current sensing.

Expectations of what it is to “work”:

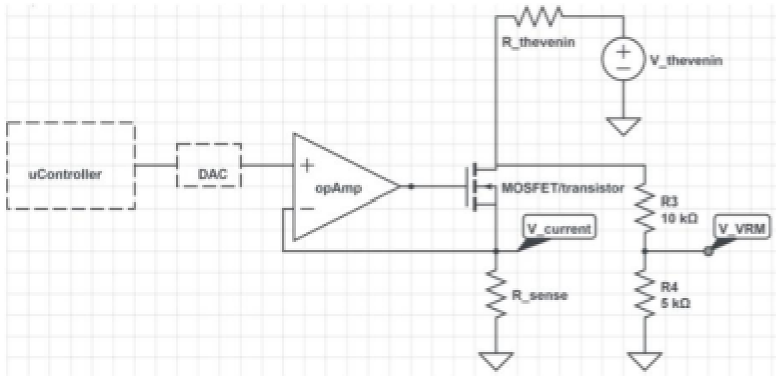
1. The board should reliably power up from both USB and a 5V external supply, ensuring consistent and stable operation.

2. The power indicator LED must promptly illuminate upon the application of power, providing a clear visual cue.
3. The voltage regulator should consistently and efficiently convert the 5V input to a stable and accurate 3.3V output.
4. The board should accurately measure the current flowing through one of the LEDs, providing a reliable metric for analysis.
5. SPI communication must function reliably to facilitate successful bootloading processes.
6. The board, configured as a Golden Arduino, should seamlessly execute any standard Arduino sketch, ensuring broad compatibility.
7. Indicator switches and lights should respond promptly to user inputs, enhancing the user interface and feedback mechanisms.
8. The reset signal should effectively reset the ATmega328, contributing to the stability of the microcontroller.
9. Bootloading processes should be smooth and compatible with various Arduino programmers, ensuring versatility in development environments.
10. Inrush current measurement, conducted using the scope, should provide accurate and meaningful results, allowing for comprehensive analysis.
11. The Low Drop-Out (LDO) regulator should consistently provide a steady 3.3V output without oscillations or instability.
12. Implemented design measures should result in a noticeable reduction in near-field emissions, contributing to electromagnetic compatibility.
13. Observing RX and TX waveforms during code loading should be clear and consistent, allowing for effective monitoring and analysis.
14. The board should demonstrate noise levels 20% to 50% lower than those observed in commercial Arduino boards, showcasing effective noise reduction measures.
15. The DAC ensures the accurate generation of voltage and waveform in accordance with the programmed code.
16. The instrument droid is equipped to conduct measurements, utilizing any external power source connected to the board's venin voltage and resistance.
17. The reset signal performs as anticipated, effectively resetting the ATmega328 when triggered

Significant Parts used:

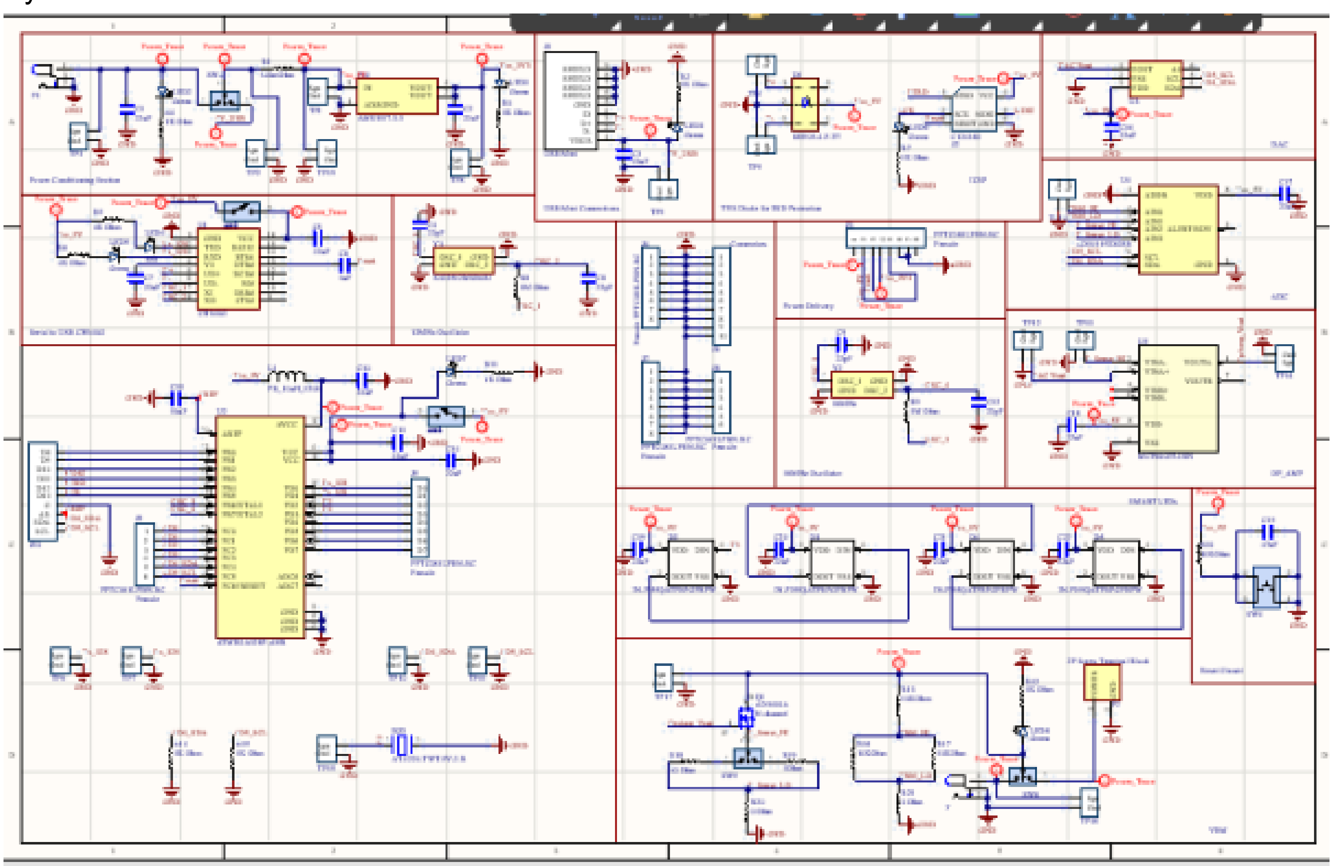
1. ATMEGA328- https://www.microchip.com/en-us/product/atmega328p
2. LDO - AMS1117 - http://www.advanced-monolithic.com/pdf/ds1117.pdf
3. CH340g - 74AHC14 -https://cdn.sparkfun.com/datasheets/Dev/Arduino/Other/CH340DS1.PDF
4. Crystal Oscillators - 16,12Mhz
5. DAC MCP4725 https://ww1.microchip.com/downloads/en/devicedoc/22039d.pdf
6. ADC ADS111x https://www.ti.com/lit/ds/symlink/ads1114.pdf
7. OP-AMP MCP6002 https://www.microchip.com/en-us/product/mcp6002

