

1. Cool thing I made



This wasn't the most complex thing I've made, but it was by far the thing I spent longest on. I made these machines to sell during the cryptocurrency fad to earn some extra cash to pay for college. At the time, I hated how technical and clunky these things were, so I made mine super easy to use. All the low-level stuff like overclocking was handled through a bash script I wrote, executed using a cronjob. I made it so that everyone could use it, and all they had to do to set it up was plug it in. Not an exaggeration. I'm super grateful for this because this was my first real engineering project and it's what allowed me to go to college.

2. IDE setup

In college, I'm away from my PC that I built, but at home I have an ergonomic chair(!!!), a G913 keyboard, and a monitor my friend had laying around. Not the fanciest setup, but it gets the job done. Sometimes the monitor acts up, and that's usually when I go for a walk to take a break. As for IDEs, I use VS Code, making daily backups of my work folder to my VPS. When I'm working on class stuff, I usually have to SSH to a VM, so I usually debug code with nano.

3. PCB that resets

I dealt with a similar situation with DAQ boards at a club that makes rockets. After a harsh landing, we usually have to manually download data from the DAQ, and we do it quickly because we don't want to be the reason we have to re-launch the rocket.

I'd find a way to image the board using whatever diagnostics we have, maybe dump the memory before/after the reset. If I have time, see what affects how long the board takes to reset (to narrow it down). Double check the pins because I might've accidentally shorted a RESET pin to some data line.

4. Engineers do wrong

As a mathematician, I disagree with "if it ain't broke, don't fix it". Anything going to prod should work even if it's getting abused. Like proofs in math, I want to be able to at least read it and know that it should work.

5. Sketchy

Back to question #1, running a bash script via cronjob 1x daily was super sketchy, but it got the job done. In hindsight I should've made a background process that monitors certain things (temp, cache, etc) and runs the relevant script when it reaches a threshold.