Carnegie Mellon University

16-681A

MRSD Project I

Individual Lab Report 04
Team C: COBORG

Author: Gerald D'Ascoli

Remaining Team C Members:
Jonathan Lord-Fonda | Yuqing Qin | Husam Wadi | Feng Xiang

Sponsor:
Biorobotics Lab

April 1, 2021

Table of Contents

1 Individual Progress

Most of my individual progress for this progress review has been centered around the development of the CoBorg Power Distribution PCB. I designed the schematic built around 2 input ports, one for power from the on-board batteries and one for power input from an external source (typically a wall outlet), to control switching power supply input between the external power source and the batteries with priority to the external power source. This switching is controlled by the LTC4412HV power switcher which senses if the output line is powered, and if not opens the channel of Q1, a p-channel MOSFET, to power the node with the battery power supply. The main power node goes to an output that powers the HEBI motors on the CoBorg at the input 36V and goes to the two voltage regulators, one for 5VDC and one variable DC-DC converter. The 5VDC output goes to power logic systems and our temporarily installed onboard router. The variable DC-DC converter is set to ~19.25VDC by design with a 33.3kΩ trim resistor. The equation relating V_{out} and R_{trim} for the JRCS016 variable DC-DC converter is $R_{trim} = \frac{700-(10*V_{out})}{(V_{out}-4)} k\Omega$ so a desired output of 19.5VDC would require trim resistor of 32.58kΩ. Rounding to the nearest standardized resistor size of 33.3kΩ puts the output at 19.25VDC which works within the system specifications. This output powers our on-board computer and is therefore rated to take up to 15A to manage any increased processing or peripheral device current. The variable DC voltage is helpful in case we need to change our on-board computer and the new device requires a different power input voltage. Both import ports have overcurrent protection in the form of 20A fuses and overvoltage protection in the form of TVSs with breakdown voltages of 39V. All output ports have overcurrent protection in the form of fuses: 10A for the motors, 15A for the computer, and 5A for the logic/router. Every port has indicator LEDs to confirm node health. This detailed schematic is shown in Figure 1.

