ET NavSwarm Write-up

Nick Snyder

Motivation and Background

ET NavSwarm is an interdisciplinary group with the end goal of sending a swarm of small autonomous rovers to another planet to prospect for materials. The group is now implementing the particle swarm algorithm to all the rovers to communicate with each other. Some areas for improvements the group have identified are active battery monitoring and hardware longevity.

When testing on Boulder Field it's hard for the group to monitor the states of charge of all the different rovers. This leads to batteries being over discharged and ruined. A way to clearly indicate when a battery's voltage drains too low and eventually turning off the rover before the battery dies would be very helpful. This would also reduce the waste created by bad lithium batteries. A problem is already appearing in the lab.

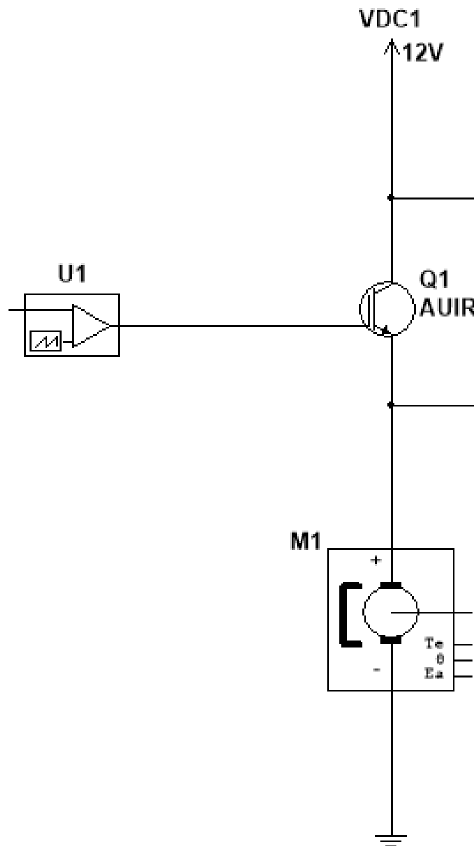
The hardware failures usually stem from the motor controllers. The cause of this is most likely the low-quality components on the board. The most recent design specification uses two HW-039 H-Bridge boards for brushed motor control. One for each side of the rover. The HW-039s are widely known to be unreliable and fail seemingly randomly. Designing a high-quality controller with an IGBT would help the longevity of the rovers tremendously.

What I hope to learn from completing this project is PCB design and real-world applications of circuits and microcontrollers. PCB design is not something taught in the curriculum but is very valuable to a professional electrical/computer engineer. Embedded circuit design with microcontrollers is another necessary skill to be an effective computer engineer.

