ECEN 3730 Printed Circuit Board Design: Board 3

Melissa Chavez

November 10, 2025

College of Engineering
& Applied Science

UNIVERSITY OF COLORADO **BOULDER**

Contents

Lab 21 : The Geng-Arduino

1 Objective

The objective of this project was to design, assemble, and test a fully custom Arduino microcontroller. This included completing a full schematic using appropriate components, creating a two-layer PCB layout that follows best practices for routing, decoupling, grounding, and header-pin placement, generating manufacturing files, and soldering the components onto the board. After fabrication, the goal was to assemble the board, burn the ATmega328 bootloader, verify correct USB-to-UART communication, and characterize the board's performance through power-rail and switching-noise analysis measurements. Overall, the project demonstrates the complete engineering workflow from concept through hardware bring-up and testing.

2 POR

- PCB will accept a 5 V external DC input and regulate a 3.3 V rail using an LDO. Stretch goal is to include a 9 to 5V LDO as well.
- Have appropriate PD system, 9 V, 5 V, and 3.3 V power outputs.
- Stable oscillator operation for both the ATmega328 (16 MHz) and CH340 (12 MHz) with correct 22 pF capacitors.
- Fully functional bootloader on the ATmega328 that allows sketches to upload over USB.
- Correct TX/RX routing between the ATmega328 and CH340 (TX $\leftrightarrow$ RX).
- Proper reset circuitry controlled by a button.
- Working USB interface with the CH340.
- Short, well-routed traces for crystal networks, USB data lines, and MCU connections.
- Designed Arduino header spacing should match commercial Arduino dimensions.
- Clear silkscreen labeling for all pins, components, and test points.
- Correct TVS diode connections on the USB D+ and D- lines.
- Stable power-on behavior with reasonable inrush current.
- Successful Blink test verifying MCU, USB interface, and bootloader functionality.

3 Design Planning

3.1 Block Diagram

To prepare for the design, I created a functional block diagram that organized the major subsystems of my board. It shows a high level overview of signal flow through the 9 V input, USB input, regulators, CH340 interface, oscillators, the ATmega328, and more. This planning step ensured I had a better idea of what to look for in inspiration designs from websites like Sparkfun.com.

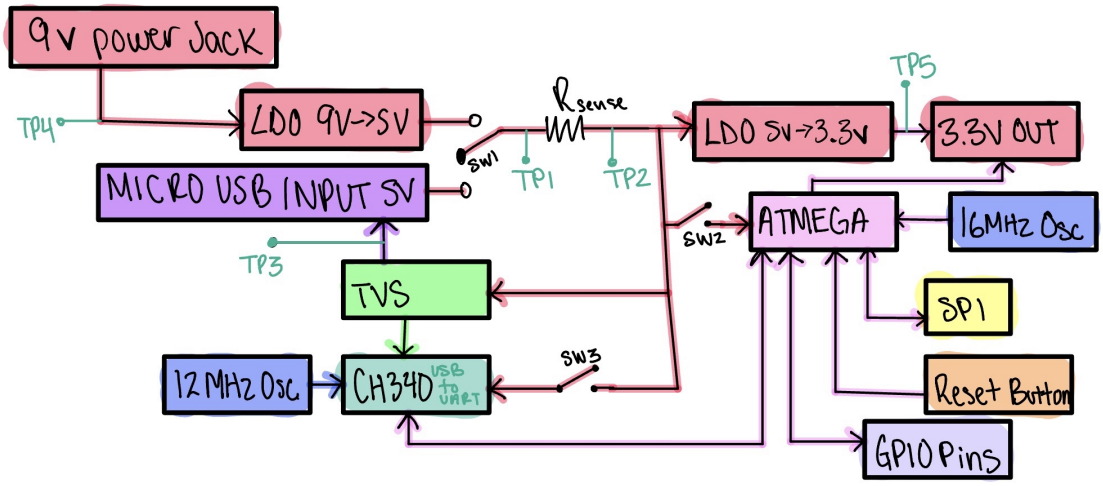

Figure 1: Arduino Functional Block Diagram

3.2 Design Specifications

The board supports two power sources (9 V barrel jack or 5 V USB), which are regulated to a stable 5 V rail and a 3.3 V rail for the USB ↔ UART interface. Each ATmega328 VCC pin is locally decoupled with 22 μF capacitors. Clock generation consist of a 16 MHz crystal on the ATmega328 and a 12 MHz crystal on the CH340, each with 22 pF load capacitors. The reset network uses a pull-up resistor network and a DTR auto-reset capacitor. Protection and measurement features include a USB TVS on D+/D–, and a low-value series sense resistor in the 5 V path for inrush/current measurements. The tables below summarizes the numerical design parameters used in this build.