Napkin Sketch:

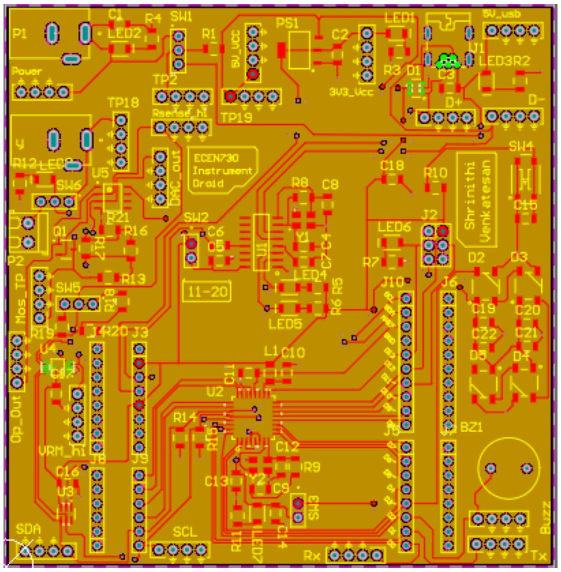


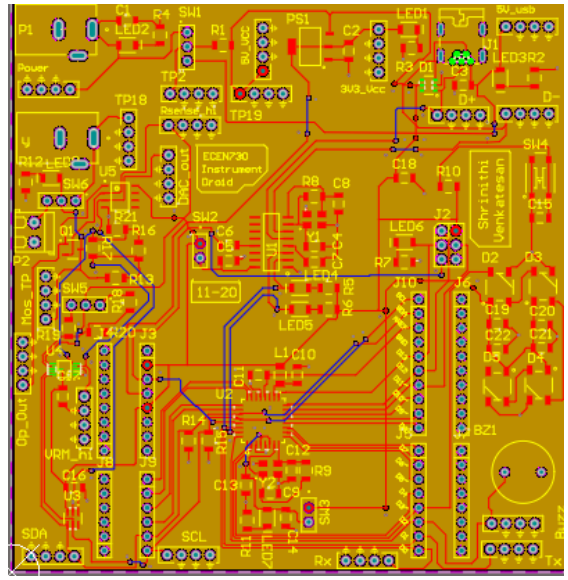
Working Schematic:

My Schematic:

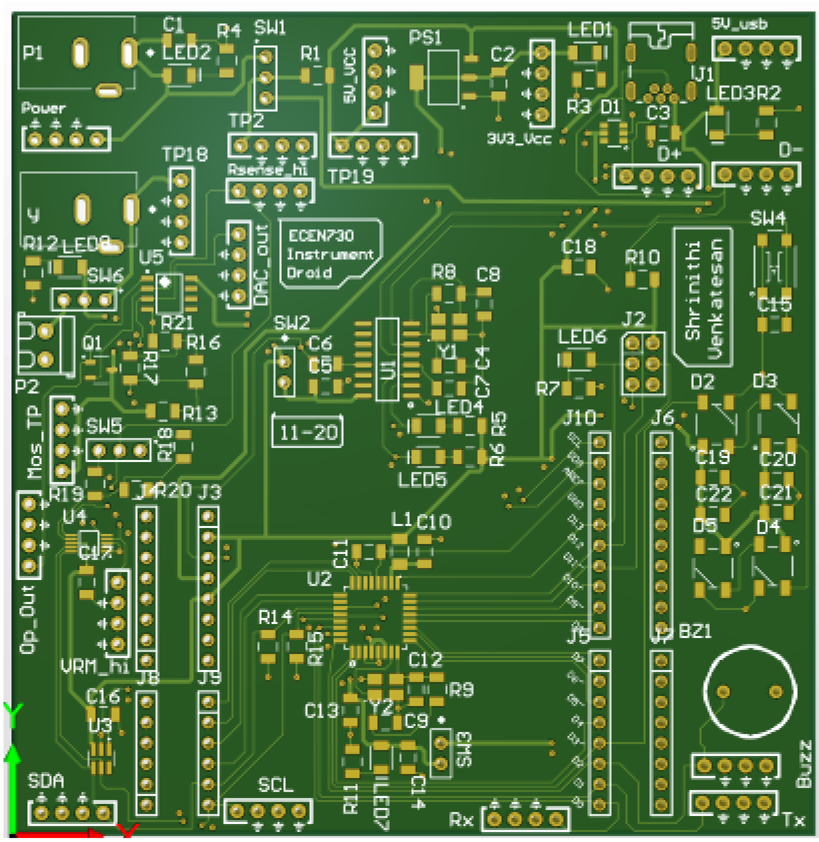


My Layout:
Top and bottom view:



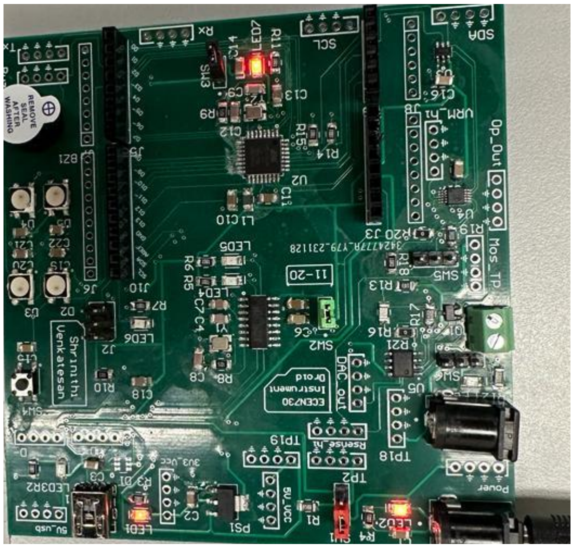


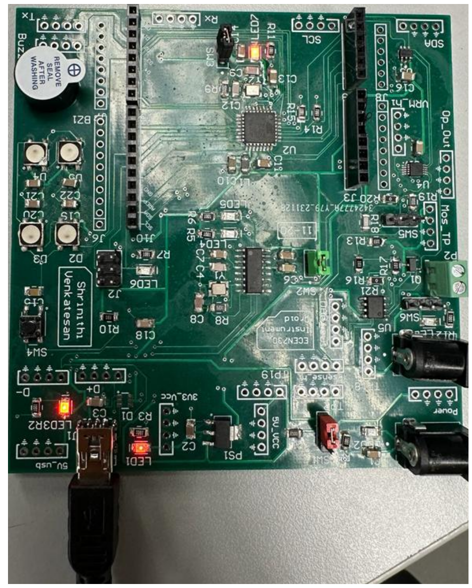
Bare board:



Assembled Board:

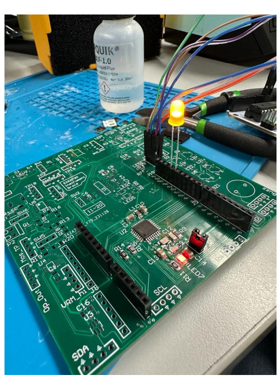
Power on design
Power plug and CH340





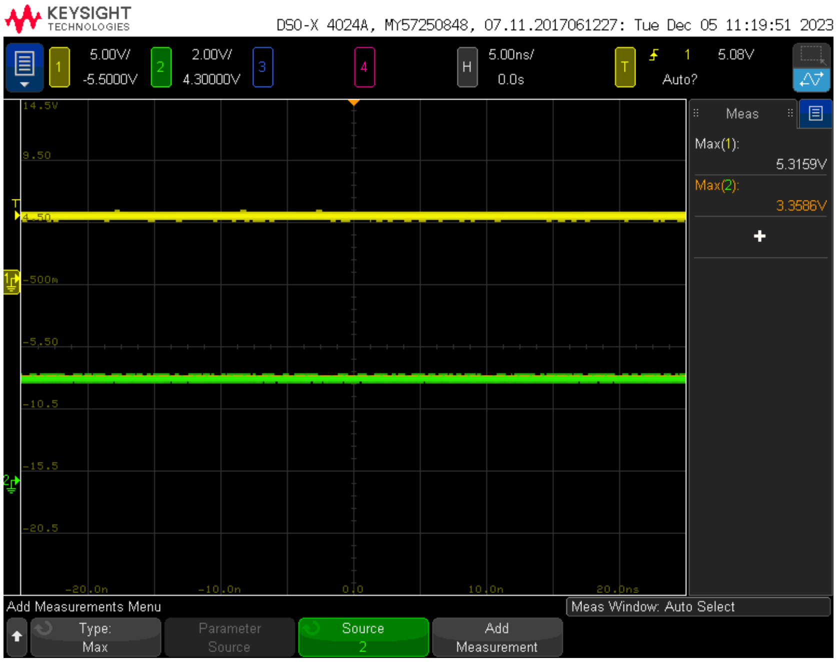
Bring-up Process Summary:

1. Conducted a short test on the bare board to verify the integrity of all connections and ensure proper functionality.
2. Placed all components, utilizing 1206 components, and ensured precise positioning before soldering.
3. Soldered the components with precision, followed by another round of short testing to identify and rectify any accidental shorts caused during soldering.Detected a short and tried to hard correct it by scratching out the trace, yet not successful.
4. 5V and LDO output was proper and verified.
5. Attempted to boot-load the controller using a commercial Arduino by just soldering the required components for that.
6. After that the blink code was dumped to verify communication with the controller working as expected.



Measurements and Analysis:

Power rail outputs:

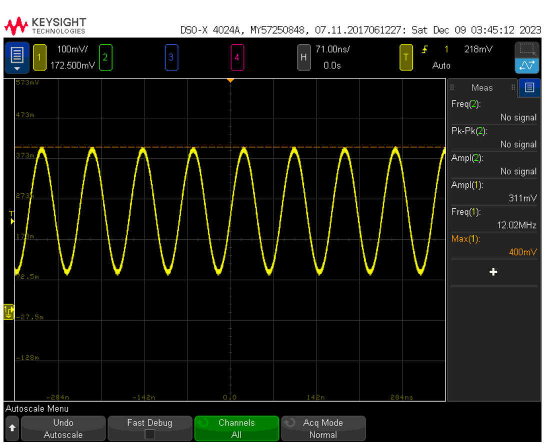


The observed voltages align closely with the expected values, with 5.3V and 3.3V measured at the 5V and 3.3V test points, respectively. The corresponding LEDs for both voltage rails have illuminated, indicating that the power supply and voltage regulation systems are functioning as intended.

