Automated Lithium Ion Battery Characterizer

DESIGN DOCUMENT

sddec20-25
PrISUm
Dr. Nathan Neihart
Joseph DeFrancisco - Team Leader
Kyle Czubak - Software Development
Ben Kenkel - Software Development
Ryan Willman - Hardware Design
Bryan Kalkhoff - Hardware Design
Connor Luedtke - Hardware Design

Team Email and Website:

sddec20-25@iastate.edu
sddec20-25.sd.ece.iastate.edu

April 25, 2020

Executive Summary

Development Standards & Practices Used

Development Standards & Practices Used

- Agile Development
- IEEE 1679.1-2017: IEEE Guide for the Characterization and Evaluation of Lithium-Based Batteries in Stationary Applications

Summary of Requirements

- Characterize 8 batteries at once
- Continuous monitoring for safe operation
- Perform full-cycle characterization of Lithium-Ion Batteries
    - Measure current in and out of individual batteries
    - Voltage monitoring for each battery
    - Calculate battery capacity and internal resistance from above measurements
    - Temperature measurement
- Storage of data for future analysis
- Serialize batteries and the associated data

Applicable Courses from Iowa State University Curriculum

- EE 230: Electronic Circuits and Systems
- EE 333: Electronic Systems Design
- CPR E 288: Embedded Systems I: Introduction
- CPR E 488: Embedded Systems Designs
- COMS 363: Introduction to Database Management Systems

New Skills/Knowledge acquired that was not taught in courses

- PCB Design
- Lithium Battery Characteristics
- Git & GitLab
    - Intense use for SE and CPR E
    - Basic Git operations for EE
- Advanced Embedded System Development

Table of Contents

List of figures

1 Introduction

1.1 ACKNOWLEDGMENT

Dr. Nathan Neihart, an Associate Professor in Electrical Engineering at Iowa State, will be the faculty advisor for this project and will be providing the technical support needed for the project. Additionally, PrISUm Solar Car will be providing lithium-ion batteries for us to use while testing.

1.2 PROBLEM AND PROJECT STATEMENT

An important step in lithium-ion battery pack manufacturing is to have an accurate characterization of the charging and discharging curves for every battery cell before it goes into a parallel module. This allows for the grouping of similar performing batteries into each module, which ensures that the batteries in a module will charge and discharge at similar rates, improving the efficiency of the pack. PrISUm Solar Car currently does not have a reliable method for performing this characterization and is proposing that we develop a system that can address this need.

Our proposed solution to this problem would be making a device that would charge and discharge the batteries and graph the current and voltage characteristics. To help group battery cells, the device would also be able to remember which cells were characterized and store that information somewhere accessible.

1.3 OPERATIONAL ENVIRONMENT

The operational environment for the final product will be used in a typical indoor environment of ambient temperatures between 10 and 30 °C. Due to the mechanical enclosure, and high-power dissipation at times during the characterization process the ambient temperature may rise significantly. The temperature will always be monitored and efforts to provide adequate cooling will be investigated.

1.4 REQUIREMENTS

Main Controller:

The main controller will be responsible for controlling the testing process. This includes starting and stopping tests, assigning battery numbers, and collecting and storing the test data. We are using a Raspberry PI as our main controller.

Module Unit:

This is where the testing will take place. We will be capable of characterizing 8 batteries at the same time on each module. Each battery will have temperature, voltage, and current monitored and reported back to the main controller throughout the test. Each battery will have a programmable current load circuit and charging circuit directed by the module's microcontroller.

Minimum Viable Requirements:

- Perform full-cycle characterization
  - Measure current in and out
  - Voltage measurement
  - Temperature measurement
- Serial number tracking every battery with associated data
- Storage of data for future analysis
- Continuous monitoring for safe operation
- Characterize 8 batteries at once

Stretch Goals:

- Battery Module Optimization Software that automatically groups batteries into modules based on gathered data
- Build a full-scale characterizer capable of ~40 batteries at once.
  - Using multiple modules.
- Web-based interface for viewing battery characteristic data.

1.5 INTENDED USERS AND USES

The project is intended to provide the ability to characterize batteries to optimize Lithium-Ion battery packs' efficiency and longevity.

There are two intended users' groups for this project:

1. PrISUm: This project was proposed by PrISUm to fulfill a need in their solar car battery pack design.
2. Non-Professionals: Most people that are not in industry skip the characterization part of designing a battery pack due to no viable market solution. As larger-scale battery pack design is becoming more feasible due to declining lithium battery costs, it is becoming increasingly feasible to make large battery packs, which results in an increasing need for proper battery characterizing methods.

1.6 ASSUMPTIONS AND LIMITATIONS

Assumptions

- End users will understand basic lithium battery safety standards.
- End users will have access to computers and basic computer skills.
- Our proposed solution will be used in a climate-controlled room.

Our Limitations

- The cost of the final product will not exceed $500
- The system will require an AC power source.
- The end product cannot be used if the ambient temperature is outside the battery's usable temperature range as specified by the lithium-ion battery datasheet.