Power Delivery	**Value**	**Unit**
Barrel-jack Max-input Voltage	9	V
USB Max-Input Voltage	5	V
LDO Output (Main Rail)	5.0	V
LDO Output (Peripherals)	3.3, 5, 9	V
VCC Decouppling Capacitance	22	μF
5 V Sense Resistor (Inrush Current)	500	$m\Omega$
Power trace width	≥ 20	mil

Crystal Oscillator	**Value**	**Unit**
ATmega328 Clock	16	MHz
CH340 Clock	12	MHz
Oscillation Capacitors	22	pF
Feedback Resistance	1	$M\Omega$

Other Important Specs	**Value**	**Unit**
ATmega328 Reset SW Pull-up	10	$k\Omega$
DTR → RESET capacitor	1	μF
Indicator LED Series Resistance	1	kΩ
Signal trace width	6	mil

3.3 Schematic

Having defined the design specifications and identified the most critical functional blocks from the system block diagram, a complete schematic was developed. The circuit integrates power regulation, USB communication, and the ATmega328 microcontroller core. A dual-LDO network converts 9 V or 5

V inputs into stable 5 V and 3.3 V rails with local decoupling and a sense resistor for inrush current measurement. The ATmega328 operates with a 16 MHz crystal, while the CH340 USB-to-UART interface runs at 12 MHz to enable reliable serial communication and bootloader uploads. Additional circuits include a debounced reset switch, auto-reset via the DTR line, ESD protection on the USB port, and Arduino-compatible headers for expansion and testing. The Altium Schematic can be seen below.