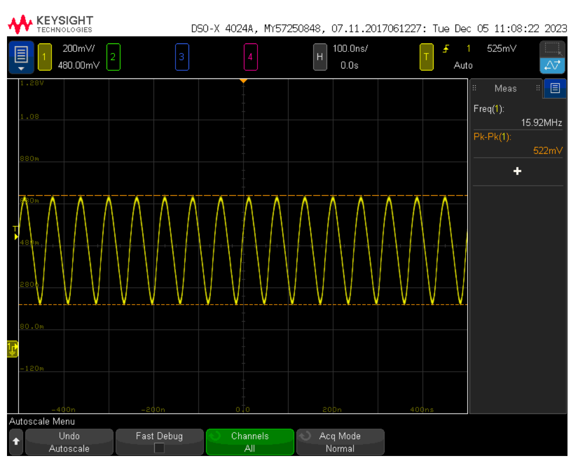
The oscilloscope channel readings further validate the stability of the voltage rails. Channel 1 (Yellow) displays a consistent 3.3V, confirming the proper operation of the 3.3V rail. Similarly, Channel 2 (Green) shows a steady 5V, validating the performance of the 5V rail.

These results demonstrate the successful power supply and voltage regulation aspects of the board, providing a solid foundation for further testing and functionality verification.

Oscillator Outputs:



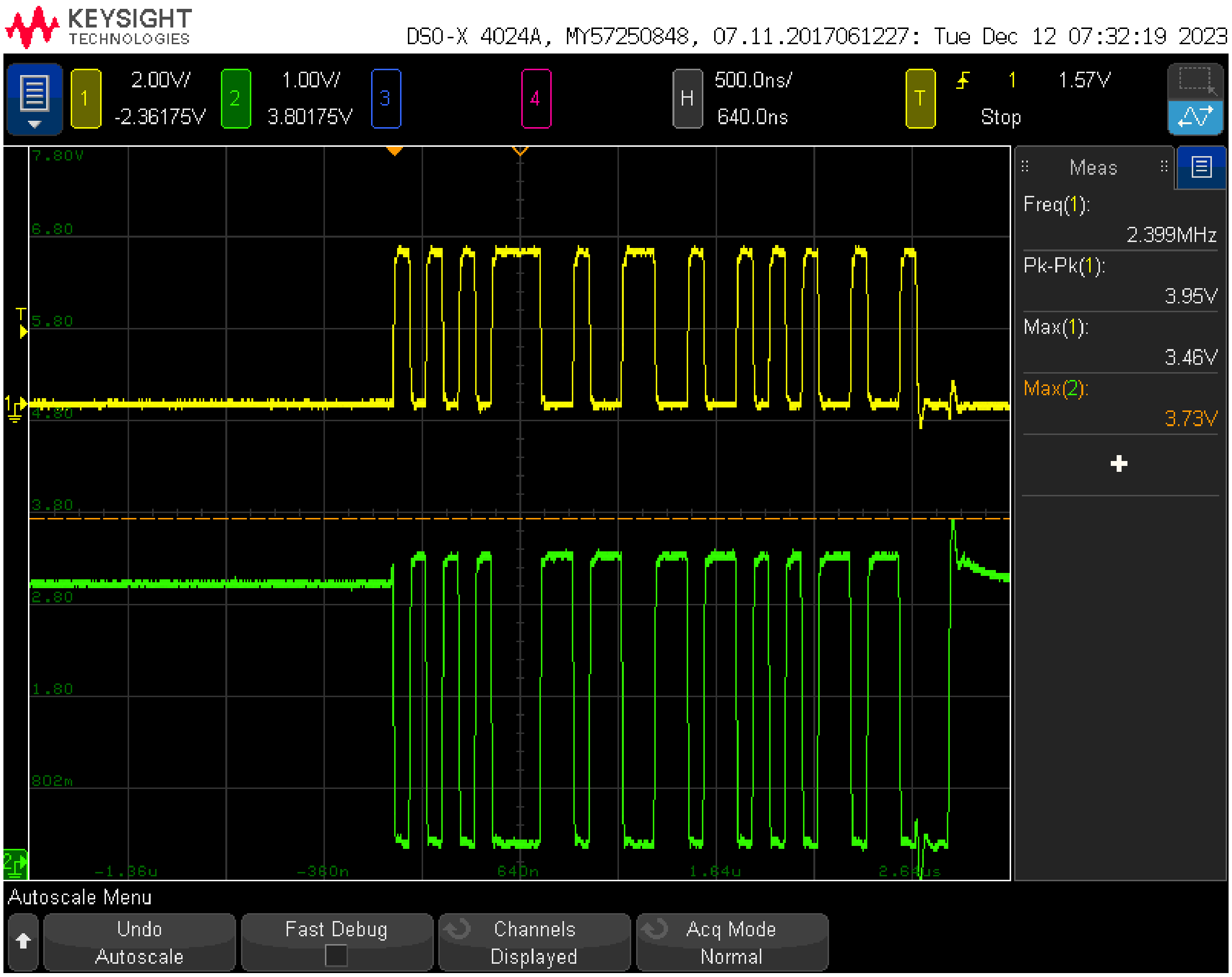
12Mhz from Y1



16Mhz from Y1

The presence of a 16MHz clock signal for the microcontroller indicates that the ATmega328 or a similar microcontroller is operating at the expected frequency. The presence of a 12MHz clock signal for the CH340G indicates that the USB-to-serial converter is functioning with the expected clock frequency.