1.7 EXPECTED END PRODUCT AND DELIVERABLES

The end deliverable will be a complete battery characterization system capable of running current, voltage, and temperature tests on up to 8 batteries at a time. The finished system will process data and upload it to a database. Once the lithium cells are inserted into the end product and the device is started, no more user input will be needed until the end of the testing cycle. To achieve this goal, multiple subsystems must be delivered, including the main node, support for multiple testing nodes, and a database for storing the

2. Specifications and Analysis

2.1 PROPOSED APPROACH

This project is broken into several main segments, the database for storing test data, the high-level main controller, and several test nodes. Each test node will conform to IEEE Standard 1679.1-2017 for the characterization and evaluation of each battery cell. A test node will need to be capable of running the test and monitoring the state of each connected battery. The test nodes will then report the data to the main controller, which will record the data to our database for storage and analysis.

2.2 DESIGN ANALYSIS

This semester we completed the battery test program and a significant portion of the schematic design work. The test program outlines the process for completing the capacity and internal resistance measurements and sets requirements for the hardware so that these calculations can be made accurately. For hardware design work we have completed the voltage and current measurement, charging, constant current load, temperature measurement, and the microcontroller hardware. It will be challenging to verify the functionality of most of these circuits without hardware in hand. The main exception is the load circuit which has been verified in simulation.

Moving forward it would be nice to prototype pieces of the design on perf board or breadboard. This is dependent on having lab access over the summer or next semester. This would allow us to verify design work before moving to the layout stage.

For the software app, we have completed the app prototype and database schema. One issue we faced was communicating with the characterizers through CAN with a web app. This will be challenging to do with having 2 systems working together. All URL paths are created and moving forward the HTML and CSS will be created so testing can start.

2.3 DEVELOPMENT PROCESS

With this project having so many different parts - hardware design, embedded system design, backend development, etc. - choosing a development process will be difficult. While agile will work well creating a website and creating a database, the waterfall process will work better with hardware design and embedded systems software. Since the hardware design and embedded systems software is the more important part of this project, we will be using the waterfall process for the vast majority of this project.

2.4 CONCEPTUAL SKETCH

The overall design of the project is shown in Figure 1. Figure 2 shows the subsystem view of the main controller. Figure 2 illustrates the layout of the battery measurements and communication. The team will follow these block diagrams when designing the product.

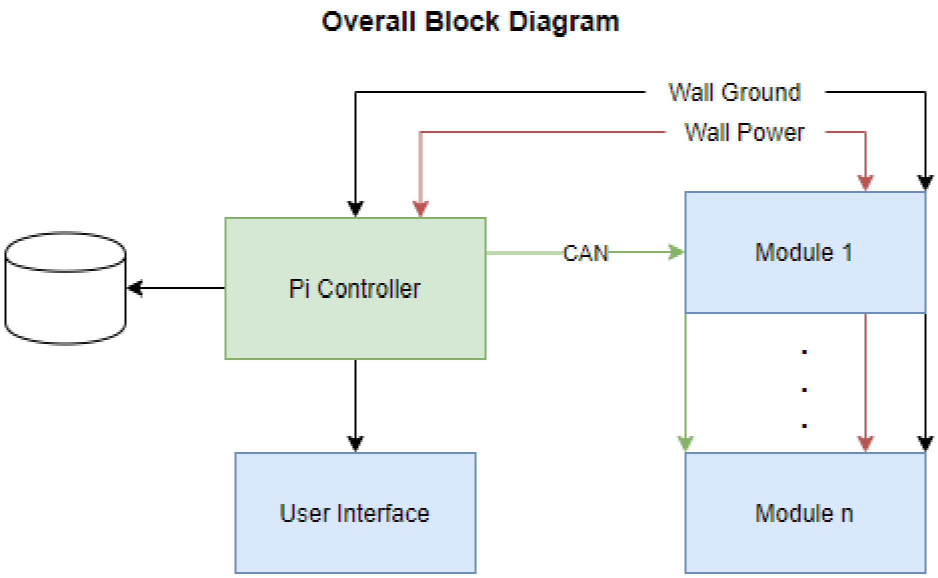


Figure 1: High-Level Block Diagram.

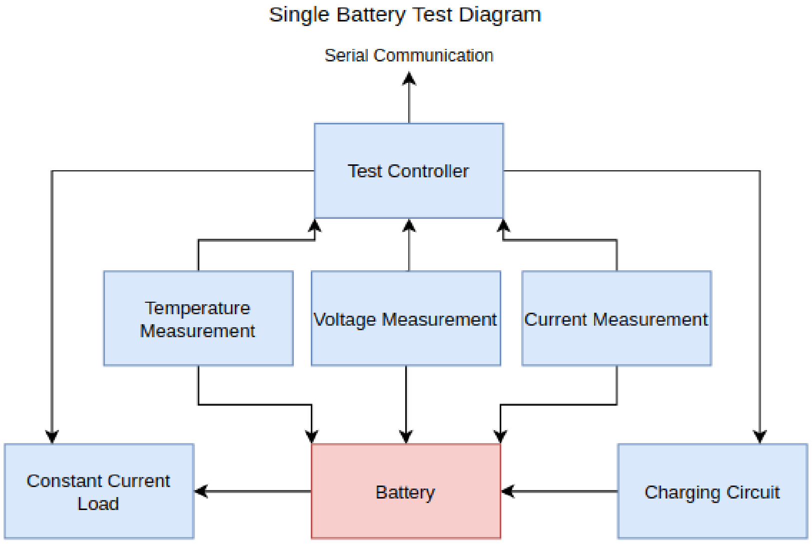


Figure 2: Battery Test Diagram

2.4.1 Enclosure

No official system enclosure has been designed yet. Once the PCB board has been designed, a 3d printed enclosure will be designed to accommodate it. The enclosure will allow for all components to be contained inside and only exposing the screen and 18650 holders. The enclosure should allow for easy access to the internal components for servicing.

