Arduino Nano clone- documentation and learnings (v1.0 > v2.0 > v3.0)

By: Devansh Sharma

little grammatical assistance by AI

This project is an Arduino Nano-compatible board designed using ATmega328P-AU and CH340G USB-to-UART IC. Unlike many Nano clones that rely on pre-assemble USB-UART modules, this design intentionally uses the bare CH340G IC, as the goal is to remove dependency on third-party modules and make the board electrically self-sufficient and gain as much learnings as possible.

The CH340G was chosen instead of FT232 (as used in official Nano schematic) because of the following reasons:

- Significantly more cost effective.
- Widely available.
- Have native windows driver support.
- It is a default choice for many modules.

Power Architecture (Why v1.0 is revised to v2.0):

In **v1.0 Power Architecture** (Initial Implementation) (which I used in UNO v2.2), the following power design was followed:

- ATmega328P was powered from external 5V via an AMS1117 regulator.
- CH340G was powered directly from USB 5V
- An SS14 Schottky diode was used with anode on USB 5V and cathode on external 5V rail.

This arrangement ensured that:

- During USB code upload, both ATmega328p and CH340G were powered from USB 5V.
- When no code uploading (no use of CH340G), ATmega328P was powered from external 5V.
- The diode prevented external 5V to back-feeding into USB when both sources were used. Here it to not that in this case ATmega328P will get power from AMS1117 and not USB (although both are available) because AMS117 keeps ATmega328P node on a higher volage node (voltage drop happens at diode), and thus USB supply will not cause issue, as it is physically impossible for USB (effective <5V) to push current in a node already on 5V.

Although this design was electrically fine and did not violate any thing. But it had drawbacks as, it splits power domain which increases complexity and makes behavior of board depend on the usage, as whether USB is used or external 5V is used. Also, this is not fully aligned with reference Nano behavior (which is of course a better Power Architecture). And this realization, learning and understanding led to v2.0, with much more robust Power Architecture.

v2.0 Power Architecture (Improved):

- Both ATmega328P and CH340G shared a single 5V rail.
- And entire board can be powered from: USB 5V or external 5V via AMS1117, and both.
- With SS14 diode orientation having anode on USB 5V and cathode to external 5V rail.

This arrangement ensures:

- USB power is used during programming.
- External 5V takes priority during normal operation.
- No back-feeding into USB
- When both are used, external 5V will be selected for power (as it will put nodes at 5V, which is greater than what USB will give through diode, and this will not increase any current cause nodes are already at higher voltage) and USB for programming.
- Predictable behavior regardless of connection issue.

Design and components selection decisions:

Why USB C was used and avoided Micro-USB, as in Nano board:

USB C is more robust than Micro-USB, have symmetric shell, is small in size as compared to other connectors like Type-A, Type-B. It is not easily damaged just cause cable is not connecting because of wrong orientation, as here electrical orientation is handled by CC pins and not as a default 5V port as Micro-USB (explained later in documentation), allowing board to correctly identify itself as a USB device (sink).

Here I am using a 12 pin USB C (USB2.0), and not 16 or 24 pin to ensure universal usage and also the fact here is no usage of fast PD. So, using any other will not give a significant impact. Thus, it is here ensuring universality, robustness, and still being electrically simple.

Why was 5.1kΩ was chosen with CC pins and not any random value:

First of all, here 5.1kΩ resistors are the USB-C CC pull-downs (Rd) and not present for function of current limiting, signal conditioning, protection or Arduino logic.

Their only job is USB-C role detection. Because USB-C is not like Micro-USB, as with USB-C power is not guaranteed, unless the device identifies itself correctly. As here host (PC or Laptop) must detect “A USB device is connected, provide 5V”. And this is done by CC pins and not by VBUS pins. Now here we use two 5.1kΩ resistors because here are two CC pins, as USB-C can be used in any orientation.

The value of 5.1kΩ is defined by the USB Type-C specification for sink (device) identification, where source (host device) is your PC or laptop. If we don’t use 5.1kΩ, then VBUS may not be enabled, preventing enumeration (no resistor used), host may not enable VBUS (wrong value), host misinterprets role (too low value), detection may fail (too high value).

Thus 5.1kΩ is not a arbitrary value, but a mandatory one.

Why both 100nF and 1uF capacitors both are attached to AVCC of ATmega328P:

Although AVCC is internally connected to VCC and Port C circuitry, it is an electrically sensitive area, as its stability directly impacts the stability and noise in ADC pins (or analog pins) output. Thus, it is to be treated as separate analog supply rail rather than a simple VCC pin. In this design AVCC is decoupled using both a 100nF ceramic and a 1uF electrolytic capacitor, with 100nF placed close to AVCC. Here 100nF will be a high-frequency decoupling, reducing fast transient noise caused by internal digital switching in MCU. Whereas, 1uF will act as bulk decoupling, removing low frequency noise, stabilizing AVCC against slower voltage fluctuations and current demand during ADC pins working.

While most UNO reference uses only 100nF, here an extra 1uF is used to provide extra stability, as it is suggested by official datasheet.

Why 1kΩ as pull up reset resistor instead of 10kΩ (in UNO):

Unlike Arduino UNO which uses a 10kΩ reset pull-up, the nano design uses a stronger 1kΩ pull-up to improve reset stability and noise immunity in compact, USB-powered layouts (as it is a denser board). The auto-reset circuit still functions correctly due to the capacitive DTR pulse.

Now the layout decisions which led from a board with routing limitations (v2.0) to final layout (v3.0):

v2.0-why it went wrong:

- In v2.0, I started routing immediately after placing components, not thinking properly how the routing will go.
- In v2.0 I directly started to do routing thinking to pour ground at last, which resulted in routing congestion as during routing I didn't think about minimum fabricable clearance. And this mistake made grounding pour impossible and gave more than 100 DRC errors, all based on clearance issue, via size. So, without putting my excess head into it, fixing each DRC on an already non improvable routing (as fixing was giving out new issues), I decided to move to another PCB layout (v3.0).

How it went in v3.0:

- In v3.0, I first placed all the components, and after learning from mistake in v2.0, I optimized placing so that routing doesn't become inefficient.
- ATmega328P, USB-C, both crystals, electrolytic capacitors were placed above. While CH340G, AMS1117 at bottom. Capacitors and resistors are placed in such way to achieve as short possible short return paths.
- Then I runed auto-router **JUST TO SEE WHAT IT WAS THINKING AND NOT USE IT AS FINAL CHOICE.** I realized that how it used top layer for most connections and bottom layer as a support, not as a battleground (v2.0).
- Also, first of all I pour ground on top and bottom layer with a clearance on 0.203mm of clearance, so that I do not run again into same clearance issue.
- As inspired from thinking of auto router, I proceeded to use top layer for major connections, and placed components, mainly resistors and capacitors having some pad connected on opposite layer, connected through via (not an issue here as I will be hand soldering all components).
- Carefully routed decoupling capacitors, 16 and 12 MHz crystals, load resistors with as short as possible traces, without any via, to optimize noise reduction.
- Placed USB-C, ATmega328P and CH340G close to increase D+/D-, RX, TX signals.
- Traces of D+ and D- are made up of as same length as possible.
- After these important traces, D4-D12 and other analog output were done, as these made v2.0 feel very difficult at end.
- Then bottom layer components were traced, accordingly. And leftover traced we made carefully taking clearance as priority.
- This all work then resulted in board which is schematically, electrically correct with DRC=0.
- Also, I had used vias in empty areas of board to stich top and bottom ground layer, to further reduce impedance and provide even more shorter return paths.