

Prototyping Tricks and Tips

SparkFun Electronics Summer Semester

Prototyping Tricks and Tips

Here are some tips and tricks that will help anyone, no matter what their level of expertise is, when prototyping, debugging or just plain old fixing a Physical Computing project. Some of them are from Nate's experience working at a Kid's Museum called World Of Wonder, other tips and tricks were gathered from other people at SparkFun. Got a tip or trick that you think would be useful to others? Let us know and we'll put it in this document.

1) Document your work. Realize you won't be around next month so please leave behind everything you can to help the next person support (repair) your work. Things I now leave inside the actual exhibit: clear schematics, a wiring diagram, an overall layout, all the firmware on a jump drive, operation instructions and maintenance information for the support staff, and contact details (my name, my email, cell # if needed, etc).

2) Use polarized connectors. This may seem silly but I've seen a lot of projects where the creator soldered wires point to point. This means that the wires were soldered directly onto the connections on the board. Not only is this slow, it is unmaintainable. You should *always* be able to remove a board or sensor by unplugging the various wires. This makes it possible to swap out parts to determine what might be broken, increasing the speed of debugging. Furthermore, you should always use polarized connectors. If the staff needs to relocate a piece to a different part of the museum, asking them to decipher and follow a thin sharpie mark (or even worse labeling) on a perf board is dangerous. Polarized connectors will help prevent your piece from sitting unused in the back room.

3) Buy pre-made cables. This goes hand in hand with the polarized connector lesson. You should not be running down gremlins (aka problems) in your wiring harness. Use premade cables that you can trust. I personally love to use CAT5 cable: it's cheap, readily available in many different lengths, and you can even get different colors! If multiple of the same type of cable run into a board then throw the cable color ("Red") or cable name ("IR2") on the PCB for easy identification. I also use the SparkFun JST assemblies by the handful. Buying these pre-made cables and connectors allowed me to concentrate on bigger problems than getting my crimps correct.

4) Label stuff. On a complex piece there will be large amounts of wires running everywhere. A \$30 label printer will save your sanity and the person who looks at the piece after you. Label *everything*. This is obviously a USB cable, but what does it go to? **Main controller debug 38400bps** would be a great label. **"IR LED 4"**, **"Red to Detectors 1-4"**, **"To Light Controller"** are all great cable labels.

Prototyping Tricks and Tips

SparkFun Electronics Summer Semester

5) Leave the debug cables in the unit. Illumitune (piece at World Of Wonder) had a total of 4 microcontrollers. At any given time I needed to reconfigure two of them regularly. I had the idea of running USB cables from the boards down to the side access panel. I left these USB cables permanently installed in the exhibit. This was awesome! It saved me from having to climb a ladder and plug in a USB cable every time I needed to debug the system. By leaving the 'debug cable' in your exhibit you also increase the odds the next person to maintain your exhibit will be able to connect to it. Please don't assume that next user will be able to connect to the three odd spots on your perf board to get to TX/RX/GND. By leaving a standard USB A-to-B cable in the exhibit I assume that USB ports will be supported for the next 3-5 years.

6) Give me some debug information. Make the debug port obvious. I should be able to walk up to a piece and within a few minutes be able to plug in a netbook and see something coming out. Ideally the output would tell me what I'm looking at and looking for. For example, when you power up Illumitune, it states 'IR Controller Online' then 'Light Controller Online' then it displays '\$3#' when beam 3 is broken.

7) Own a logic analyzer. I wasted about 8 hours of painful troubleshooting because my IR timing was off. 15 minutes after hooking up my logic analyzer and it was obvious what was wrong and I had a solution in place. I usually do all of my debugging with print statements. This works for much of the time, but for the situations where one is needed (troubleshooting I2C, SPI, or carefully timed procedures like IR transmission) a logic analyzer is worth its weight in platinum. No really, get one. Your sanity will thank me later.

8. Double check your Serial lines. This is the number one error every engineer at SparkFun occasionally makes. It's silly and it's easy to do even after years of experience. Luckily it's easy to check as well.

9. Work with a breadboard first. There's a reason why engineers use breadboards before embedding. With breadboards you can see if your circuit works before moving on to the step of embedding or hooking up an interface. Breadboards are your friends.

10. Don't be afraid to just plug it in. Stop worrying and see if it works. That said, we highly recommend using PTC's for basic breadboard prototyping. If it worked on the breadboard it should work when you plug it in.

11. Order twice what you think you need of the cheap stuff. You're going to want to make a second prototype, you're going to burn some materials, and you're going to measure incorrectly. Why not save yourself another trip to the hardware store by buying a little extra on the first trip?

Prototyping Tricks and Tips

SparkFun Electronics Summer Semester

12. Don't be afraid of the basics. Often times the most useful and innovative designs and devices are built off of the solid foundations of previous technology. By all means don't hesitate to make those creative leaps and operate without a rope so to speak, if you feel comfortable doing so. But be aware that if you're trying something completely new and going for pure innovation, there's nothing to double-check, use as a marker for truth, or sanity check when something goes wrong. The problem with exploring completely new territory is that if you get lost, there's no easy way to find your way back home. This is why it's always good practice to utilize those basic designs that have been rehashed a thousand times. Building a good percentage of your project off of tried and true technology gives you a compass for troubleshooting your design should something malfunction, eliminating the guesswork, and isolating the problem to a much smaller set of probable issues. The best engineers will tell you that drilling the basics is the most important and useful way to ensure that your next crazy, new idea gets off the ground.

13. Don't use ferric chloride for etching your own circuit boards. Use hydrogen peroxide mixed with muriatic acid (both available from home depot). If you have access to a laser engraver, you can make decent double-sided prototype boards by coating the copper clad board with flat black spray paint and then etching away the paint around your circuit traces. Then just etch the board using muriatic acid and H₂O₂.

14. Double check your PCB files. When troubleshooting your PCB try to catch as many errors as you can instead of just fixing one error and waiting for the PCB to come back so you can test it and find another error. If you try this approach you will save yourself a lot of time as well as PCB cost. In order to catch all the errors first troubleshoot do whatever it takes to get the first run PCB working with your hardware. Green wire fix everything that doesn't work and take note so you can adjust in your Gerber files. Also assume the manufacturer is going to misprint your board a couple of times you order PCBs.

15. Less moving parts is better. Try to find simpler, more elegant solutions for things like physical interfaces and mechanical aspects of your prototypes. Less moving parts means less stuff that could conceivably break, makes sense right?

16. And finally, even though you don't want to... take a break. No really, take a break, eat some food, take a walk, take a nap, or if you refuse to walk away at least work on a different portion of the project. Save yourself some frustration, because things will seem so much clearer at 3 pm than at 3 am.