3. Statement of Work

3.1 Previous Work And Literature

There are already existing small scale battery testing equipment that can be purchased commercially, such as the Opus BT-C3100, but they are normally very limited in functionality [1]. Our team is attempting to design and implement a system that can test a larger quantity of batteries at one time.

Many members of our team have extensive experience designing circuits and creating PCBs. Also, our software developers have experience creating embedded software systems from scratch. Regarding Lithium-Ion batteries and their safety, we have a team member who is familiar with the proper handling and storage of Lithium-Ion batteries.

3.2 Technology Considerations

Hardware

- The hardware must interface with the digital communication protocols we are using
- The hardware must respond to unsafe operating conditions for the batteries
- Analog measurements reported over digital bus protocol decrease hardware complexity but increase software complexity.
- PCB will be designed using a 4-layer stack up to take advantage of large thermal mass to dissipate large amounts of heat.

Software Section

- The software must be deployable onto the chosen microcontroller.
- The software must be able to run continuously for the entire duration of the test.

3.3 Task Decomposition

The project is broken into several subsystems. On the hardware side, there will be the power circuit, charging system, the discharge circuit, the microcontroller, and measurement circuitry. For software, there is a Flask app on the raspberry pi for interfacing with the user, a database for storing information, and the testing management software. The power circuit will provide a 5-volt rail for the digital circuitry and a 12-volt rail for all other systems. The charge/discharge systems will provide constant current charge and discharge of each cell. Measurement circuitry will relay voltage, current, and temperature data back to the main controller for each cell.

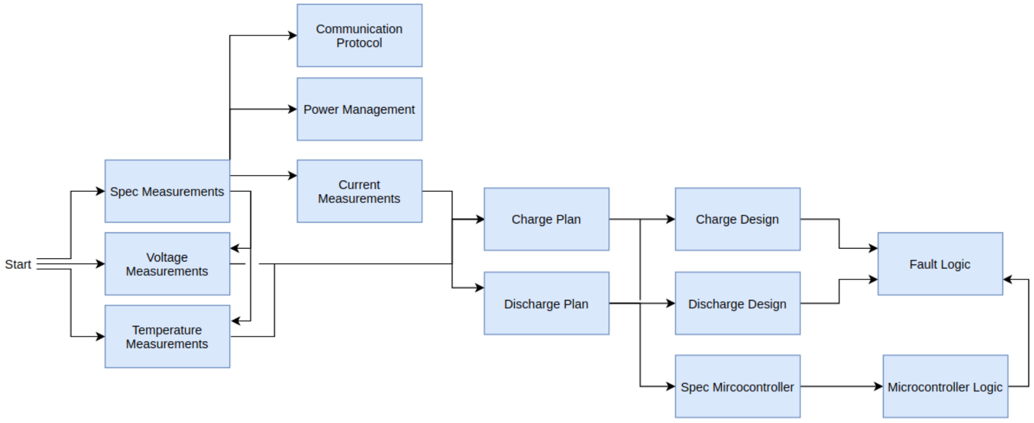


Figure 3: Flowchart of the project

3.4 Possible Risks And Risk Management

Lithium-Ion batteries are volatile and require continuous monitoring for safe operating conditions. If Lithium-Ion batteries are not handled within the specifications, batteries can react with unplanned thermal events.

To manage risk, both hardware and software solutions will be implemented. The chosen charging circuit follows JEITA guidelines [2] where thermal management components do not let the lithium cell exceed 60 Celsius. The circuit also has short circuit protection if the batteries are improperly inserted and contain fuses if too much current is drawn. In addition to our hardware protections, we will have our software monitoring all variables as well so that the testing can be halted if parameters such as temperature or current go outside of expected ranges. However, if an issue does arise, we will be following Iowa State University's Environmental Health & Safety's guidelines for handling problematic Lithium-Ion batteries, which includes isolating cells in the sand, having a fire extinguisher nearby, and we are aware of how to properly dispose of damaged batteries.

3.5 PROJECT PROPOSED MILESTONES AND EVALUATION CRITERIA

Overall, key milestones would be the completion of the different subsystem designs, communication, web app, database, charging, load circuit, sensing, and fault logic.

For the first semester, we have proposed the completion of all necessary schematics with simulations. This will be composed of finding components, designing the circuits, and testing using PSpice. Following this timeline is ideal to provide adequate time in the second semester to design the PCB and troubleshoot if needed. The backend and database for the web app are complete. Next semester will be working on the HTML and CSS for the web app and then do testing for it.

