

WATERLOO ROCKETRY TEAM

Electrical Design and Assembly Standards

All mission critical electrical systems going to comp must meet or
exceed these standards

Aaron Morrison

June 30, 2017
Revision 0.1

1 Guiding Principles of all decisions

This document exists to set out a standard for all the electrical gear (avionics, telemetry, ground support equipment). This document is very much still in development (it's based on 3 pages of notes I made in a notebook in the desert), and I am by no means an expert. Changes/revisions are very welcome. All design decisions on mission critical electrical systems should be made with the following four points in mind.

1. Our electrical stuff needs to work 100% of the time at competition. If that's unattainable, 6 sigma is also acceptable. A lot of work from a lot of people go into building this rocket, and missing our 1 launch attempt because of an electrical problem is not an acceptable thing.
2. Once we leave the bay, all electrical work should be done. Fixing something in the desert requires running a generator in a trailer in 40 degree heat. No one does their best work in 40 degree heat, any MacGyvered solution you come up with there will be orders of magnitude worse than a proper solution you create in Waterloo, and if you're in the desert, there are better things you should be spending your time on.
3. The rest of the team should never be waiting on a fix from us. Electrical stuff should work, and it should work as soon as we need it to. With electrical systems we have the luxury to make sure they work in advance (the engine team does not have this), and we should know all the failure modes and be prepared for them.
4. We should never be asking another university's team for gear in the desert. Whether or not we fly should not be at the whim of random chance of whether we can find a thing we could have packed a spare of. If something can break, you should have a spare. If there's a tool you need to do a thing, you should have it with you. We bring everything we need, we shouldn't be asking around for gear.

This document is written in descending order of importance. If you need to decide between whether to follow a guiding principle or a hard rule, follow the principle. As long as your decisions demonstrably don't violate any of these 4, you're probably ok.

2 Rules

Here are the rules for all the design and assembly decisions. If you want to break one of these rules, make sure you have a very good excuse. Again though, this document was written by someone who doesn't actually have any idea of what he's doing, so if you think a rule is stupid, feel free to cross it out (provide a reason though). If you think a rule is unclear, feel free to elaborate on it. If you think there's a good rule that should be added, please write it in in the margin or pull request the repository (we should get a repository). Anyways, list time.

- Don't solder stranded wire to protoboard. Don't solder stranded wire to a connector. It gets brittle and it breaks.
- Don't crimp something onto solid core wire. That doesn't work
- All mission critical components should have a hotswappable replacement. You should have that replacement's location written down.
- Hotswappable means you can replace it without any tools (except maybe pliers or a screwdriver. But no generator)
- All exposed wire should be heatshrunk, all heatshrink should be clear.
- If supplies allow, use different a different insulation color for each wire (at the very least, positive should be red, ground should be black, and all other wires should not be red or black)
- If you screw up a soldering job (we all do it), redo the job
- All connections must be tug proof.
- If your system has any software anywhere, that software should be unit tested
- All wires you use need to be low enough gauge to match the current you're putting through them
- If you're using an RC lithium polymer battery, it should be fused (those fuckers can sink 50 amps)
- If it's conceivably possible for a component to be installed backwards, there should be idiot proof documentation of which way it goes, that documentation should be less than 2 inches from the install point
- If a connection could be asymmetric, it should be asymmetric
- All switches need idiot proof labelling
- But don't use actual tape labels (they peel off easy. Use a sharpie)
- All mission critical software should be expressible (and accompanied by) less than 100 lines of pseudocode.
- If your wires get handled or moved or torqued at any point, it should be strain relieved on both sides of the torque point
- If you absolutely need to solder stranded core to something, make sure it's very strain relieved (I personally like P clamps, but zip ties work too)
- Tight wires between two points are accidents waiting to happen
- Strain relieve everything to the point that it can survive violent shaking (if it's actually going in the rocket, it should survive very violent shaking)

3 Documentation Standards

If you're building some piece of gear, or you're leading a project that does, it is your responsibility to ensure that all the forms of documentation are written decently and put in the google drive.

There are three types of documentation that should be written for every piece of electrical gear that comes to competition:

1. Users guide
 - Step by step operating instructions
 - Simple enough to be used by artsies
 - Partway through a procedure, if a thing can go wrong, say what it is, how you can tell it's gone wrong, and a reference to your troubleshooting guide on how to fix it
 - This probably just needs to be a single checklist/procedure
2. Troubleshooting Guide
 - detail every failure mode you can think of (and try to think of more)
 - include the symptoms of every failure mode, what the worst case scenario is, how to diagnose this failure mode conclusively, and how to fix it
 - this should have a list of every component in your system, and how to test that component in isolation. This list will be useful for building spares kits

Development Logs

- Every meeting's minutes, every block diagram you draw, all your rambling dev notes, everything you do pertaining to this system during the development cycle
- Take all this, pick good filenames, and stick them all in a folder in the google drive
- This will be incredibly useful for future generations who want to learn this stuff
- Include data sheets for everything you use

4 Life advice from someone who doesn't actually know what they're doing

- Buy spares. You will fuck things up, be prepared
- Never solder at hotter than 700 degrees. Even that is probably pretty aggressive. Use leaded solder, it melts at a low temperature so you won't burn off all you flux

- All software should use a VCS (probably git), and all final versions of your software should go with your dev logs on the drive
- Don't short things. If you have any bare wires flailing around, don't apply power to the system
- Assume that you will not be there when your system is used. Your documentation should be so good that someone who has never seen your system can fix/use it
- Wires and electrical components do take up space. Please keep this in mind
- Keep good isolation. That means standoffs and that means decent creep-age distances.