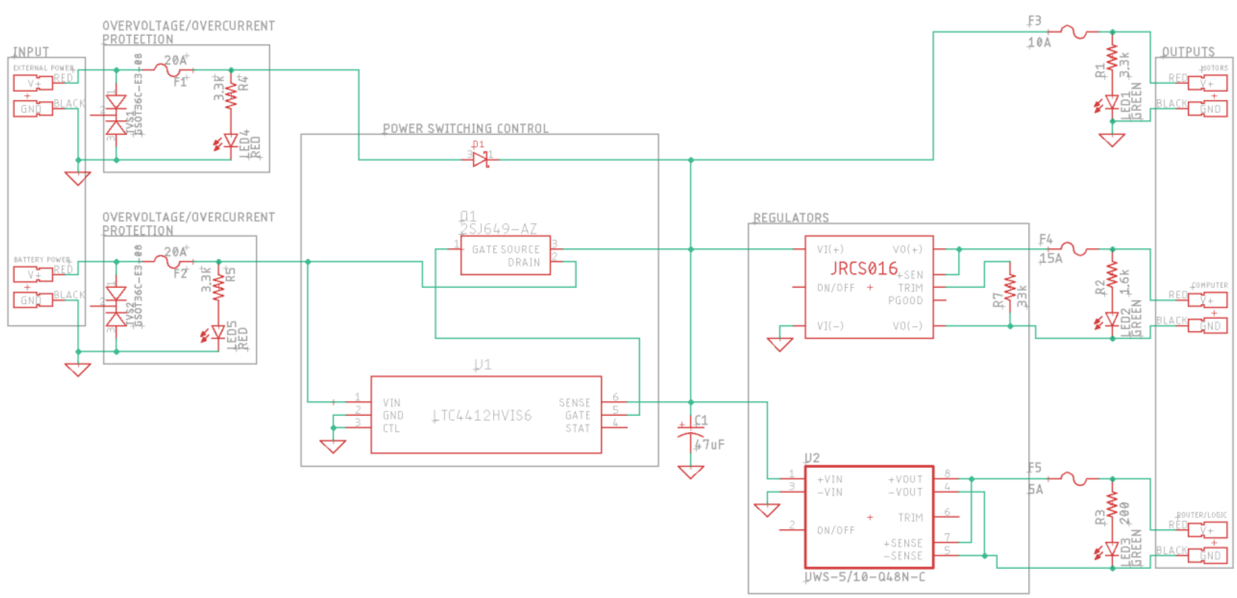

Figure 1. CoBorg Power Distribution PCB Schematic

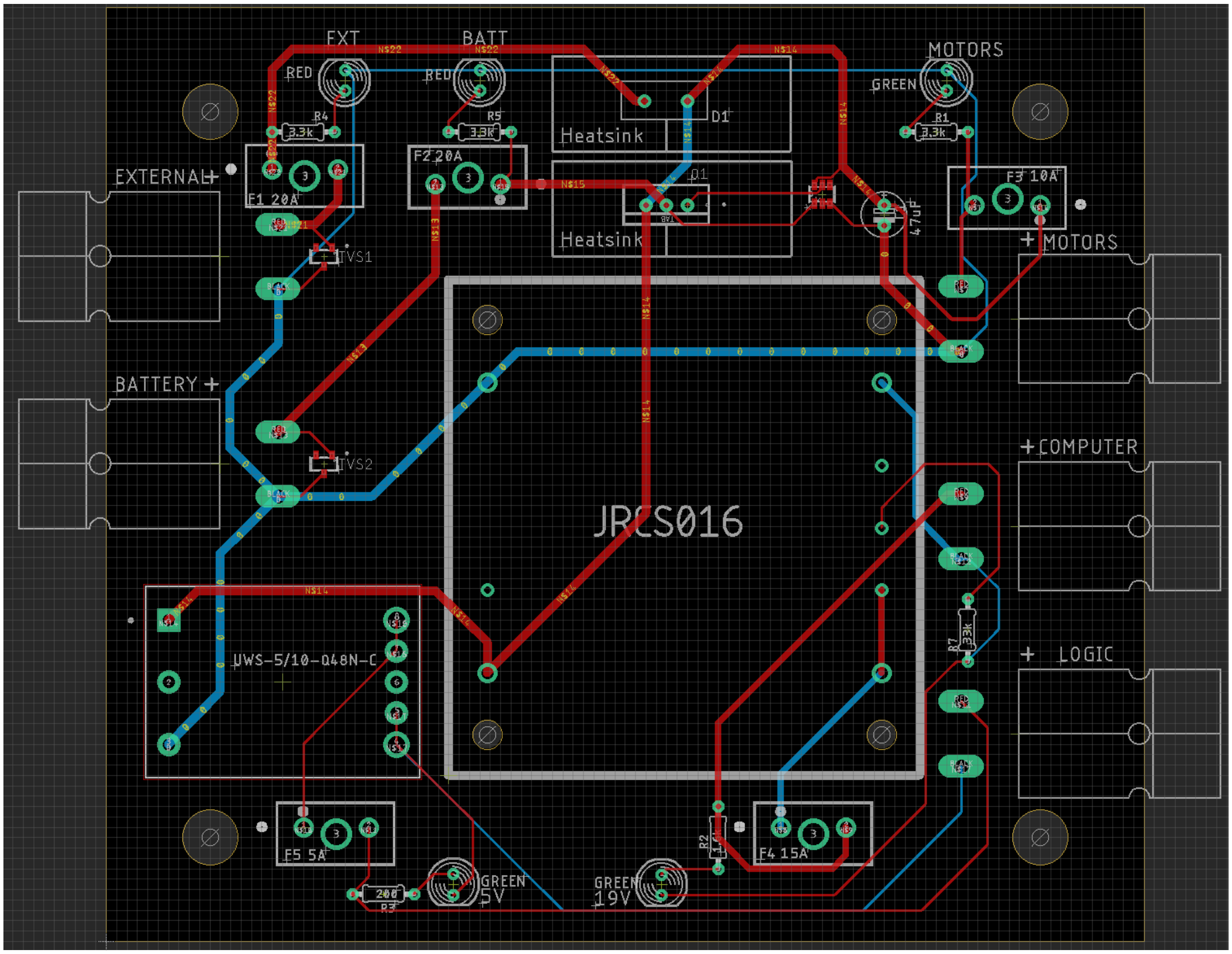

Figure 2. CoBorg Power Distribution PCB Layout

The PCB board shown in Figure 2 was built with 25mil thick top and bottom copper layers with a 25mil core in-between. These thick layers were necessary to handle up to 20A current at worst case scenario while also keeping the trace widths narrow enough to route on a reasonably size board to fit within our CoBorg case (10”x7”x4”). The final board is 6.129”x4.648”x0.075”. To properly test, we will need an external power supply AC-to-DC converter that outputs 36VDC and one or more 36V batteries. All should be in supply in the Biorobotics lab and available for our use.

For the voice subsystem, since the last progress review, I worked with Jonathan to run through his draft of the subsystem validation plan. We discussed criteria for failure modes and formats for the test that will be conducted during the Spring Validation Demonstration. One issue this test brought up more significantly is the excessive frequency of false positive triggers on our keyword “COBORG” and our e-stop keyword “STOP STOP STOP”. Jonathan and I had decided that the failure mode for false positives only occurs if an actual command signal was sent to the main_state_node meaning a valid command was recognized. This would not occur if just “COBORG” was recognized but would occur if the e-stop keyword was falsely recognized, which posed a significant issue. Husam sat down with me to talk over the function of the code in the voice_recog node, and we brainstormed ways to filter out false positives. I found functionality in Pocketsphinx that allows for placing thresholds on keywords when in the keyword search mode. With this new functionality, I modified how the e-stop keyword was recognized to drastically reduce the false positives while also strengthening the recognition of true positives. For the “COBORG” trigger keyword, it weakened the recognition slightly but also drastically reduced the false positives so the system is overall improved.