3.6 PROJECT TRACKING PROCEDURES

This project will be tracked mostly through the provided Git repository. All project files will be maintained through git, with the master branch being reserved for functional prototypes and the completed project. The team will be using GitLab Issues for tracking tasks that need completed, as well as documenting their progress.

In addition to git, the team has weekly meetings with Dr. Neihart to discuss the team's progress and to highlight what tasks should be worked on over the next week. Additional meetings will be held within the team for working on different tasks, planning future work, and for working on system integration and reporting.

3.7 EXPECTED RESULTS AND VALIDATION

The high-level end product of this project will be a battery characterizing system that is capable of automatically characterizing at least 8 battery cells at a time. The system will output data to a database that can be used to create graphs showing the charging and discharging rates of each tested battery.

4. Project Timeline, Estimated Resources, and Challenges

4.1 PROJECT TIMELINE

The project timeline for the first semester is implemented by using a Gantt chart in Figure 4. Day 0 starts on 1/26/2020 while day 77 is on 4/26/2020. We will start by setting and understanding our system requirements for roughly 2 weeks. Then transition into researching for needed materials in understanding how to design our project. Once the research is complete, components will be chosen based on the parameters found in our research. Finally, the circuit will be implemented through Altium and then simulated through Spice for validation.

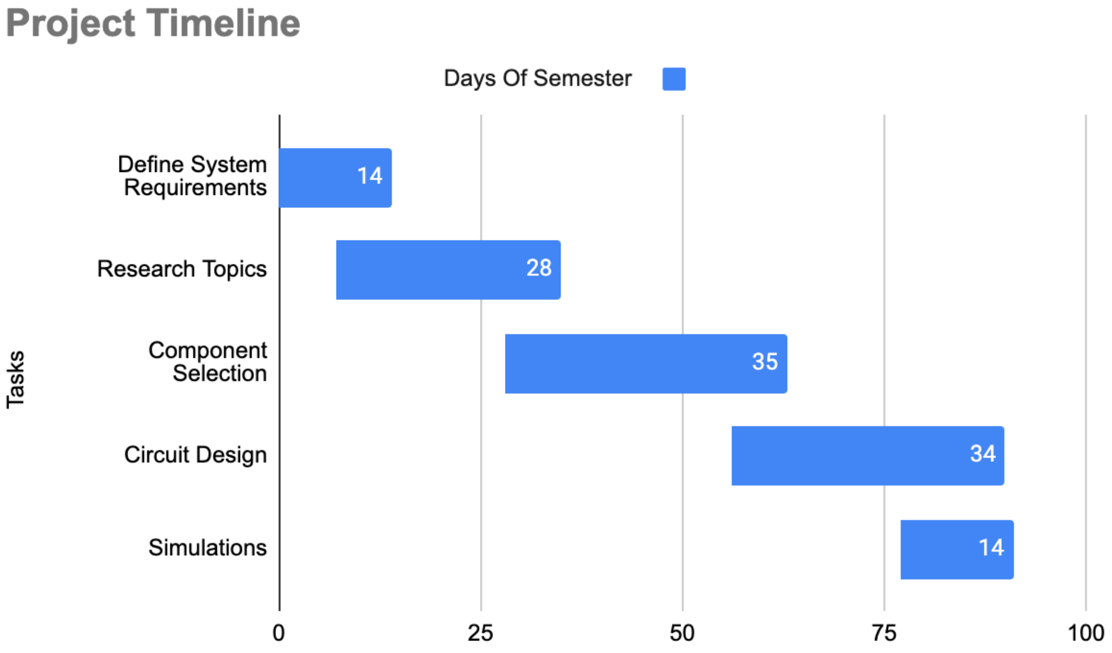


Figure 4: Project Gantt Chart

Below are specific deliverables that should be completed by specific dates.

- Hardware Team
    - 2/9 - Define System Requirements
    - 3/1 - Research
        - Spec current, voltage, and temperature measurements
    - 3/29 - Component Selection
        - 3/14 - Main Components
        - 3/29 - Sub-Components
    - 4/25 - Circuit Design
        - 4/10 - Main Schematic
        - 4/12 - Micro Circuit
        - 4/15 - Load, Charge Circuit
        - 4/25 - Other Circuits
    - 4/26 - Simulations
        - 4/25 - Load Simulation
- Software Team
    - 4/26 - Web App and Database