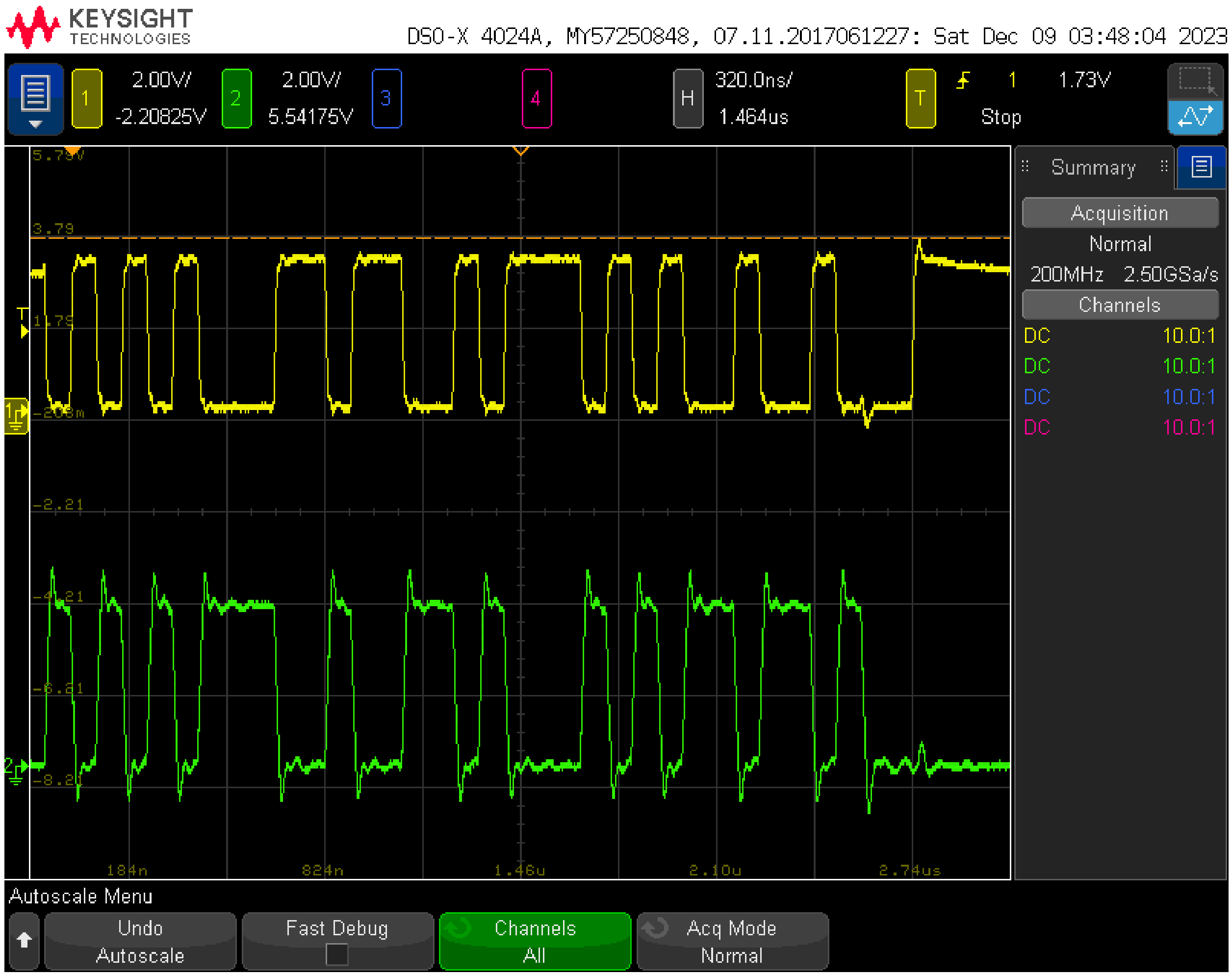
Data Signals: D+ and D-:



The presence of both the D+ (yellow trace) and D- (green trace) signals, specifically when the board is powered by a mini-USB cable and a USB port is detected, verifies the correct connection and functionality of the CH340G chip. These differential signals carry the data and inverted data signals, and their successful observation under the specified conditions indicates the proper operation of the USB communication interface.

The detection of a USB port serves as a positive confirmation that the D+ and D- signals are correctly connected. This validation is crucial for ensuring the reliable communication and functionality of the USB interface on the board.

Rx and Tx signals:

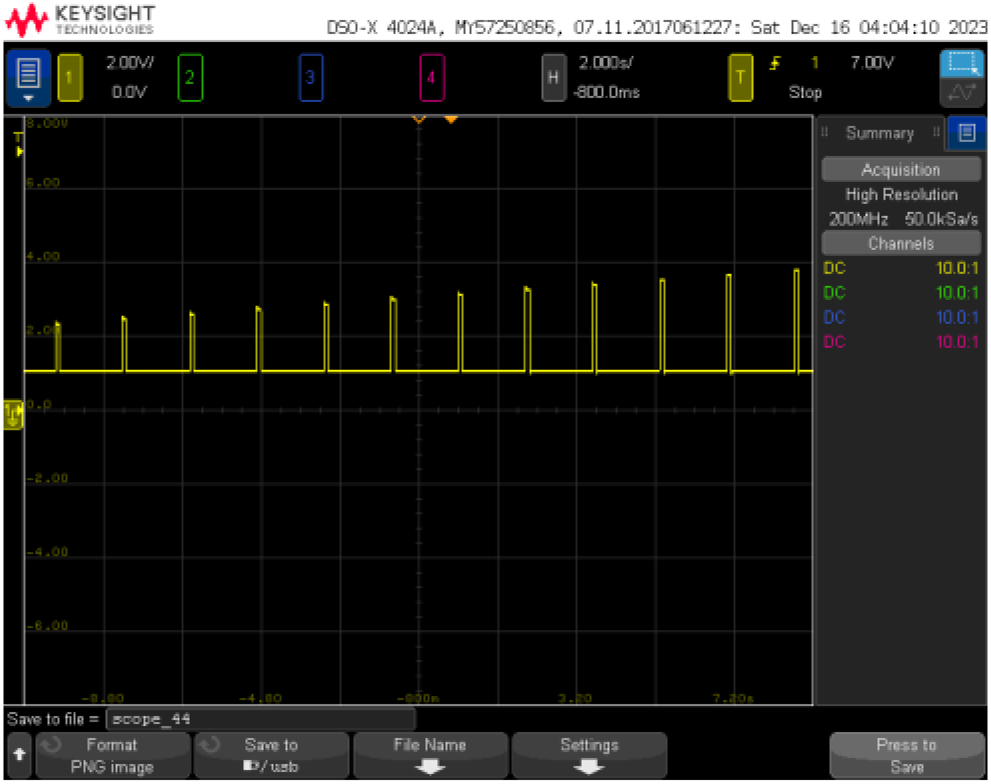


The observed UART Rx (yellow trace) and UART Tx (green trace) waveforms confirm the interaction between the ATmega328 microcontroller and the CH340G during the code uploading process through USB. The blinking of the Rx and Tx LEDs during code upload indicates the successful communication between the microcontroller and the USB interface.

