Arduino UNO

Objective:

The primary objective of this laboratory is to create an Arduino board with substantially reduced switching noise and crosstalk compared to commercially available Arduino boards. The board should be designed with the utmost care to minimize near-field emissions and noise. It should be able to receive code uploads via a USB connection, work with the Arduino IDE, and be fully compatible with most Arduino Uno R3 shields. To achieve this, we will begin with the essential functions required.

The following steps were taken to execute this project:

1. Start by manually sketching the circuit schematic on paper or a napkin.
2. Gain an understanding of datasheets, particularly the electrical characteristics.
3. Use Altium to design schematics and layout plans.
4. Send the board for manufacturing and conduct Critical Design Review (CDR).

Plan of Action:

Key features of this board include:

1. The addition of a switch at the input to select power from either the power jack or USB, but not both simultaneously.
2. Voltage conversion from 5 volts to 3.3 volts using an LDO.
3. The inclusion of an indicator LED near the input power supply.
4. The use of a 12 MHz clock for the CH340G.5. Addition of ICSP Pins for the initial boot loading of the ATmega328P.
6. Creation of a ground plane by copper pouring on the PCB's bottom layer.
7. Utilization of a ferrite filter on the ADC's AVCC pin due to its sensitivity to noise.
8. Integration of a TVS diode on the computer's USB port for ESD protection.
13. The use of the CH340G for USB UART communication.

To reduce noise:

1. Place decoupling capacitors close to the ICs to minimize noise.
2. Implement a common ground plane with no shared return paths.
3. Minimize the use of cross-under and keep them short for noise reduction.

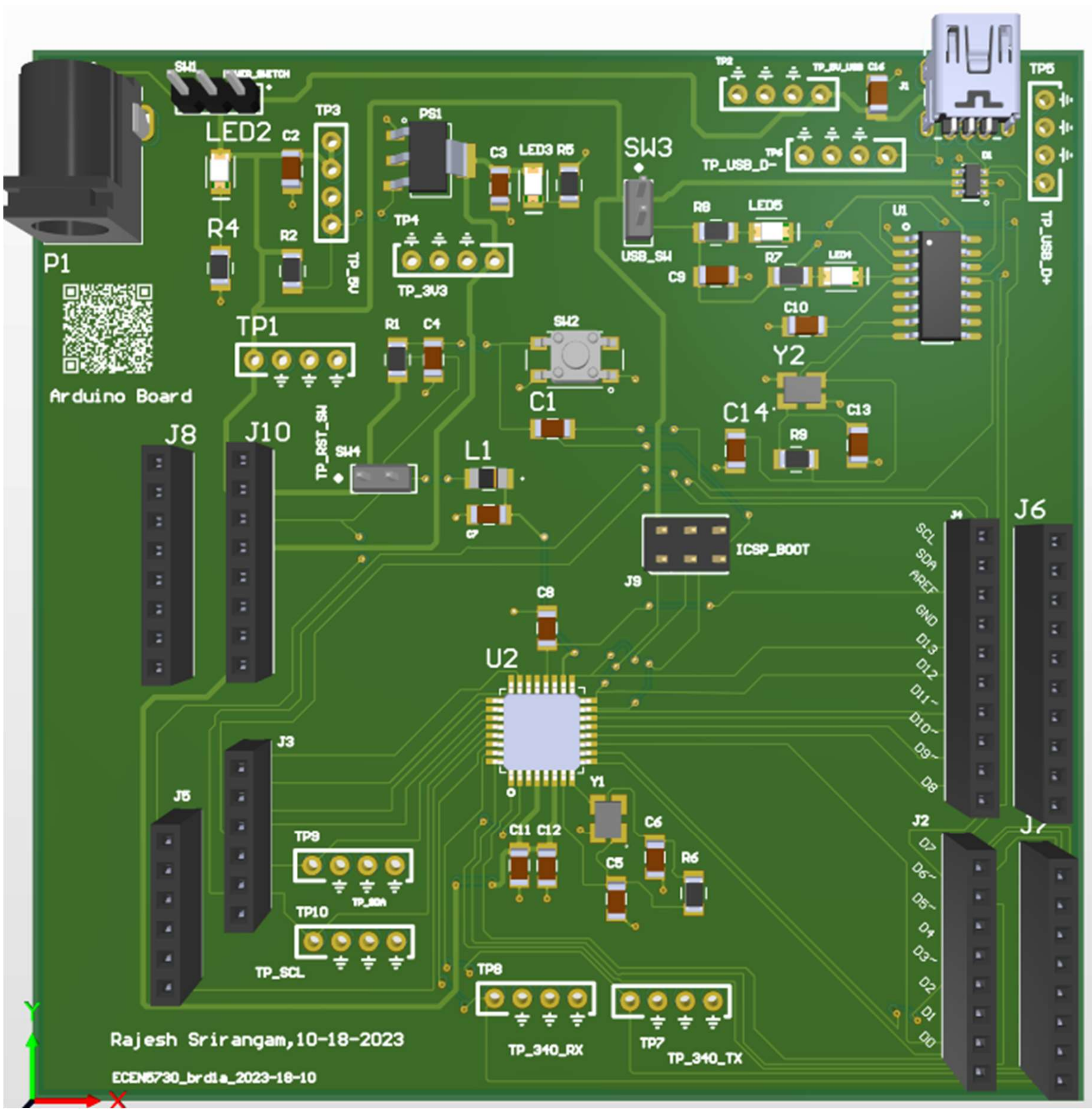
To mitigate risks:

1. Adopt a consistent naming convention for all components on the board.
2. Include test points for inrush current measurement, D+, D-, TX, and RX.
3. Utilize a filter capacitor at the output of the LDO to reduce voltage output oscillations.
4. Use switches where necessary to control power to the board.
5. Position a 16 MHz clock close to the ATmega328P and a 12 MHz clock near the CH340G.
6. Ensure that power lines are 20 mils wide, while signal lines are 6 mils wide, as per the fab vendor's guidelines.
7. Keep the routing of RX and TX lines roughly the same length to avoid delays.
8. Incorporate indicators for easy debugging.
9. Use test points for troubleshooting and board validation.

What it means for the board to work:

1. The board should be powered at 5V, either from USB or a 5V supply.
2. An LED near the 5V and 3.3V should light up when powered.
3. The voltage regulator should successfully convert 5V to 3.3V.
4. Current flow through one of the LEDs should be measurable.
5. SPI communication should function for successful bootloading.
6. Any sketch that runs on a commercial Arduino should also run on this board, as with a Golden Arduino.
7. The board should have indicator switches and lights.
8. The ATmega328 should be reset using the reset signal.
9. Any Arduino board with an Arduino programmer acting as the ISP can be used for bootloading.
10. The scope should be utilized to measure the inrush current capability.

PCB Board:



Near Field Emission (NFE) Measurement:

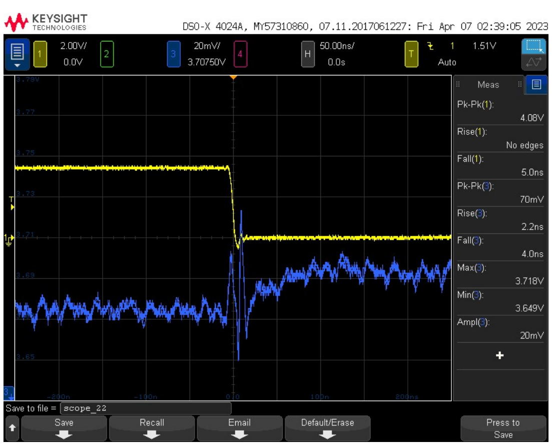
An antenna was created utilizing the scope probes and kept on the underside of the boards as a means of testing NFE. Near field emissions on the Arduino, the blue trace indicates NFE on the board's bottom side, while the yellow trace indicates the scope trigger. (D13). The NFE amplitude on a commercial Arduino is around 60 mV, while it is around 10 mV on a golden Arduino.

NFE- Arduino

NFE- Arduino

Key Observation: Arduino has substantially lower Near Field Emission than a commercial Arduino. This is a significant advancement since Near Field Emission creates Far Field Emission, which causes EMC difficulties.

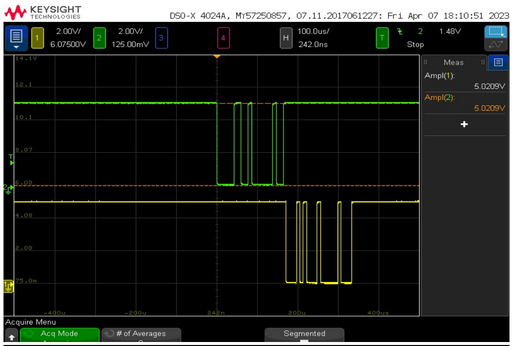
5V Power Rail Switching Noise Measurement:



Key Observation:

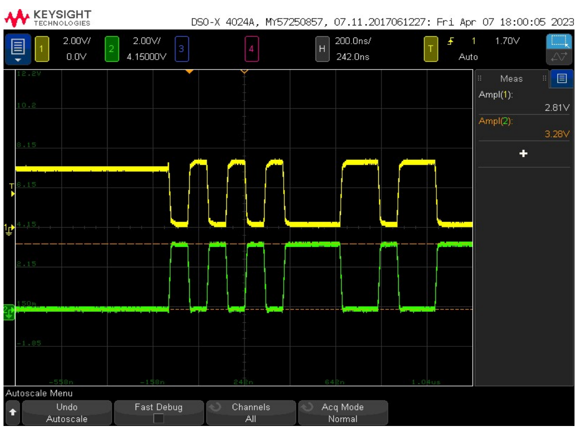
Switching noise on the Commercial or Arduino's falling edge 5V power rail is shown by blue trace, while the yellow trace is the scope trigger slammer output. The amplitude of the noise on the Arduino is 20mV.

Rx-Tx Signal Measurements:



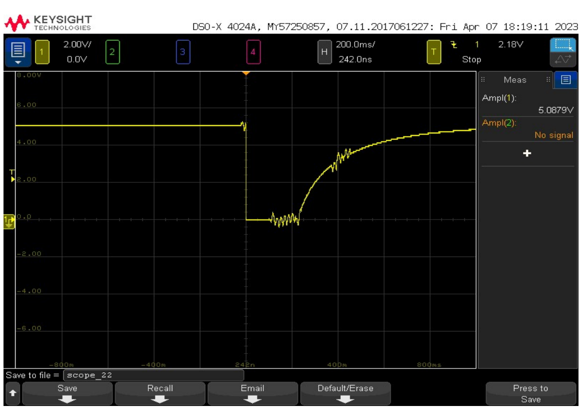
As shown in the above picture, Rx and Tx signal measurements are seen. Green trace represents the Rx signal whereas yellow trace represents the Tx signal.

D+ and D- Measurements:



Key Observation: If we observe the above waveform, we can see that the D+ and D- have opposite polarities.

Reset Pin Measurement:



Key Observations: The reset signal is always high. When we push the Reset button, the voltage drops briefly before returning to normal.

What Worked:

1. Both the power jack and USB produced a 5V output from an AC-DC charger.
2. The LED next to the power source illuminated when the board was powered on.
3. Successful boot loading was achieved using a commercial Arduino and ICSP.
4. The Blink code functioned correctly, causing the LEDs to flash.
5. Test points on the board were accessible to measure the voltage between LEDs, such as 5V and 3.3V.
6. The switches distributed across the board allowed testing of the circuit in different required configurations.
7. The crystal oscillators at 12MHz and 16MHz functioned as expected.
8. Noise reduction of 20% to 50% was achieved compared to a commercial Arduino.
9. There was a significant decrease in Near Field emissions compared to a commercial Arduino.
10. The RESET switch worked as intended. The developed board met all the criteria mentioned above.

Key Takeaways:
1. To reduce inductive crosstalk of signals, maintain a continuous return path beneath signal traces.
2. Utilize the bottom layer of a two-layer PCB as a continuous ground return channel.
3. Avoid using individual wires or traces as return paths.
4. Consider Near Field Emission effects before building a PCB, as they can lead to Far Field Emission, which violates EMC standards.
5. Apply ferrite beads to filter very sensitive IC pins susceptible to mutual-inductive or self-inductive noise.
6. Position low-inductive decoupling capacitors close to the IC power pins to significantly reduce switching noise.
7. Consider placing ground vias beneath the IC surface to save space.
8. Use test points for quicker problem debugging.
9. Lay out signal pathways as far apart as practical.

Best Design and Measurement Practices:

1. Enhance debugging capabilities with isolation switches, indicator lights, and test points.
2. Position decoupling capacitors (VCC) close to each component's voltage line.
3. Distinguish between signal and power nets by using 6 mil-wide signal lines and 20 mil-wide power lines.
4. Implement a common ground plane or return path to reduce crosstalk.
5. Use spring-tipped probes to reduce measurement artifacts by lowering loop inductance from long cables.
6. Minimize circuit load by using 10x probes instead of 1x probes.
7. Utilize test points for efficient debugging.
8. Employ indicators for streamlined debugging and troubleshooting.

Corrections for Future Implementations:
1. To minimize ground bounce noise, maintain a continuous ground-return plane beneath all signal traces and keep cross-under connections to the ground plane as short as possible.
2. Implement a clear naming convention for components to facilitate debugging.
3. Use a standardized font size for improved clarity.
4. Consider individual isolation switches and test points for each of the load LEDs in a similar setup for better analysis.
5. Minimize the number of cross-under, if any, to reduce noise.