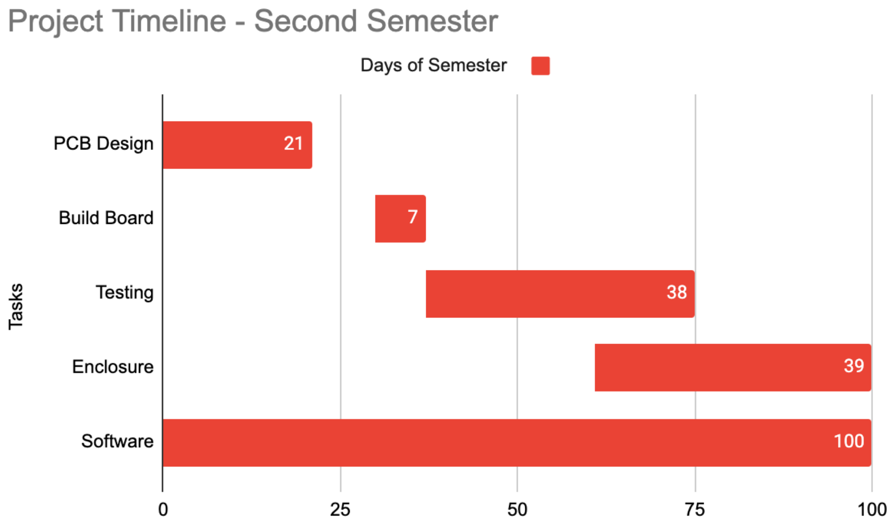


Figure 5: Estimated Second Semester Timeline

Figure 5 shows a tentative timeline for semester 2, this timeline may be altered when the semester gets closer. In this timeline, we will complete the PCB Design within the first month then have to wait for components and PCB to arrive before soldering the board. Once fabricated, a long testing phase will occur to validate the accuracy and safety of the battery characterizer. An enclosure will start to be designed during testing to accommodate any modifications. The software will be implemented during the entire semester to make sure proper communication is working in the user interface. Both of these timelines illustrate an easily achievable path to have a working unit at the end of the second semester. Even if the timeline slips by a week or two there will be no impact on the final project readiness.

4.2 FEASIBILITY ASSESSMENT

Despite the added challenge of having to work from home, we have concluded that the project won’t need to change. We came to this conclusion because:

- We were beginning the schematic layout which can be done remotely
- Code can be done remotely, and we have access to hardware to run it on
- Conferences/meetings can be done via Hangout

4.3 PERSONNEL EFFORT REQUIREMENTS

Table 1: Personal Effort Requirements

Task	Est. Time	Description
Define System Requirements	2 Weeks	Functional requirements needed to begin research.
Research Topics	4 Weeks	Individuals will be assigned topics to explore.
Battery Component Selection	2 Weeks	Find components on DigiKey that meet project needs.
System Integration	2 Weeks	Integrate software and hardware.
System Testing	2 Weeks	Collect data to see if the device is functioning properly.

4.4 OTHER RESOURCE REQUIREMENTS

Basic lab equipment found in Coover.

4.5 FINANCIAL REQUIREMENTS

The project has a budget of $500 with the possibility of PrISUm being able to obtain items if needed. The primary goal is to stay within the budget to create the final project.

5. Testing and Implementation

5.1 INTERFACE SPECIFICATIONS

The team is currently evaluating methods for carrying out the testing of our system. We plan on testing large subsystems individually first, then testing again once integrated.

5.2 HARDWARE AND SOFTWARE

Testing each sub-system and the final product will require the use of a few hardware and software tools.

Multimeter: The multimeter will allow us to measure both the voltage and current of the batteries to confirm the functionality of our measurement systems. Additionally, the multimeter will be greatly beneficial for debugging issues in the circuit designs.
DC Power Supply: Will be required to provide known and stable voltages to various parts of the circuit.
Atmel Studios: The built-in debugger for Atmel Studios will be beneficial for debugging any software related issues.
Thermal Imaging Camera: A thermal imaging camera will be used to monitor the temperature of the batteries throughout the testing process while we verify our temperature monitoring circuit.
Chrome: A web browser to do testing with our web app.
Postman: A program to test POST and GET messages to our website backend.
Sand: The team will have buckets of sand available to isolate batteries that are beginning to show symptoms of improper treatment to minimize the likelihood of a large Lithium-Ion fire.

5.3 FUNCTIONAL TESTING

At the end of 491, we did not have any physical hardware yet, so we have only completed simulations of our designs where possible. Once we get hardware in and access to lab equipment again we will be purchasing small subsets of the project components to prototype some designs.

5.3.1 Hardware: Board Power and PMIC Functionality

There are multiple stages to powering this project. First, there is a wall AC adapter that will provide a 12V rail. Testing will involve checking for the presence of the 12V at various test points on the PCB. Next, to make sure that the AC adapter is operating in spec, we will observe a 12V test point on the oscilloscope. We will be checking for transient startup performance and steady-state ripple.

We will be using a TPS82130 switching voltage regulator IC that will buck down from 12V to the 5V rail for the digital circuitry. We will also be using a 12V linear regulator. This is just to clean any voltage ripple coming from the power brick.

5.3.2 Hardware: Battery Charging

For the charging circuit, we are using a TI bq2425 which will provide constant current charging while the battery is below 4.2V. The charge rate is programmable and will be determined by the microcontroller via I2C communication. Testing this circuit will require careful attention as a battery will be involved. We will create a test bench that has just the battery charger, its required components, and a single lithium-ion cell. The cell will be monitored for the temperature to ensure that it does not exceed the temperatures listed by the JEITA standards. It will also be fused to a fairly low current level. These precautions should prevent any major disasters. Of course, a fire extinguisher rated for lithium fires will be close at hand. The microcontroller will provide the charge enable signals over I2C, these messages will be monitored over the oscilloscope. Current into the battery will be measured by a multimeter and the voltage profile will be monitored on the oscilloscope. We will expect a constant voltage charge at the start to precharge the battery, once this initial stage has completed, the charger will move into a constant current charge. This will take place until the battery has reached approximately 90% of its full charge. Once that threshold has been crossed the charger will again move into a constant voltage charge to finish the charging cycle.

