CORNELL MARS ROVER

**Dean Reiter**

**Luke Griffiths**

**Nivant Desai**

**Esther In**

Controls Electrical

# *BLDC Hardware Final Report*

# I. Overview

This report discusses the design, testing, and performance of the BLDC hardware on the rover. The BLDC hardware is responsible for driving and controlling any brushless DC motor. In particular, this report discusses the BLDC hardware with regards to the drives system. We discuss the performance of the hardware before and during SAR. As a result of the SAR failures, we put forth our testing methodology and explain the results of it, including the necessary redesigns that will happen after the Spring 2022 semester. It is instructive to note that none of the designs for any BLDC hardware was created by the current members of this team. As a result, a lot of our work revolved around understanding the system that was created by past members.

# II. Pre-SAR Design and Testing

## i. BLDC Hardware Overview

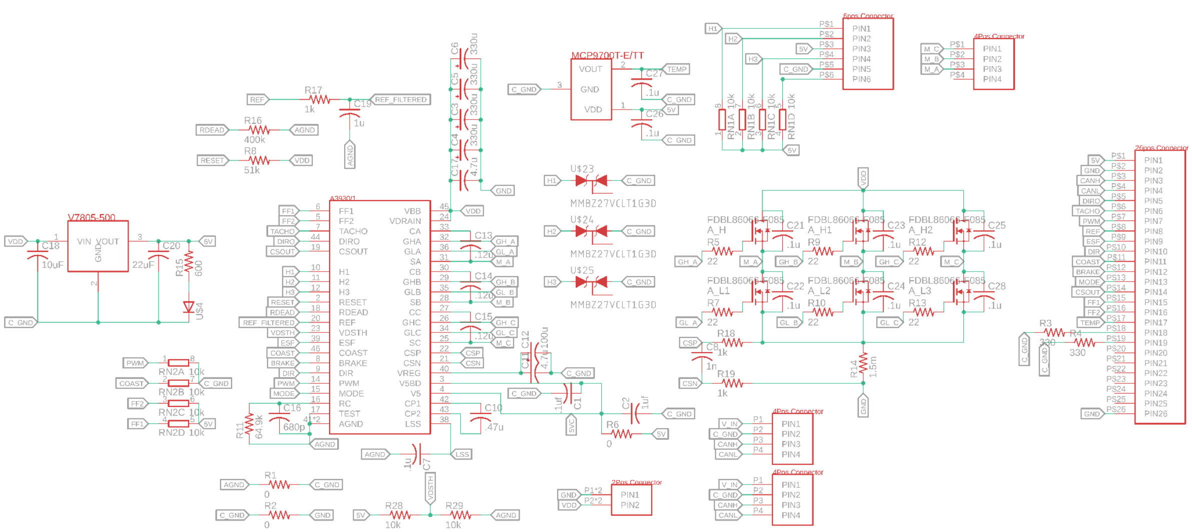
The 2022 rover's drive system was designed with a rocker-bogie suspension system and six wheels, each spun by a brushless DC (BLDC) electric motor. To support this drive system, we powered and controlled each motor with a Distributed Task Board (DTB), which combined two circuit boards, a power board and a logic board. The BLDC DTB Logic Board is designed to receive motor control commands from the Central Communication Board (CCB) via a Controller Area Network (CAN) message protocol. The Logic Board then translates these commands and sends them to the BLDC DTB Power Board. The Power Board then powers a BLDC motor using those control signals and also retrieves sensor readings from the motor to send back to the Logic Board.

## ii. Design Choices

Our initial plan this year was to use the 2020-21 designs for the BLDC DTB Logic and Power boards.

The 2020-21 Logic Board was designed for the additional purpose of wirelessly debugging the CAN bus. The 2020-21 Logic Board interfaces with the DTB Power Board and with the CCB via a CAN connection, and it receives 3.3V power from the Power Board. The 2020-21 Logic Board includes a PIC microcontroller for logic.

The 2020-21 Power Board was a totally new design from the previous iteration, combining features from the previous year's Drives Power Board and DTB Power Board. The 2020-21 Power Board receives power from the battery, interfaces with the Logic Board, includes both BLDC power and hall effect sensor connectors, and uses an A3930 predriver and an N-FET H-bridge to send out 3-phase control signals to the motor.