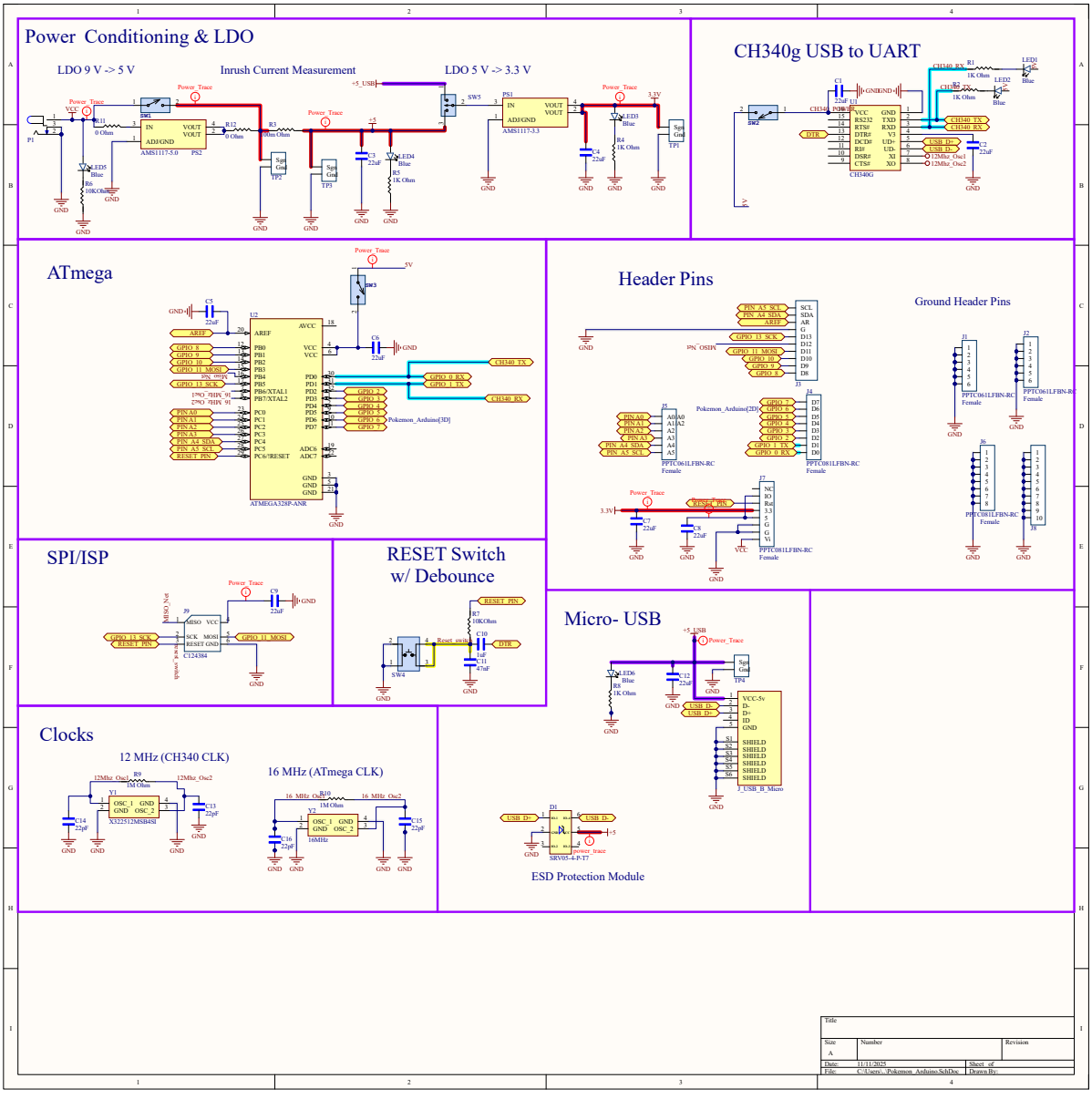

Figure 2: Arduino Schematic

4 Altium PCB Design

Please note the component values from the timer schematic do not reflect the component values calculated in Section 4. Due to the PCB being assembled by me, and the values were chosen accordingly during assembly.

4.1 Altium PCB Layout

After finalizing the schematic, a two-layer PCB layout was created in Altium Designer following best practices for grounding, trace routing, and component placement. The board was uniquely shaped in the outline of the Pokémon character Gengar, giving it the name Geng-Arduino. The placement of the ATmega328 and CH340 ICs was optimized to minimize trace lengths for critical signals such as the USB data lines, clock networks, and power rails. Wide copper pours were used to reduce voltage drop and noise, while local decoupling capacitors were placed close to each IC. The design also includes clearly labeled headers that match the Arduino Uno footprint, along with silkscreen identifiers for test points and components. The final layout demonstrates both electrical functionality and creative PCB design aesthetics.