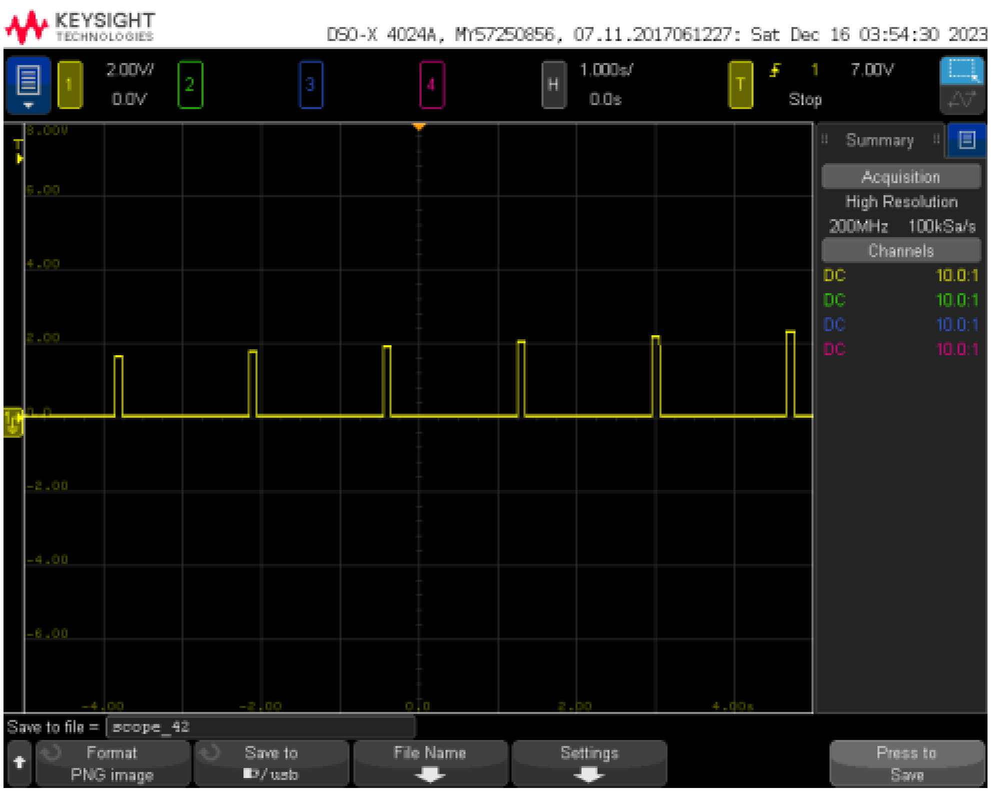
The triggering of the signals on the falling edge, as mentioned, aligns with UART communication standards where a 0 bit is used as the start bit. This behavior is consistent with proper UART communication protocols.

These results validate the functional interaction between the microcontroller and the USB interface, ensuring the successful upload of code to the ATmega328. The observed waveforms provide valuable insights into the communication dynamics and help verify the integrity of the UART interface on the board.

Op Amp Output:



DAC Output:



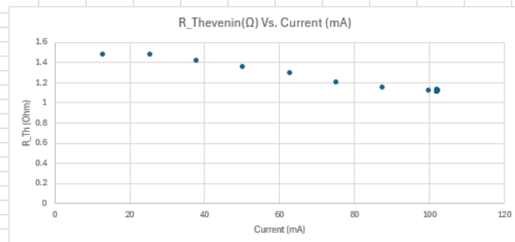
As per the input through Arduino code, we see that DAC output sweeps through the voltage between 0 - 5V with a division in duty cycle of 1.25mV per level.

Power source characteristics:
As a known source, we apply an input of 5V, 9V and 2A through the source and plot our respective outputs through the circuit. We observe expected outputs when using 10ohm sense resistor value.

9V 2A
Index, current in mA, V_thevenin, V_loaded, R_thevenin

Index	Current(mA)	V_Thevenin (V	V_Load (V)	R_Thevenin(Ω)
1	12.833	8.1352	8.1162	1.4841
2	25.215	8.1354	8.0981	1.4805
3	37.733	8.1354	8.0817	1.4214
4	50.071	8.1355	8.0672	1.3634
5	62.541	8.1352	8.0539	1.3003
6	74.908	8.1353	8.0446	1.2114
7	87.341	8.1356	8.0348	1.1539
8	99.73	8.1358	8.023	1.1306
9	101.742	8.1361	8.0226	1.115
10	101.775	8.1357	8.0218	1.1191
11	101.808	8.136	8.0212	1.1275
12	101.829	8.1363	8.0226	1.1165
13	101.866	8.1363	8.0217	1.1251
14	101.885	8.1361	8.0223	1.1169
15	101.926	8.1362	8.0216	1.1246
16	101.935	8.1365	8.0209	1.1339
17	101.962	8.1365	8.0222	1.1213
18	101.988	8.1358	8.021	1.1254
19	102.019	8.1367	8.0217	1.1267
20	102.04	8.1361	8.0215	1.1238

Current(mA	_Thevenin(Ω)
12.833	1.4841
25.215	1.4805
37.733	1.4214
50.071	1.3634
62.541	1.3003
74.908	1.2114
87.341	1.1539
99.73	1.1306
101.742	1.115
101.775	1.1191
101.808	1.1275
101.829	1.1165
101.866	1.1251
101.885	1.1169
101.926	1.1246
101.935	1.1339
101.962	1.1213
101.988	1.1254
102.019	1.1267
102.04	1.1238