5.3.3 Hardware: Constant Current Electronic Load

The load circuit has been tested for transient performance as well as phase margin. This was done through the SPICE simulation. To test the phase margin, a test bench to measure the open-loop gain of the system needs to be created. This test bench, created in TINA-TI, is shown in Figure 6. The magnitude and phase of the top op-amp output are plotted in Figure 7. To find the phase margin, the open-loop gain of the system needs to be measured. This is done by breaking the feedback loop and injecting an AC signal of increasing frequency. The direct output of the system is the open-loop gain. To prevent the system from performing differently, the broken feedback loop needs to be terminated with the same impedance that it saw before. This is done by attaching a copy of the circuit coupled through large capacitors and inductors. This allows for AC interaction between the circuits but also maintains the correct DC bias conditions. Phase margin is measured at the point of 0dB gain and occurs at 380kHz with a 2.2uF compensation capacitor. This gives a fair amount of stability headroom while also allowing the system to respond quickly.

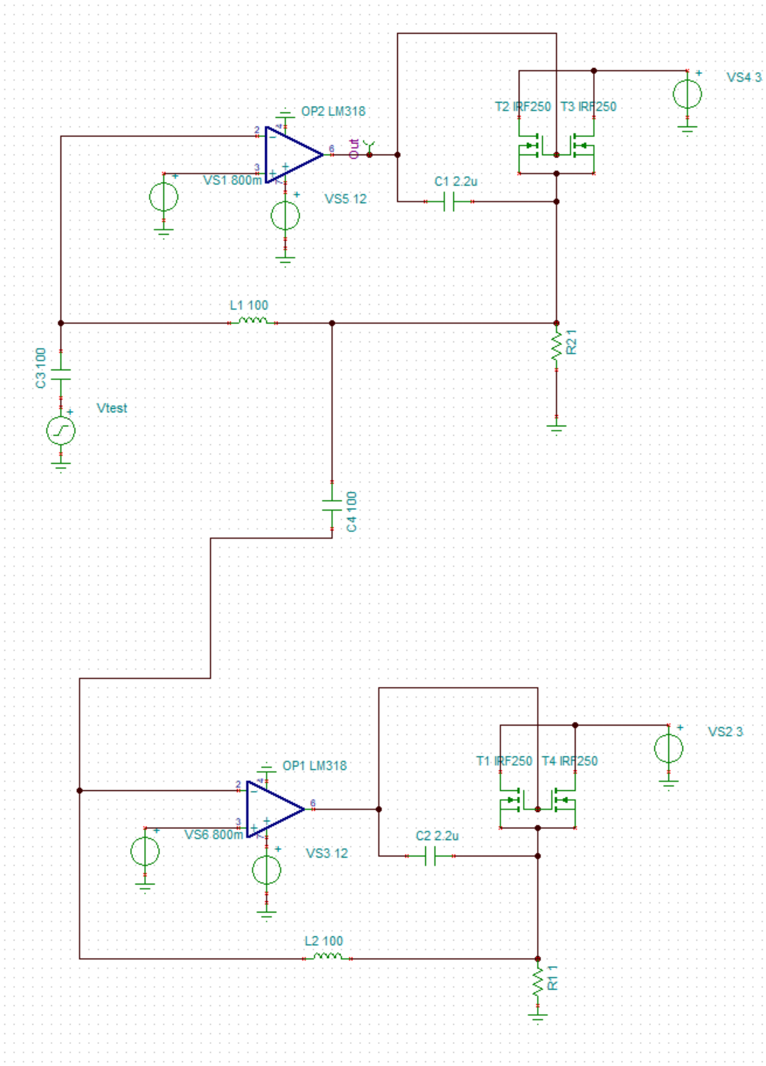


Figure 6: Simulation for Constant Load

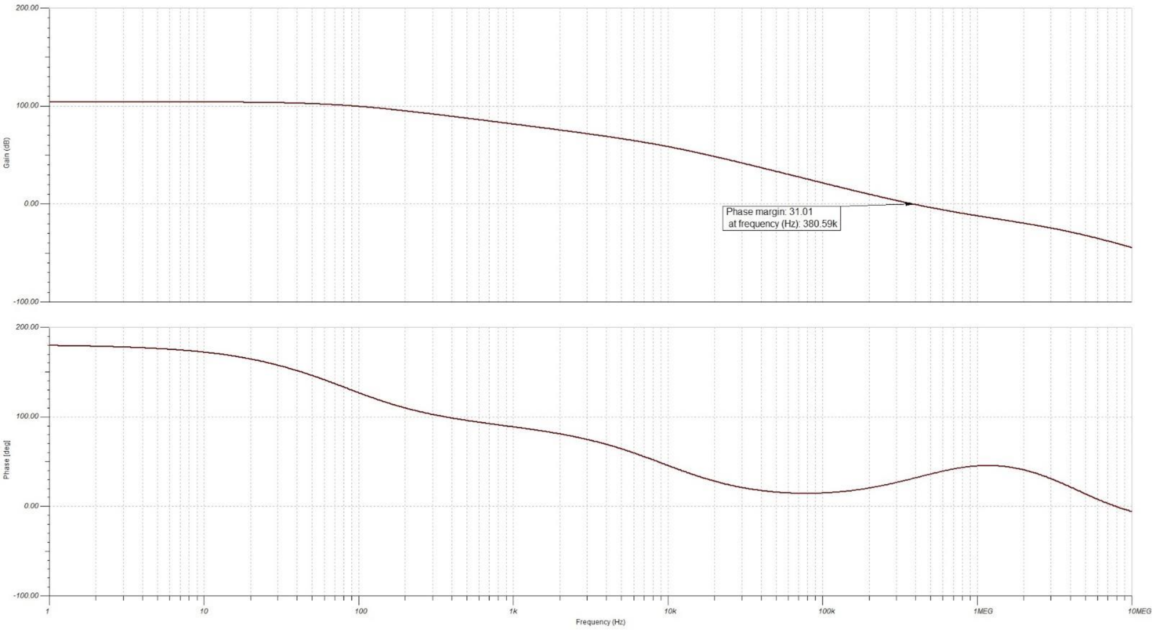


Figure 7: Simulation Result for Constant Load

A transient simulation was also performed to get a better sense of the overshoot and response time of the circuit. The test bench to perform this simulation is shown in Figure 8. The voltage reference is supplied by a 1kHz square wave with a 500mV amplitude and a 1V DC offset. The load current is measured by the ammeter in series with the resistor R1. The battery voltage is supplied by VS2 and is set to 3V, although due to the design of the circuit this value is not very significant. The compensation capacitor is set to 2.2uF. The results of the simulation are shown in Figure 9. As expected, the input pins of the op-amp pull to be the same value, this has the effect of setting the load current through the resistor R1. The overshoot of this system for a 1.5A load is approximately 50mA. The settling time is less than 100us.

