Waterloo Rocketry Team

Electrical Design and Assembly Standards

Aaron Morrison
Alex Mihaila

October 9, 2017
Revision 0.21

1 Guiding Principles of all Decisions

This document exists to set out a standard for all the electrical gear (avionics, telemetry, ground support equipment). 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 acceptable.

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 whether we can find something we could have packed a spare of. If something can break, you should have a spare. If there's a tool you need, 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 (`https://github.com/akmorrison/waterloo_rocketry_electrical_standard`).

2.1 Connections

- Don't solder stranded wire to protoboard. Don't solder stranded wire to a connector. It gets brittle and it breaks. Use crimped connections, screw terminals, etc instead.
- If you're making a crimped connection, use a proper crimp tool if at all possible. Pliers work in a pinch, but the connection is messier and more likely to fail. Good crimp tools are expensive but worth it. (If you're in MME, you can borrow one from MESS.)
- 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).
- Crimped connections should be done on stranded wire only. Don't crimp onto solid core. It's weak and it will break, often when you least want it to.
- All exposed wire should be heatshrunk; all heatshrink should be clear (IREC regulation).
- If you screw up a soldering job (we all do it), redo the job. When soldering surface mount components, inspect solder joints with a small magnifying glass.
- All connections must be tug proof.
- If a connector could be asymmetric, it should be asymmetric, since asymmetric connectors are much harder to plug in backwards.

2.2 General Wiring & Assembly

- Wiring should be done so that the circuit is as readable as possible. 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).
- All wires you use need to be low enough gauge to match the current you're putting through them.
- If your wires get handled or moved or torqued at any point, they should be strain relieved on both sides of the torque point.
- 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).
- Tight wires between two points are accidents waiting to happen. Avoid these.
- If you're using an RC lithium polymer battery, it should be fused (those fuckers can sink 50 amps).

- When designing a box, circuit layout, or other assembly, keep in mind that wires and electrical components do take up space. Leave clearance in the design stage; this problem should be solved before fabrication/assembly.
- Keep good isolation. That means standoffs and that means decent creepage distances.
- When using shielded cable, ensure the shield is properly terminated and grounded. Shields should only be grounded at one end of the cable.

2.3 Software

- If your system has any software anywhere, that software should be unit tested.
- If your system requires a computer program to be used, that program counts as mission critical. It should run on at least two machines and at least two people should be trained to use it.
- All mission critical software should be expressible (and accompanied by) less than 100 lines of pseudocode.
- All software should use a VCS (probably git), and all final versions of your software should go with your development logs (see Documentation Standards) on the drive.

2.4 PCB design

- Given the choice, please draft your schematics and board layouts in KiCad. It's free (GPLv3), used by the rest of the team, and runs on all platforms. `http://kicad-pcb.org/`
- if your PCB's are meant to be screwed onto something, put those mounting holes on your layout, and if at all possible, use M3 screws (for standardization purposes, if you have a space restriction, smaller screws are fine).
- Before ordering any components, make a list of all components in your design, and go through their datasheets one by one and ensure that they meet or exceed the requirements of your design. Do not just trust the Digikey table values, they are occasionally mislabeled.
- The cheapest board that can be purchased in small quantities is a 2 sided 1oz 5cm by 10cm 10/10 (10 mil (.254mm) min clearance and min trace width) 0.4mm smallest drill size board. If at all possible, try to make your design fit that requirement.
- If a pin is listed as NC on a datasheet, don't pull it to ground. Just leave it unconnected. Unused IO pins are a different story, they may need to be shorted to ground or VCC. Consult your datasheet in this instance.

- If you plan to hand-solder your boards for assembly:
  - If you plan to hand solder components to your board, use pads that are longer than you need. KiCad has specific footprints whose names end in _handsoldering, which have extended pads for easier handsoldering.
  - Use flux. It makes surface mount soldering orders of magnitude cheaper. 10ml syringes are less than $20 at Sayal, and should last for a while.
  - SOT-23 and 0805 SMT components are quite easily handsolerable. SOT-23-5 are also doable, but are harder. I've never attempted anything smaller than those.
  - BGA components can't be handsoldered. Bear this in mind while choosing components. There might exist a reflow oven somewhere on campus that we have access to, but laying paste will be a pain. Avoid BGA packages if you plan to assemble by hand.
  - Don't batch assemble boards (if you need multiple) when they arrive, because you will be less inclined to change your design if you notice small bugs if you've already committed lots of components (which are worth far more than the cheap boards) to this design. Build one, test to death, then decide whether to make another revision or to assemble the rest.

2.5 Miscellaneous

- All switches need idiot-proof labeling. Labeling should be permanent (i.e. not tape, etc).
- 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).
- 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.
- Buy spares. You will fuck things up, be prepared.
- Don't solder at hotter than 700°. Even that is probably pretty aggressive. Use leaded solder, it melts at a low temperature so you won't burn off all your flux. If your solder isn't melting below 700°, the iron's tip is probably bad and should be replaced. Take care of your tips!
- Don't short things. If you have any bare wires flailing around, don't apply power to the system.
- All battery connections should be terminated in a female connector to prevent battery shorting and enable easy battery swapping.

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 five types of documentation that should be written/assembled for every piece of electrical gear that comes to competition:

1. Users guide
    - Step by step operating instructions.
    - Partway through a procedure, if something 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.
    - 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.
    - 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.
3. Development Logs
    - Document every decision you make. Include meeting minutes, block diagrams, etc. The point of this is so future generations maintaining your project know the reasoning behind your engineering decisions. They shouldn't be left guessing.
    - Take all this, pick good filenames, and stick them all in a folder in the google drive.
4. Datasheets
    - Include datasheets for everything you use. Keep these in the same folder as the rest of your project documentation.
5. Schematics & Board Layout

- Any circuit you build should be thoroughly documented in a schematic. Schematics should be correct, clear, and well-organized. The primary purpose of a schematic is to communicate the circuit to someone who hasn't seen it before. A messy schematic won't make the person reviewing your circuit too happy, so it pays to keep things neat (yes, you should be getting other people to review your designs).
- Some guidelines can be found at `https://electronics.stackexchange.com/questions/28251/rules-and-guidelines-for-drawing-good-schematics`.
- This may seem obvious, but the components in your schematics should match what you intend to use in the circuit. If you are using a particular component, label it as such. If your schematic editor doesn't have a symbol for the part you intend to use, make a custom one (all decent editors support this). Eg. a TL780 voltage regulator should not be represented by a BD244 transistor.
- Label output terminals properly. For example, a 12V output should be labeled as such. It should be clear what the inputs and outputs of any particular part of your circuit are.
- If you are making a PCB, the board layout and all supporting files should be uploaded to the Google Drive. Include both PDFs and the original files.

4 Design Reviews

If you're designing any part of a system, you should be conducting design reviews. This applies whether there are multiple people working on the system or if you're the only one. If you're alone on a project, this is a good opportunity to make someone else familiar with your work. Apart from resulting in a better circuit, design reviews will also help ensure that you are never the only one who knows how your project should work.

- Conduct design reviews on every major revision of your system. This means bringing in another person to discuss your circuit. Keep a record of what was discussed and decided in the review. When presenting a new revision, indicate what has been changed since the last review.
- Design reviews should be thorough. Even innocent looking sections should be discussed. Chances are you won't find a bug there, but it will save you pain down the line. Discuss failure modes and alternative solutions.
- The person doing your review should know what they're doing.
- If appropriate, invite junior members to listen in on the review. They will benefit from being part of the review process and will become better designers and reviewers in the future.