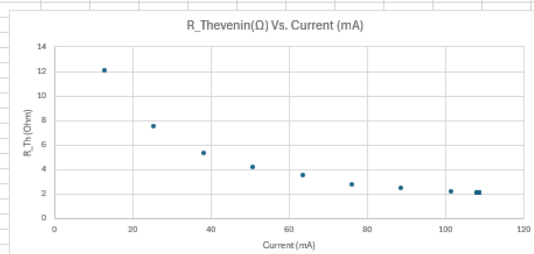


The yellow trace here is the Rth plot

5V 2A
Index, current in mA, V_thevenin, V_loaded, R_thevenin

Index	Current(mA)	V_Thevenin (V)	V_Load (V)	R_Thevenin(Ω)
1	13.089	4.4115	4.2544	12.0075
2	25.627	4.4112	4.2189	7.5034
3	38.383	4.412	4.2071	5.3381
4	50.856	4.4131	4.1986	4.2178
5	63.552	4.4133	4.1892	3.5255
6	76.177	4.414	4.2055	2.7369
7	88.778	4.4162	4.1977	2.4608
8	101.453	4.4163	4.1913	2.218
9	108.374	4.4168	4.1883	2.1091
10	108.661	4.417	4.1877	2.1101
11	108.272	4.4173	4.1878	2.12
12	108.338	4.4172	4.1881	2.1142
13	108.086	4.4173	4.1875	2.1267
14	108.698	4.4171	4.1879	2.1078
15	108.651	4.4172	4.1869	2.1198
16	108.183	4.4173	4.1879	2.1205
17	108.412	4.4174	4.188	2.1164
18	108.753	4.4177	4.1889	2.1035
19	108.521	4.4175	4.1879	2.1151
20	108.633	4.4176	4.1884	2.1098

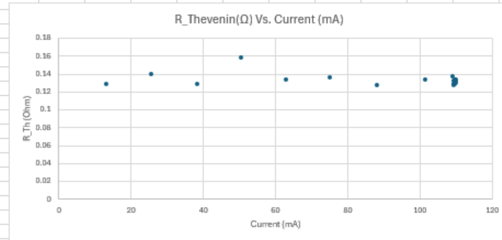
Current(mA	_Thevenin(Ω)
13.089	12.0075
25.627	7.5034
38.383	5.3381
50.856	4.2178
63.552	3.5255
76.177	2.7369
88.778	2.4608
101.453	2.218
108.374	2.1091
108.661	2.1101
108.272	2.12
108.338	2.1142
108.086	2.1267
108.698	2.1078
108.651	2.1198
108.183	2.1205
108.412	2.1164
108.753	2.1035
108.521	2.1151
108.633	2.1098



Here, I have compared the values between Current(blue) vs Rth(yellow)

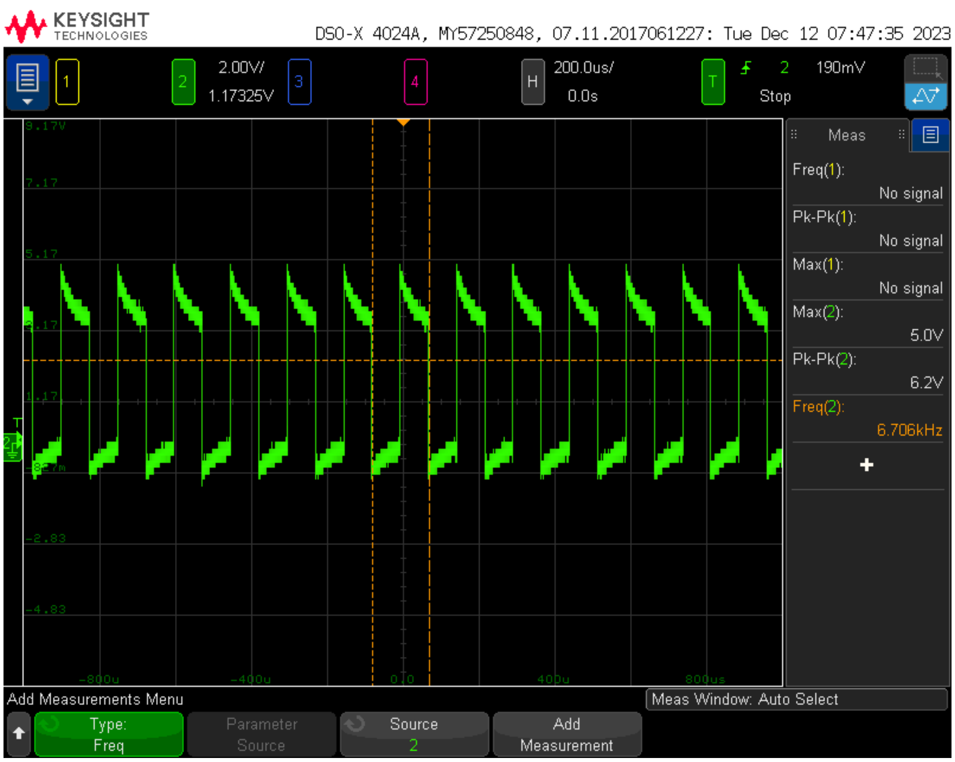
5V 1A (bench power gen)
Through the function generator we provide input of 5V 1A and observe the following output as expected values. We first experimented with unknown values of VRM using a variety of sense resistor values from 1 Ohm and finally used 10 Ohm to observe the following output.

Index	Current(mA)	V_Thevenin (V)	V_Load (V)	R_Thevenin(Ω)		Current(mA)	R_Thevenin(Ω)
1	13.104	4.7756	4.7739	0.1286		13.104	0.1286
2	25.734	4.7756	4.772	0.139		25.734	0.139
3	38.477	4.7756	4.7706	0.1285		38.477	0.1285
4	50.693	4.7757	4.7677	0.1576		50.693	0.1576
5	63.006	4.7754	4.7669	0.1336		63.006	0.1336
6	75.289	4.7756	4.7654	0.136		75.289	0.136
7	88.251	4.7756	4.7644	0.127		88.251	0.127
8	101.624	4.7757	4.7622	0.1333		101.624	0.1333
9	109.42	4.7756	4.7615	0.1289		109.42	0.1289
10	110.123	4.7756	4.7609	0.1328		110.123	0.1328
11	109.605	4.7755	4.7609	0.1327		109.605	0.1327
12	109.265	4.7757	4.7607	0.1368		109.265	0.1368
13	109.694	4.7755	4.7612	0.1306		109.694	0.1306
14	109.782	4.7756	4.7609	0.1336		109.782	0.1336
15	109.61	4.7755	4.7616	0.1268		109.61	0.1268
16	109.669	4.7756	4.7615	0.1285		109.669	0.1285
17	109.871	4.7755	4.7611	0.1306		109.871	0.1306
18	109.714	4.7756	4.7611	0.1323		109.714	0.1323
19	110.165	4.7756	4.7611	0.1324		110.165	0.1324
20	110.182	4.7755	4.7613	0.1294		110.182	0.1294