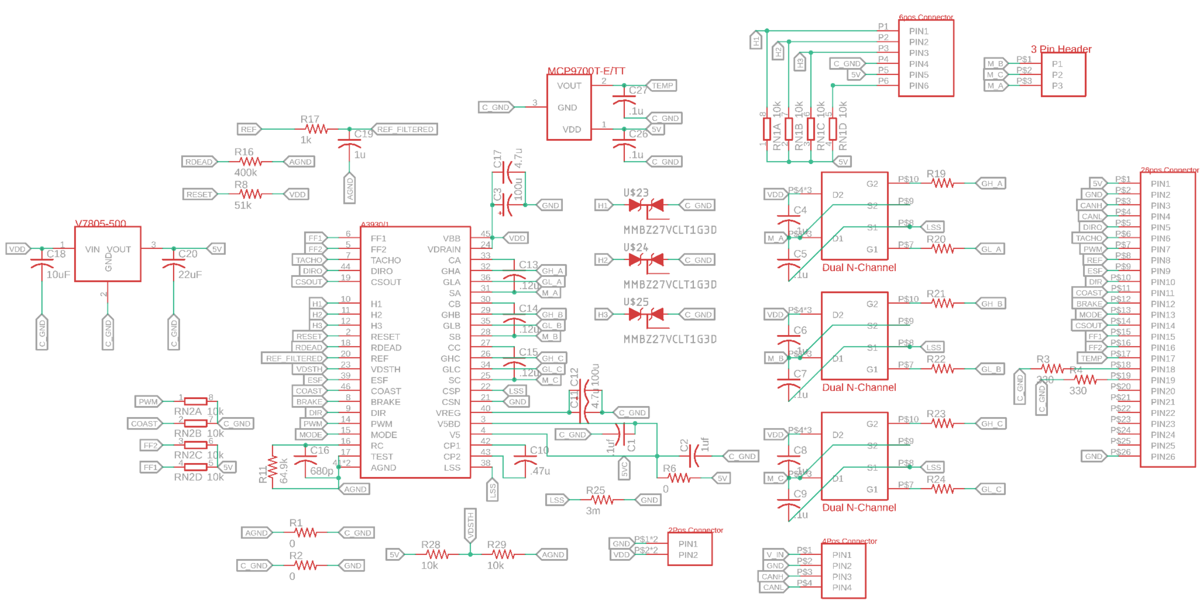
2020-21 BLDC DTB Power Board Schematic

Unfortunately, several errors were found in the 2020-21 Power Board's schematic and several issues arose during testing that prevented the board from functioning. The boards were manufactured with pads where there should have been through-holes, the design included a possible short-circuit, the board experienced power issues, and its firmware was not functional.

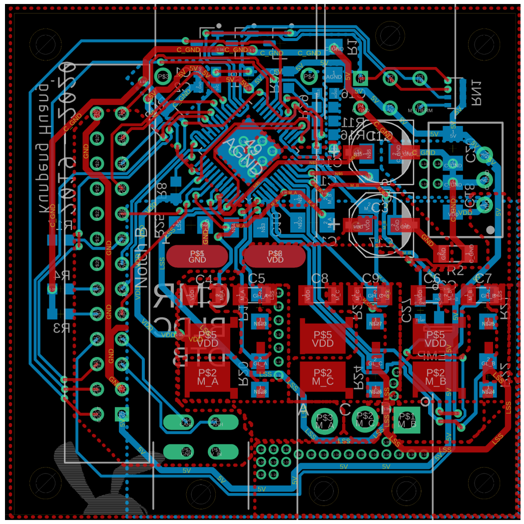
Due to the issues with the 2020-21 Power Board, we decided to discard the newer board designs and use the 2019-20 DTB Power Board and Logic Board designs instead. The 2019-20 DTB designs have been used on previous rovers successfully, so we figured that this would be a reliable solution.

The 2019-20 Logic Board is simpler than the following year's but still receives CAN commands from the CCB and interfaces with the Power Board to send it motor commands. The Logic Board uses a PIC microcontroller for logical processing.