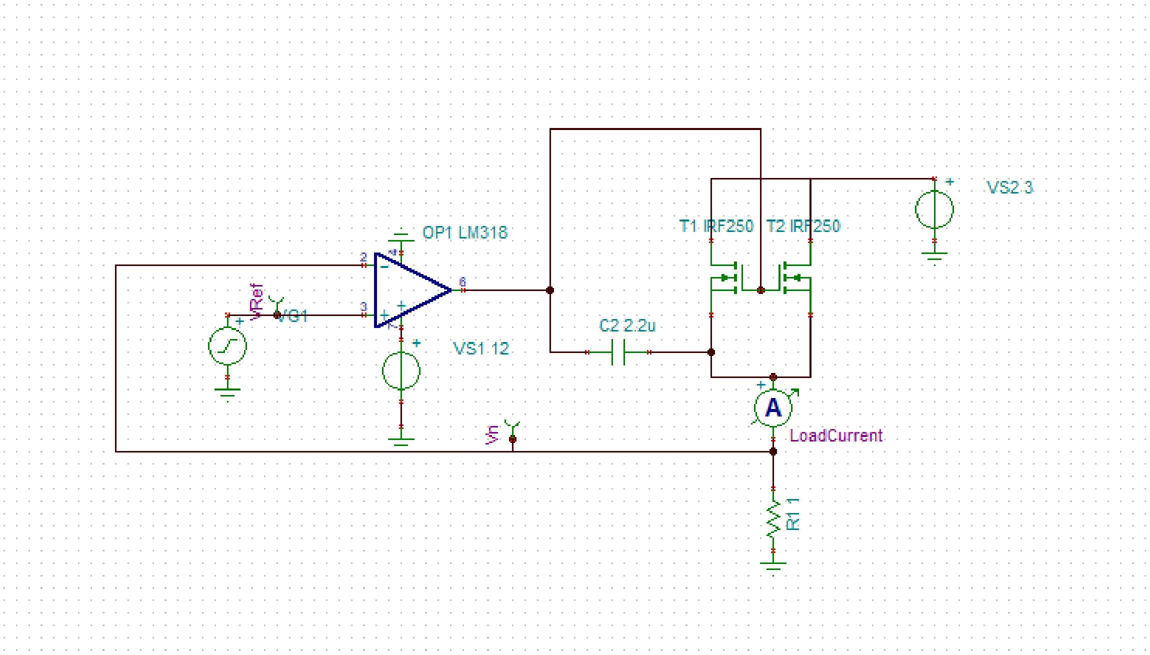


Figure 8: Transient Simulation

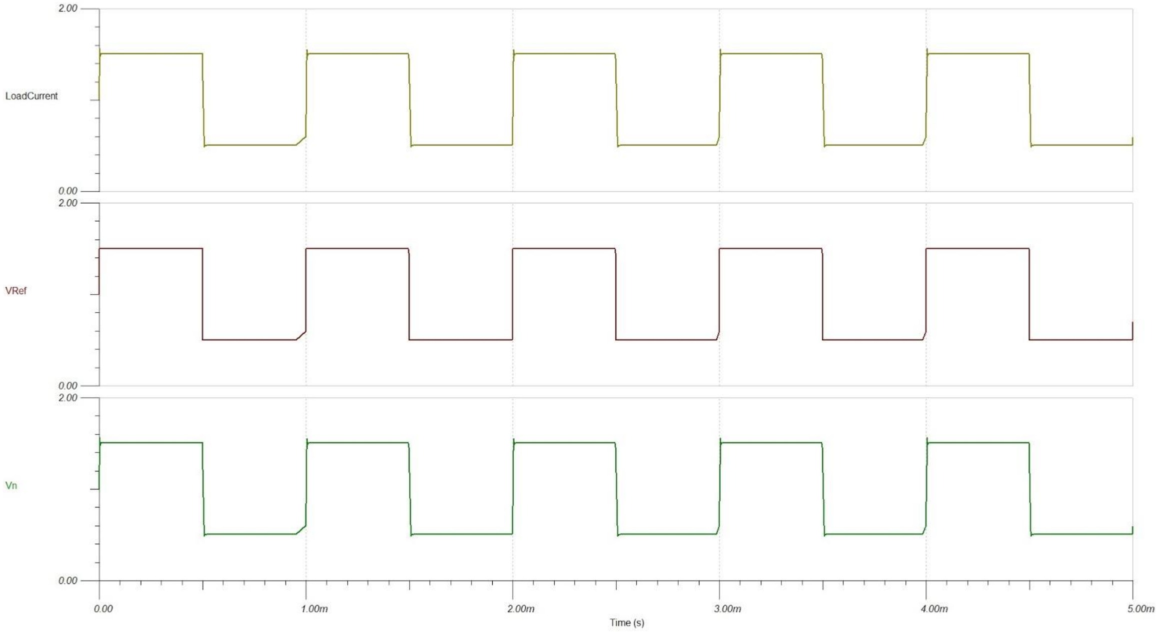


Figure 9: Result of Transient simulation

The simulation results should be fairly accurate, but manufacturing variance could throw a wrench into the circuit performance. Also testing at higher ambient temperatures will need to be completed to get a more accurate picture of circuit performance. Testing with physical hardware will also occur once we have access to lab equipment. We will be able to check steady-state performance as well as switching performance. Testbenches similar to the simulations will be created to verify the validity of the simulations.

5.3.4 Hardware: Voltage and Current Sensing

For the voltage and current sensing, we are using the Texas Instruments INA3221 along with a 100mΩ sense resistor. The output is communicated through I2C. To eliminate initial variables, we will provide device power and test voltage with a bench DC supply. The INA3221 has a power good pin with an LED connected to it. When we provide device power, the power good LED should turn on. To test the accuracy of the measurements, we will need to provide a dummy load to the circuit in the form of a potentiometer and apply a voltage with the bench supply. Using a multimeter, we will get baseline results for current that we can compare to the INA3221 measurements. To read the output of the device, we will use the serial decoder on a lab oscilloscope to read the I2C messages.

The INA3221 will be continually monitoring the current into and out of the battery. We will be using the digital output to communicate if there are any fault conditions. To test this, we will set up the device with a low current limit to make the test safer to conduct. Then we will force a current below that limit and steadily increase it. Once we cross the threshold, we should receive a message on the I2C bus that the current limit has been exceeded.

5.3.5 Hardware: Microcontroller

The microcontroller we will use is an Atmel ATSAME51J. The main features that will need to be verified are:

1. The microcontroller can be programed
2. The microcontroller can communicate to the pi via CAN
3. The microcontroller can communicate to the other circuits via I2C