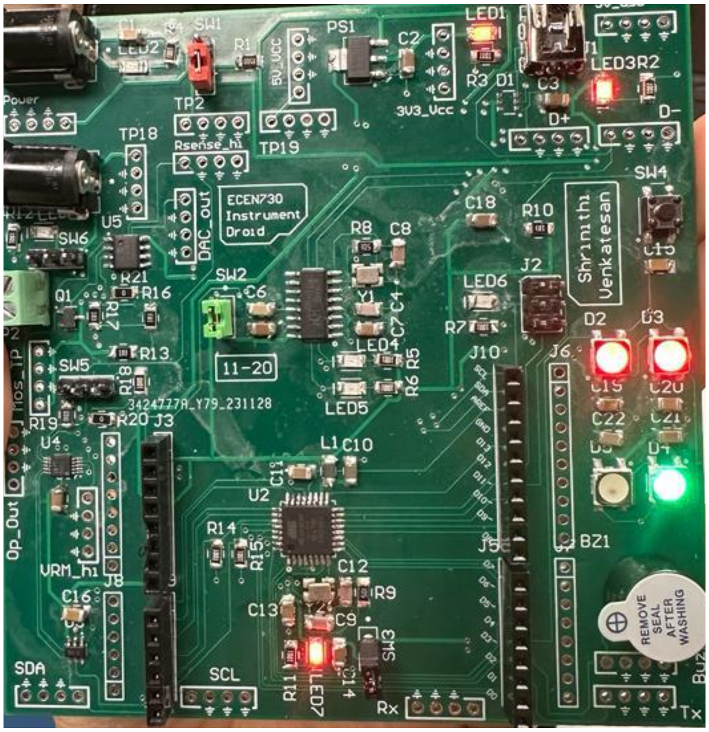
Here, I have compared the values between Current(blue) vs Rth(yellow)

Buzzer output:



When the arduino code was applied to make the buzzer ring which was connected to my pin2, I could see the ringing in oscilloscope output.

Smart LED out:



When arduino code was applied to smart leds connected to onboard pin3, I could make use of the example code to test and verify the working of my LEDs.

Analysis of my project:

THROUGH CORRECTING ERRORS FROM MY DESIGN OF LAST BOARD, I COULD INCORPORATE BETTER DESIGN AND MY BOARD 4 WORKED WITH VERIFYING MY BOARD 3 PARAMETERS AS WELL.

What worked?

1. The board reliably powers up seamlessly from both USB and a 5V external supply, ensuring consistent and stable operation.
2. The power indicator LED promptly illuminates upon the application of power, providing a clear visual cue for successful power activation.
3. The voltage regulator consistently and efficiently converts the 5V input to a stable and accurate 3.3V output, ensuring reliable power regulation.
4. Accurate measurement of the current flowing through one of the LEDs is achieved, providing a reliable metric for in-depth analysis.
5. SPI communication functions reliably, facilitating successful bootloading processes and ensuring smooth communication protocols.
6. Configured as an Arduino, the board seamlessly executes any standard Arduino sketch, demonstrating broad compatibility and versatility.
7. Indicator switches and lights respond promptly to user inputs, enhancing the user interface and providing effective feedback mechanisms.
8. The reset signal effectively resets the ATmega328, contributing to the stability and reliable operation of the microcontroller.

9. Bootloading processes are smooth and compatible with various Arduino programmers, ensuring flexibility in development environments.
10. Inrush current measurement, conducted using the scope, provides accurate and meaningful results, enabling comprehensive analysis of power dynamics.
11. Observing RX and TX waveforms during code loading is clear and consistent, allowing for effective monitoring and analysis of communication signals.
12. The CH340 USB-to-serial converter successfully established communication, detected the COM port.
13. Transmit and Receive LEDs indicated seamless communication activity.
14. The reset circuit reliably initializes the system, contributing to stability.
15. Analog-to-Digital and Digital-to-Analog converters provided precise signal conversion.
16. The Op Amp and Mosfet combinational circuit functioned effectively.
17. Thermal resistance (Rth) was accurately calculated using the Voltage Regulator Module (VRM).
18. The LEDs and BUzzer worked as per the expected code functionality.

What did not work and what to correct for the future?

1. Fortunately by correcting my errors of carefully placing the oscillators and other components, I was able to get my board working at first shot this time.
2. I had, although missed, placed the decoupling capacitor of one component a little far from the IC, but it didn't bother the circuit as the component did not involve in measurements.
3. Initially, I had no reading through DAC and I had replaced the component to make it work and I anticipate it might be the component issue. Also, I had to figure out the address of the DAC carefully by trial and error, which took a little anticipation to know what is expected. Rule 9 helped here as we already did a breadboard version of this board successfully

Learning Outcomes:

1. Develop the circuit design by thoroughly understanding and incorporating insights from available references, adhering to best design practices. Prioritize a meticulous layout to optimize performance. Perform Clock and Data Recovery (CDR) analysis before proceeding to board fabrication to ensure precision and reliability.
2. Employ strategic routing techniques and grounding practices to mitigate potential signal integrity issues.
3. Implement best routing techniques and measures to ensure I get better noise resistance compared to this design.
4. Understand the importance of components, like where to use ferrite beads over power rails and when not to. Always, a main rule, anticipate the expected outcome.

In conclusion, by implementing these best practices and maintaining a systematic approach to circuit design and testing, you can enhance the reliability and success of your projects. Learning from past experiences, both successes and challenges, will contribute to my further best implementations.