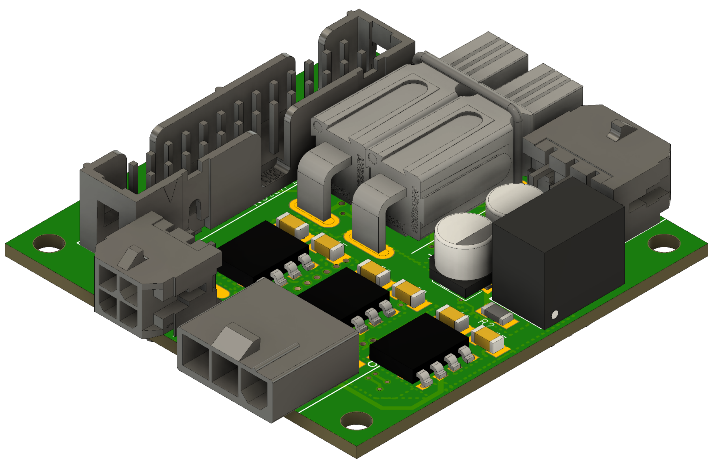
The 2019-20 Power Board also receives power from the battery, communicates with the Logic Board to receive motor commands and send back sensor readings, has an A3930 predriver and N-FET array H-bridge to send out motor signals, and has connectors to power the BLDC motor and to connect to its hall effect sensor. The Power Board also includes a 24V-5V regulator, and the A3930 predriver is equipped with diagnostic features and outputs to read current and velocity.




2019-20 BLDC DTB Power Board Schematic



2019-20 BLDC DTB Power Board Layout



2019-20 BLDC DTB Power Board 3D Model

## iii. Hardware Testing

Before testing our BLDC DTB boards, we first tested that our BLDC motors functioned themselves using an off-the-shelf motor controller known as ODrive.

We then tested controlling and powering BLDC motors using the 2019-20 DTB Power and Logic boards with commands sent using PCAN software. We successfully powered and controlled our BLDC motors using the 2019-20 DTB, but the motors were only tested on the bench, without significant load, and unconnected to gearboxes. The only slight issue we found during testing was that when we powered the BLDCs at less than 100 percent effort values, they would sometimes require a physical nudge to start moving.

# III. Performance During SAR

## i. Hardware Setup

At the time of SAR, we had only manufactured four BLDC DTB Power Boards, so we were only able to test and use four out of the six drive motors. Also we were unable to correctly mount some of the wheels.

## ii. Real-World Performance and Testing

After integrating and wiring all the necessary electronics on the rover in preparation for SAR, we began testing our BLDC hardware and planned to build up our tests until we could drive the rover on the ground with all four motors engaged.

We began by testing the BLDC hardware on the table. We lifted each wheel off the table and tested each motor individually, so that the only load on the motors was the minimal internal friction in the wheels and gearboxes. We were successfully able to spin each motor, but there were some small issues. The BLDC motors would spin at different speeds in the forward versus the backward direction, and we sometimes had to physically push the wheels to get them to move when the motors were given low effort values.

We then tested our BLDC hardware with the rover on the flat ground, now under high load. Due to issues with our radio communication system, it is worth noting that we could only send start and stop commands to one motor at a time. We would still be able to activate all four motors at once, but they would start sequentially. We also tried controlling motors directly using CMR serial or PCAN. The rover was able to drive somewhat successfully on flat ground but had issues where some motors would stall. The motors mostly seemed to struggle to overcome static friction and start moving, so we often had to pick up the rover and start the motors off the ground, before placing the rover back on the ground so that it could drive.

We then tested our BLDC hardware with the rover driving up a slight incline, which put the motors under very high load. When the rover began to ascend the incline, several of the drive motors stalled and three of them burned out. This was because the motors sustained a high level of current without being able to turn. This resulted in the coils inside the motor to melt. The increased load was clearly too much for our BLDC hardware to handle and fixing this issue required immediate, serious attention.

