

Qubo Electrical System

Interface Control Document

Ben Hurwitz



August 7, 2017

RoboSub 2017: Submarine Hackathon

RoboNation

Department of Electrical and Computer Engineering

University of Maryland

Contents

1	Overview	2
2	Details	2
3	Extras	2
4	Moving Forward	3

List of Figures

List of Tables

1 Overview

RoboSub 2017 came in a hurry, with little electrical working properly. The charging system lacked testing, the ORing chips had not demonstrated full capabilities, and the LT8640 5V/12V rail generators were completely unresponsive. This is to say that there was justification for the decision to scrap the charging and the LT8640s entirely for the far more reliable (and better known) PTN78020, the wide-range input 6A DC/DC buck converter made by TI that is controlled by a single resistor. It was a changed intended to make things simpler and improve robustness with known quantities. Instead, it pushed us (and by us, I mean me) towards to panic-region. In keeping with the Robotics At Maryland (RAM, or R@M) spirit, when we got to competition, none of the boards that were developed worked in any meaningful way, and thus we were forced to decide: spend time debugging, which was not something we are good at, or spend time further simplifying and hack it to get exactly the minimum of what was needed. By the end of day two, we had opted for the latter.

The resulting system was close to as barebones as could be, and by the time we actually went in and completed our qualifying run, the system was, in my opinion, the most minimalist system possible for our configuration. We had removed nearly all the sensors entirely, including the AHRS, depth, and internal temperature, choosing to rely simply on our mechanical stability and dead-reckoning to get us through. Thermal considerations were lost, or left to passive cooling options. The power control modules for the batteries were jettisoned for simple SPST switches and 30A fuses. Even the PCA9685 chips that we were using for PWM signalling were tossed aside. But in it's simplicity, Qubo worked for the first time, and qualification was achieved, so in this there was success where there had not been before. The system cuts out all the fat from previous iterations and reduces the entire electrical subsystem to a half dozen major components, including all voltage generation, that allowed Qubo to move forward.

2 Details

I have no images to show you at all, so if you'd like a more hands on view of the system, we should facetime or skype or whatever you kids use these days to go through it. The system runs off of a single battery, which was triggered by a simple SPST rocker switch to avoid the issues associated with plugging live power into other components. This 14.8V nominal goes through a single PTN, which is resistor-set (2.61k Ω) to output about 10V. (Why 10V? Because the PTN needs a 3V cushion to do it's converting, and a 12V output would require a minimum of 15V, not really feasible with this.) This 10V powers the Jetson (which takes down to 7V), and through the magnetic reed switch, triggers the main thruster relay. There are a number of capacitors on the input to the PTN - **DO NOT REMOVE THESE!!** Each of the ESCs on the output of the relay has a 220 μ F capacitor on it, which draws a large inrush current when voltage is applied. This causes significant voltage drop down to nearly 5V of the input source (batteries or shore), which kills the PTN, and thus kills the Jetson. The extra caps at the input of the PTN (notice how close they are) prevent this droop from being seen by the PTN, thus keeping it in operation. The ESCs themselves are mostly the same, except for the PWM signal generator. The PCA chip shorted, and we didn't have a backup, so we switched the embedded Arduino Uno to a TeensyDuino (TD), which has a dozen or so PWM outputs built-in. The TD is connected to the Jetson via a USB cable, which also power it. This gives us the ability to easily generate PWM signals directly from the embedded system. The only other thing is a single NCP1117 that converts the battery/shore voltage into 5V for the ethernet switch.

That, in essence, is the entire system. A single 10V rail powers the Jetson, which powers the embedded, and switches the thrusters on and off. Viola.

3 Extras

There are so many wires and so many parts that are still there but left unused or unconnected that it's hard to see this simplicity. The terminal block is used mostly to connect all the ESCs to the same high-voltage (V_Batt) input, and to connect their outputs to the same ground. A second NCP1117 is left hanging and disconnected; it was used for some other bits that got summarily culled from the robot. There is a depth sensor which we believe (**but are not sure**) shorted at some point, thus rendering its data worthless. The AHRS works, but it's not reliable with all the changing currents within the pressure vessel. There are loose wires from two fans that were initially part of the vehicle as well - they are 3-wire black ribbon cables. The I2C line was removed entirely because of the removal of all associated devices, but the wires still remain. All

the PWMs from the ESCs are bundled together into the TD; the signal and power grounds are tied together for simplicity; and the power and ground lines go right into the terminal block.

4 Moving Forward

To be frank, the system looks like shit. It needs a good cleaning up. But I cannot emphasize enough that **it works**. Do not take this for granted! It is not easy to get all of this working, and tearing it down to make it look pretty will put a lot of pressure on you to build it back up again. It was tricky enough once. Use it as a platform to continue forward with, to build off of. Personally, I would use this next year to design small boards for each of the primary components (5V generator (PTN), 10V generator (PTN), TD), thoroughly test these, then combine them into a larger board with the necessary inputs and outputs. I would spend the first semester doing this; they're also smaller projects that can be given to new students without too much risk since they're all only a half dozen small components each at most. Make sure that the ESCs can more easily attach to the board! It should be a decent project for you. Pay close attention to how the wiring will be laid out! A major disappointment was not taking the wiring into enough consideration. It's a major size, weight, and obnoxious consideration. Take some time to think about how the wires should lay out, and discuss with Peter. Try to pull together mock-ups if you can, it'll help to visualize where things should be.

If you get a nice single board that houses the single input, the 5V/10V generators, and the TD, and it has been tested thoroughly, then there are a number of small additional projects that can be developed. Add back in the depth sensor first. A single 5V I2C line **should** suffice, even though the Bar30 asks for a 3.3V I2C line (this has been borne out in other testing, but you should verify! Collaborate with the software team on building a small jig to test this). The AHRS connects directly to the Jetson, so this is not a concern of yours, but something to keep in mind is shielding it from stray magnetic fields - use your PHYS271 and ENEE380/1 knowledge for this! (Moving charges generate magnetic fields.) Add some internal sensors - I'm thinking temperature in particular - using the same method: build a small component-specific board, test it completely, then integrate it into the larger board. Current and voltage sensors would also be nice, but those are something that a new EE can do without being critical to the system.

My sense is that the critical parts on a single board can take a semester. For the second semester, focus on less critical, perhaps more long terms components. For this, you might consider integrating the DVL. We have the UWE isolated regulator already, and you can work through the noise reduction stuff while the other EEs develop the current sensor + fuse circuit we had previously for each of the thruster outputs. By next year's competition, you should be shooting for a very robust power system that is essentially a cleaner version of what I hacked together. Do not wait! Starting too late was a huge failure on my part. Get boards out to fab within a few weeks of starting, and continuously iterate. Note that for small boards, OshPark is fantastic (takes a bit longer, but only \$5 a square inch), while larger boards can be made with Advanced Circuit's \$33 deal. New EEs are developing breakout board for certain components. BUT the best thing you can do to make everyone's lives easier is to learn to use Design Blocks in Eagle. These will allow you to easily move a design from a smaller board to a larger one without any extra work.

I won't tell you what to do, only suggest. But I believe it would be a colossal mistake to throw away the system as it is now to redesign again. Use what you have to test, and develop slowly. You have a nice foundation; don't throw it away because it looks like crap. Slack me if you have questions and I can try to help. Good luck with it - think of it as a bundle of small parts, and it becomes very doable.