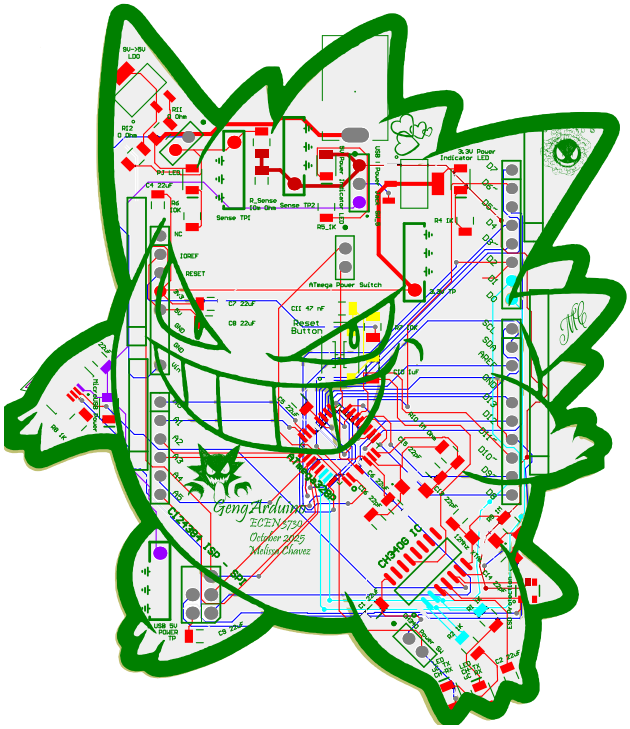

Figure 3: Altium Trace Layout

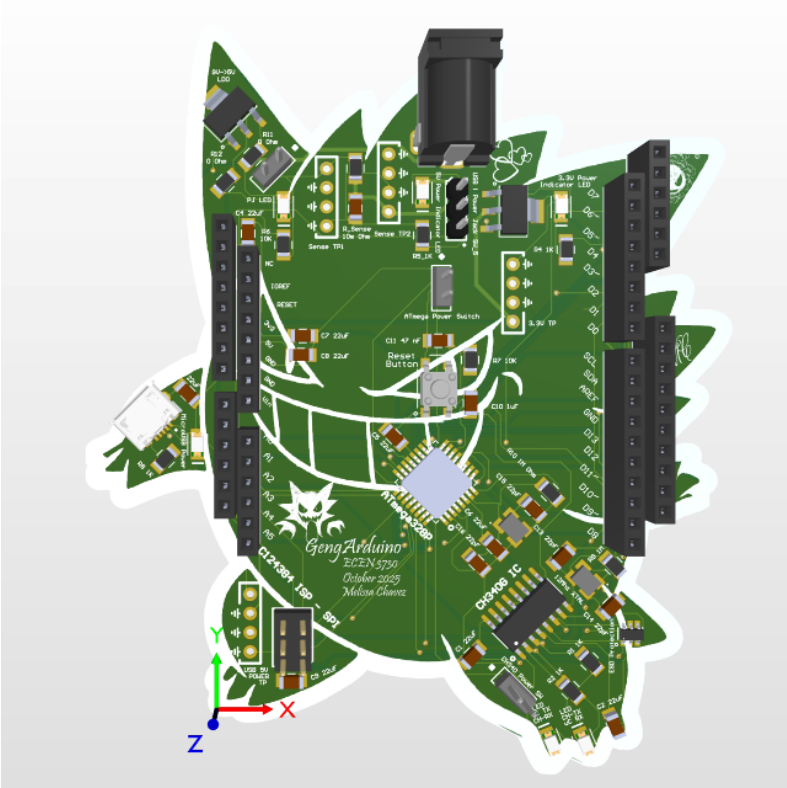

Figure 4: 3D View of Altium Design with Components

5 PCB Assembly, Bootloading, and Debugging

5.1 Assembly

The assembly process began by applying solder paste to all surface-mount pads using a fine-tipped syringe to ensure precise placement. All surface-mount components were then carefully positioned using tweezers and reflow-soldered in a PCB oven. After reflow, each component was inspected visually and tested for continuity and shorts to verify solid connections and proper orientation. Once all SMD joints were confirmed to be secure and free of defects, the through-hole components, including headers, connectors, and switches, were hand-soldered. This hybrid assembly approach ensured reliable electrical connections and a clean, professional-quality board finish. The fully assembled board can be seen below.