After damaging a few of our drive motors, we replaced them with other motors and tried to address the stalling issue. For instance, we tried rewiring the BLDC motors in several different configurations to see if it could be a wiring issue, but this did not solve anything. It was clear to us that the issue was design related, and much of our driving methodology would need to be carefully studied and possibly redesigned.

## iii. Summary

The hardware's performance during SAR told us that there were several issues we would need to address. Although we were able to power and control our BLDC motors to an extent, they would frequently stall and could not perform under high load. Since the issue was likely not due to how our hardware was wired or integrated on the rover, we deduced that the issues must lie in the BLDC hardware or firmware.

# IV. Post-SAR Testing and Analysis

## i. Goals

After SAR, our goal was to figure out what caused our BLDC motors to stall. We wanted to isolate the causes and determine whether they are mechanical or electrical issues.

## ii. Testing Plan

Our overall plan of hardware testing was to go through every step of how our BLDC motors are implemented, from the bench to the rover, and isolate issues along the way.

First, we planned to test the BLDC hardware on the electrical bench. We would first test a known working motor with known working control methods (such as PCAN and ODrive) and measure current from line-to-line AC voltage between phases. We would also plan to measure hall effect sensor data and note the motor's mechanical behavior. The goal of this step was to analyze a healthy BLDC motor's behavior and power/signal requirements.

Second, we planned to test all four motors connected to four sets of DTB Logic and Power Boards on one CAN line. We would power the setup via battery, send motor commands using PCAN, and run the motors with no load. We would examine current and power measurements and record any other observations about how the hardware behaves. The goals of this step is to test the reliability of our CAN system and logic hardware.

Third, we planned to run similar tests as the last setup but just on one BLDC motor, now in a gearbox. We would again measure current and power draw and observe other motor behaviors. We would initially test running the motor at 100 percent power and then decrease the effort value we send them until the motor starts to stall. The goal of this step is to examine how the gearboxes and a slight mechanical load affects BLDC motor performance.

Fourth, we planned to repeat the second step, testing all four motors with our DTB hardware and one CAN line, but with the motors all inside gearboxes. We would add one motor to the system at a time and test each step of the way to isolate any potential variables and issues. We would continue to test the functionality of each motor in every way possible.

Fifth, we planned to test one motor with a gearbox and wheel attached. We would test how adjusting the mounting of the wheel, such as changing screw tightness or adding grease, could affect motor performance. We would again record observations and measure current and power draw.

Lastly, we planned to test all four motors running with gearboxes and wheels. We would first test the system running on the bench by picking each wheel up and testing them individually. Then, we could test all four motors running under load by driving the rover on the ground.

## iii. BLDC Motor Testing

After SAR, we first determined which BLDC motors were working, damaged, or dead. We tested for dead motors and found four working and three dead. We used the following method to test for dead motors: We measured the frequency between all the phases of a motor. We attached a drill to the motor shaft and turned the motor using the drill. We measured AC frequency and voltage across phases using a multimeter. If we measure different frequencies or voltages across different phases of the motor, then we know that the motor is broken.

In our initial tests, it became very clear that when our motors stall and stop moving, they still receive current, which is likely what caused the motors to burn out. We measured current and power draw using a multimeter. We tried measuring AC signals using the oscilloscope but found it difficult to get good data.

In order to read hall effect sensor data from our BLDC motors, we set up a Raspberry Pi and wrote a Python script to read, interpret, and verify hall states. When just rotating the motors by hand, we observed very strange behavior from the hall effect sensors. We noticed that when rotating a motor forward, the hall effect sensor skips the state transition from six to one, and when rotating a motor backward, the hall effect sensor skips various hall states seemingly at random. We thought that the motor skipping or returning invalid hall states could cause the predriver to respond inconsistently and contribute to their physical “crunchiness.”

When we tested a BLDC motor inside a gearbox, we found that the gearbox created a lot of resistance and the motor needed some physical assistance to start turning. Using lubrication or modifying the gearboxes did not significantly affect motor performance. We observed that when the motor failed to immediately start, it only drew a quarter of the current we would expect it to draw.