Circuit Design

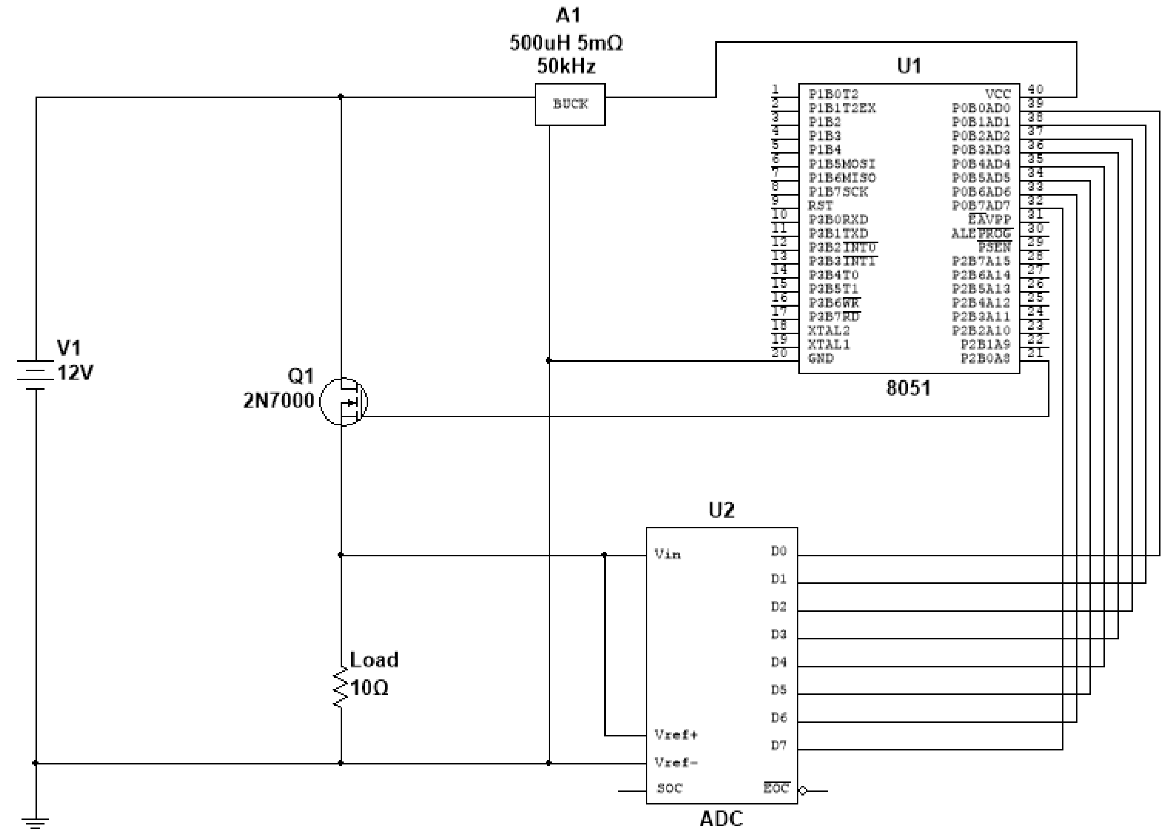
Motor Control

My design for the motor controller does away with an H-bridge. This is because the custom airframe loaded into the Pixhawk by QGroundControl does not have capabilities for reverse thrust. The main reason to use an H-bridge is to reverse the motor direction meaning the simplification brought by my design would not lose any functionality. The Pixhawk sends out a PWM signal from its I/O ports that in this design is connected to the base of each IGBT. When a pulse is high the base closes the circuit allowing current to flow.



Battery monitor

For my battery monitor circuit, I'm looking to turn off the circuit by having the load of in series with a FET that can open the circuit when the low voltage cutoff is reached. I plan on using a microcontroller to control when to switch off the circuit at the threshold as well as show the current voltage on a display. I will need to use a buck converter to step the battery voltage down to the voltage needed by the microcontroller and an ADC to convert the battery voltage to an 8-bit digital signal that the microcontroller can use. The current battery voltage would also be displayed on an array of 7-segment displays for an exact readout to the user. When the low voltage cutoff is reached the circuit would enter a low power mode where only the Pixhawk and a red flashing LED would be powered on to alert users from afar or at the ground station.



Testing Plan

To test the motor controller, I would try to replicate the behavior of a known good off the shelf motor controller or a HW-039 as well as testing the safe range of voltages and currents. I could do this by connecting it up to a benchtop power supply and testing it with varying voltages.

To test the battery monitor, I would replace the battery with a benchtop power supply and measure the current flowing while varying the voltage. This is to make sure the microcontroller is turning off the FET when it needs to.

Fabrication Plan

To create a PCB layout, I would use software like KiCad a free to use software suite that can create Gerber files, bill of materials, and 3D models of a PCB.

To fabricate the board, I will use JLCPCB. JLCPCB can fabricate boards up to twenty layers and has reasonable build times with good customization options.

Timeline

Oct-Nov-Dec: Design schematic

I hope to have the completed schematics for both the controller and battery monitor by the end of the semester.

Dec-Jan: Fabrication

This stage consists of designing the traces of the board in KiCad and sending it out to be fabricated by JLCPCB.

Jan-Feb: Testing/programming

This is the time it would take to verify the designs and program the microcontroller.

Feb-Mar: Revision

If any designs don't work as intended from the testing phase, this is the time where I would revise the PCB design and/or microcontroller.

Apr: Conference preparation

This is when I would compile my results and create my poster.

Projected costs

The budget would only need to go toward the fabrication of the PCB. If revisions are made, I would need to fabricate another board. Since I don't know the number of layers or the exact components, I'll be using I cannot state an exact price for the project, but I think it's safe to say it won't be over budget.