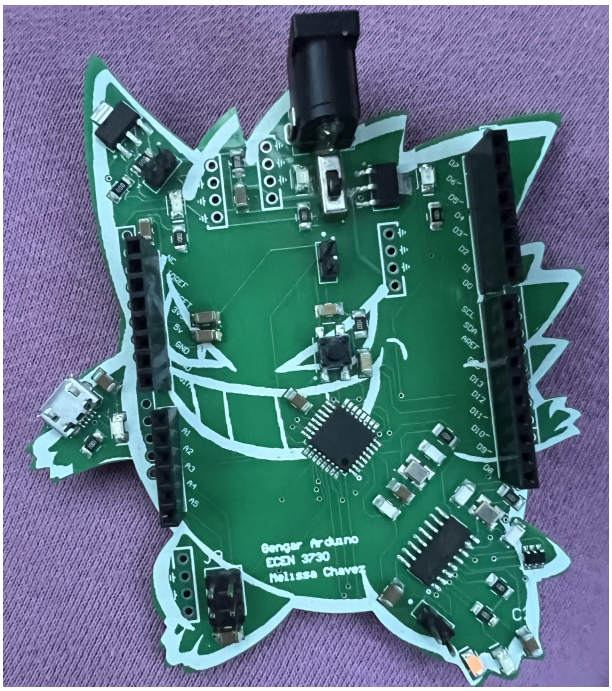

Figure 5: Fully Assembled Geng-Arduino

5.2 Testing and Bootloading

- Test 1: Powring On - Plug into 5V USB and 9V power jack. Indicator LED's should indicate power and multimeter measurements will verify the voltage of each rail.
- Test 2: Bootloading - USB TX $\Leftrightarrow$ RX
- Test 3: Uploading Code - This step verifies the bootloading as well. A blink sketch will be uploaded to verify successfull bootloading and USB to UART programming.

5.3 Testing and Bootloading Outcomes

- Test 1: Powring On - Indicator LED's for the 9V, 5V, and 3.3V rails powered on. Multimeter measuremnts verified the voltage at each rail, and proper functionality of the LDO's.
- Test 2: Bootloading - The arduino IDE kept stating it was expeting a different ATmega signature. This will require further debugging.
- Test 3: Uploading Code - Sketched will not upload, proper bootloading cannot be verified.

5.4 Debugging Tests

Each test is described in detail below, in the same order as listed above:

1. A continuity check of all manually assembled components showed that the indicator LED for USB power connection was not routed to its corresponding series resistor. It also revealed that the USB power switch, which allows selection between the barrel jack and USB power, did not actually connect the USB line to the 5 V rail. This was corrected using a jumper wire, visible as the red wire in Figure 6.
2. Verification of resistor and capacitor values showed no issues.

3. Power delivery testing revealed that when USB was connected, the ATmega328 and CH340 did not receive power. This was due to a schematic mistake corrected using the same jumper wire mentioned in Test 1.
4. Comparison of the Altium schematic and PCB layout verified correct routing.
5. Verification of CH340 TX and RX connections to the ATmega RX and TX pins showed no issues.
6. SPI and ISP programming connections to the ATmega were verified with no errors.
7. Crystal oscillator orientation and connections to both ICs were confirmed correct.
8. All input and output pins of the ATmega, CH340, and micro-USB port were verified and functioning as expected.
9. The TVS diode connections were confirmed to be correct.
10. After attempting to bootload from a different computer, it was discovered that the original Arduino IDE installation was corrupted due to modified source files from previous labs. Bootloading was successful when attempted on a clean system, confirming that the hardware was fully functional.

5.5 Debugging Conclusions/Solutions

Overall, the debugging process confirmed that the board design was electrically sound after minor corrections. The primary hardware issues, an unconnected USB-to-5 V power line and a disconnected LED trace, were resolved using jumper wires. Subsequent verification of all component connections, signal routing, and decoupling networks showed no further electrical faults. The final test revealed that previous boot-loading failures were caused by a corrupted Arduino IDE installation rather than hardware defects. Once a clean installation was used, the ATmega328 successfully boot-loaded and communicated via USB. This process reinforced the importance of systematic debugging, schematic-to-layout verification, and isolating software versus hardware faults in PCB bring-up.