After some mechanical modifications were made to our gearboxes, we again tested them with our BLDC motors. For this particular test, we ran the motor forward and backward at 25 percent effort value and measured line-to-line voltage. The goal of this test was to determine if loading the motors affected line-to-line voltages. We also found with most of our gearboxes, our motors performed similarly to how they performed without load. Therefore, most of our gearboxes (with the exception of one bad gearbox) did not put significant frictional load on our motors. On the negative side, however, we found that the motor’s line-to-line voltage when moving backward was lower than when moving forward. This is an issue because line-to-line voltages should not be affected by which direction the motor is spinning.

After testing our BLDC motors with an ODrive motor controller (a reliable, off-the-shelf motor controller), we made some breakthrough discoveries. When controlled by ODrive, we did *not* measure a discrepancy in BLDC line-to-line voltages between forward and reverse rotation. This discovery told us that a fault must lie in our power and control methodology, either in our hardware or firmware. We ruled out the fault being in the motors themselves, and began searching for issues in our BLDC hardware and firmware.

We later continued to test our motors with different gearboxes to see how the mechanical differences between them affected motor performance. Also, we began working on a way to interface ODrive with a Raspberry Pi to allow us drive multiple motors at once.

## v. Board and Firmware Analysis

After analyzing the 2019-20 BLDC DTB Power Board schematic, a few major design flaws were discovered which are likely causing many of our BLDC issues (see PDF). First, the A3930 predriver on our board cannot handle hall sensor faults with software. The A3930 controller will attempt to stop the motor if it sees an invalid hall state. This will result in sporadic motor behavior (since the motor does not always output valid states). Second, our board includes a 5V voltage regulator which incorrectly sends a 5V output to a pin on the A3930. That pin on the A3930 is itself a 5V output, however, so “it is like two giant trains being driven headfirst into each other on the same track.” Third, a pull-up resistor needs to be added for the MOSFET on the interval t_dead. The “dead time” is the amount of time between MOSFET on and off events. It is important to have a non-zero dead time in order to prevent cross-conduction. In other words, we want all of the signals to be able to “flush out” before the MOSFET is turned on again. This leads to cleaner signal propagation through the circuit.

After looking at the BLDC DTB firmware, it is clear that the code was poorly written and lacks documentation. It is very likely that the firmware has faults of its own.

## vi. Takeaways

- Meetings with maxon
- Gearbox issues
- Board issues
- Firmware issues
- Predriver issues - A3930/31

# V. System Redesign

## i. Requirements

The BLDC DTB Power Board will need to be redesigned for next year's rover. In addition to correcting the schematic errors described earlier, we will need to replace the A3930 predriver with the A3931 predriver. The latter will be able to handle invalid hall states without jeopardizing other functions. The A3931 will not attempt to stop the motor when it sees an invalid hall state, unlike the A3930. This makes the A3931 much more suited to our purposes.

The BLDC firmware will also need to be largely rewritten for next year and should be better documented and organized.

## ii. Design Timeline

Our hopeful goal is to complete at least the schematic of the redesigned BLDC Power Board before May 29, 2022. We plan to complete the schematic and layout by the start of the Fall 2022 semester at the latest.

## iv. Expected Risks

The only risk associated with this redesign is that we may not get the new boards manufactured in time due to global part shortages and supply chain issues. However, we have no choice but to try to order the new boards as soon as possible because the old boards simply do not work.

## v. Summary: Moving Forward

The most important thing that needs to happen next is the redesign for the BLDC power board. Making this board in an intelligent and specialized way for our purposes is essential in order to isolate the electrical system as a point of failure. We will likely take other factors, such as power efficiency and timing, into consideration in the new design. We would like to emphasize that the BLDC redesign is not the only thing that will determine the workability of the drives system. A BLDC board that attempts to drive a mechanical system with many errors will be just as problematic as the current system. As a result, we must ensure that ALL parts of the drive system work in harmony. It will be important to communicate with the drives team to ensure that all development occurs intelligently.

# VI. Appendix

**i. Links to supporting documentation**

**Confluence links:**

2019-20 BLDC DTB Power Board Documentation

**Misc. links:**

Post-SAR BLDC Testing Plan

Gearbox Test Data

2019-20 BLDC Power Board Schematic Issues