2 Challenges

The main challenge for the PCB was finding suitable parts in stock on Digi-key. There were no DC-DC converters that convert to the PC’s voltage of 19.5VDC from the 36V power supply so we had to find a variable converter that operated with 19.5VDC in its possible range and design the control circuitry for it. This part was unfortunately significantly more expensive, but the variability of the output will be very helpful for the related output as previously stated. The other significantly rare part was the 20A fuse rated for over 36V. Most standard fuses rated for 250V would not reach up to 20A and the other form rated for 20A+ was only rated for up to 20V. Husam did some digging and found suitable fuses of a larger form factor so the PCB layout had to update with these new models.

It was a challenge to solve the false positive issue with the voice recognition system. Husam and I discussed possibilities like setting an amplitude filter to try to reduce some of the background noise getting recognized as words, then we found out the Pocketsphinx has built in background noise cancellation, so we assumed that was not the issue. Before I had tried the solution to add more words to the keyword dictionary so that they would be recognized separate from the “COBORG” keyword, but in our research we discovered that this was actually amplifying the problem by having the system recognize more words and bog down the processing for the keyword. We tried re-formatting how the keyword was processed and deep-dove into the Pocketsphinx functionality to see if there was something fundamental we could change. After about an hour diving into the background functionality, we discovered the keyword thresholding. I then tuned the threshold value to each keyword and initial testing showed dramatic improvement to reduce false positives and therefore significantly reduce the frustration factor dealing with the system without real sacrifice of the command recognition.

3 Teamwork

Team Member	Teamwork Progress	Challenges	Future Work
Feng Xiang	-Created ROS Node to publish goal positions every second to MoveIt -Move robot arm relative goal pose to D435 camera link frame	-Adjusting the time restriction on the path planning node to be able to find a path quickly -Measuring transform between robot frame and camera frames	-Implement OMPL+CHOMP motion planner for a more iterative updating motion planning model -Develop 3D goal pose to update stabilizer robot arm relative to global odom frame
Jonathan Lord-Fonda	-Wrote SVD/FVD 1-pagers -Added the speaker to requirements and validation plans -Checked in with Gerry, Jason, and Yuqing to ensure validation plans still matched voice, actuated manipulation, and vision subsystems and updated validation plans -Began setting up validation testing -Ran through voice validation with Gerry -Ran through strength test	-How to define accuracy for voice subsystem -Problems with strength validation	-Research the other Elastic Bands papers -Research and implement CHOMP (or Elastic Bands) -Run through validation testing -Update and finalize validation plans -Work with team to integrate ROS nodes
Gerry D'Ascoli	- Designed power distribution PCB layout - Re-evaluated parts with Husam based on system requirements and Luis's recommendations - Ran trial runs of Jonathan's validation tests for the voice subsystem. - Fixed major issue with Voice Subsystem having too many false positives	- Figuring out a method to reduce the sensitivity of the voice command trigger word while increasing the sensitivity for our set of valid commands.	- Establishing criteria for validation procedure for the voice subsystem - Validating that requirements are met under aforementioned criteria - Possibly improving the accuracy of the command set (post-keyword recognition) if requirements aren't met or use triggers frustration in new user(s). - Re-implementing the voice node on the new NVIDIA Jetson Xavier - Install components on PCB once it is delivered

Yuqing Qin	- Implemented the postprocessing node (goal_getter) - Implemented the surface normal - Set up the validation environment	- Make sure the accuracy of 3d position - Run time restriction on the Zotac	- Measure the accuracy of vision system - Further improve the accuracy of averaged 3d position
Husam Wadi	-Created launch files for main node and voice node -Assisted Gerry with removing false positives in voice subsystem -Assisted Gerry with PCB design and refinement	-Creating a timing service that keeps track of how long it takes for a voice input to translate into a command and how long the vision subsystem takes to recognize and publish the goal point	-Tie in the voice subsystem to the vision subsystem through the main node

4 Plans

For the CoBorg PCB, we have to install all of the components on the board and test to make sure the power switching and power converting is all functional for our system requirements. This will involve soldering the through-hole and surface-mount components to the board which could be a difficult and strenuous task. Testing will require an external 36V power supply and one or more of our 36V batteries as well as some form of a multimeter to confirm the voltages at the input and output ports.

Upcoming plans for the CoBorg voice subsystem mostly involve developing and testing the validation plan with Jonathan as the base functionality of the voice subsystem is complete. There is some potential improvements in the realm of improving the recognition of post-keyword commands which I would like to look into, but as far as the SVD is concerned there is only testing left.