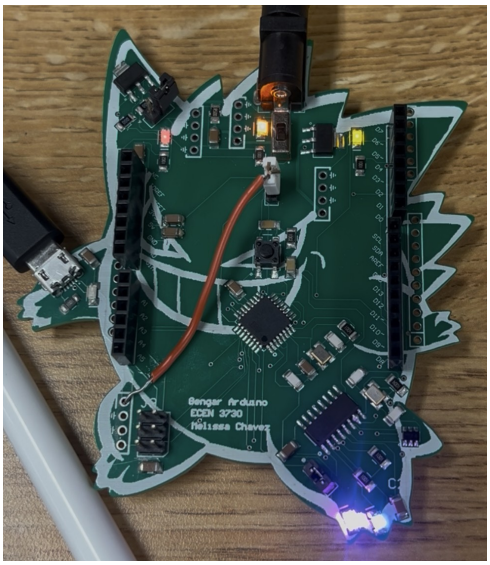

Figure 6: Populated, Functional PCB Showing Uploaded Blink Sketch

6 PCB Measurements

A major layout oversight discovered during testing involved the header pin alignment on the right side of the board. The GPIO pins 0-7 were positioned above pins 8-13 instead of below them, preventing the class-provided Arduino shield from mating properly with the board. As a result, direct performance comparisons with the commercial Arduino were not possible, and no shield-based measurements were taken. However, switching noise was still characterized by performing a differential oscilloscope measurement across the 500 $m\Omega$ sense resistor on the 5 V rail. The resulting differential measurement across the sense resistor is shown below. The pink waveform is the numerical difference between the two waveforms.

6.1 Oscilloscope Shots

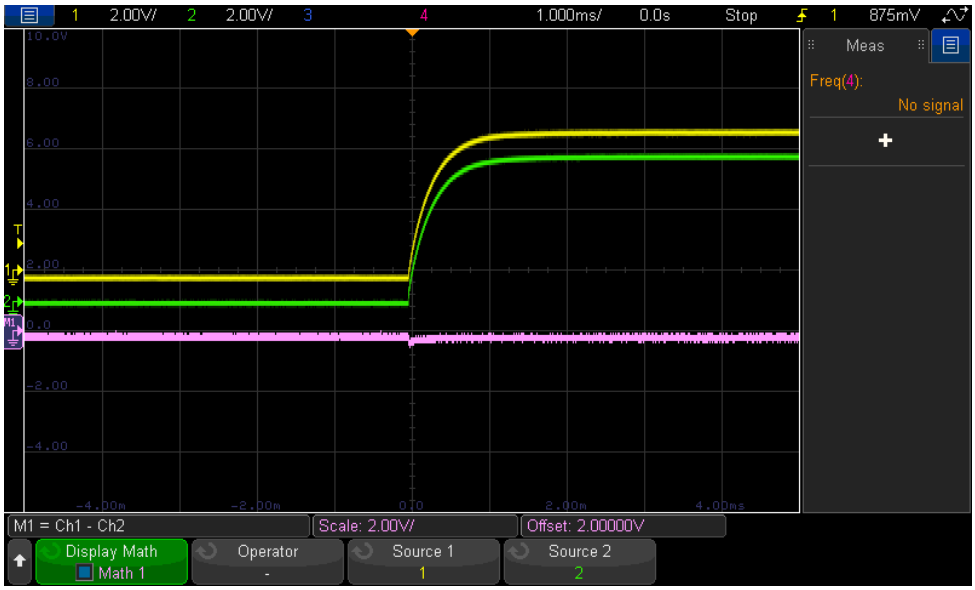

Figure 7: Inrush Voltage of GengArduino

6.2 Measurement Results

Measurement Results	Value	Unit
Inrush Max Voltage	6.7	V
Sense Resistor	500	$m\Omega$
Calculated Inrush Current	3.3	A

7 Conclusion

This project successfully demonstrated the complete design process of a custom Arduino-compatible microcontroller board, from schematic creation and PCB layout to assembly, testing, and debugging. Although minor design oversights were encountered, such as reversed header alignment and an incomplete USB power connection, these were systematically identified and corrected during debugging. Final testing verified stable power delivery, proper crystal operation, and full USB communication once software issues were resolved. Overall, the GengArduino project achieved its core objectives and provided a strong foundation for understanding the end-to-end workflow of printed circuit board design

and validation.

7.1 Lessons Learned

Key takeaways and lessons learned from this project include:

- The importance of verifying header pin orientation and connector alignment before fabrication.
- Careful review of schematic-to-layout consistency to prevent routing and power delivery errors.
- The value of systematic debugging and continuity testing to isolate hardware versus software faults.
- Recognizing that software issues, such as IDE corruption, can mimic hardware failures.
- Gaining hands-on experience with solder paste application, reflow soldering, and inspection of SMD components.
- Understanding the workflow of designing, assembling, and validating a functional PCB from concept to prototype.