To verify that the microcontroller can be programmed, we will write a simple program that will simply turn on an LED. This will provide a visual indicator that the microcontroller was programmed.

To verify that the microcontroller can communicate via CAN properly, we will connect the system to a known working CAN device (provided by the client), that will send and receive messages over the CAN network. When the CAN network on the microcontroller is confirmed working, the next step would be to verify CAN working on the pi. Similarly, to verifying the microcontroller, we will write a simple program to make the pi send and receive CAN messages.

Finally, to verify that the I2C communication bus is working, we will have the microcontroller send messages to each of the IC connected. Probing the bus line with an oscilloscope to determine if we see the proper messages being sent over the bus.

5.3.6 Software: Embedded Software

This testing will follow the same pattern as the hardware testing plan. Since we are using CAN for communication, we will have debug IDs to send out messages when something happens.

5.3.7 Software: Web App

We are going to use a web app for the user interface. For internal functions, we will have unit tests and for the user interface (buttons, text boxes, etc) we will use a browser and test manually. For communication requests, we will use Postman to check the return values.

5.4 NON-FUNCTIONAL TESTING

Once we can start sending battery data to a database, we will need to observe the data transfer speed and determine if we will need to reduce the amount of data that we send, or if more data is required to produce clear charging and discharging characteristics.

5.5 PROCESS

The team is still developing our initial designs and have started creating a detailed testing plan.

5.6 RESULTS

This section will be updated once a testing plan is created and implemented.

6. Closing Material

6.1 CONCLUSION

At the time of this report, our team has been primarily focused on developing initial hardware designs and the planning of our communication protocols and database. We feel that we are in a pretty good spot for this semester and shouldn't have too many issues moving forward with this semester.

6.2 REFERENCES

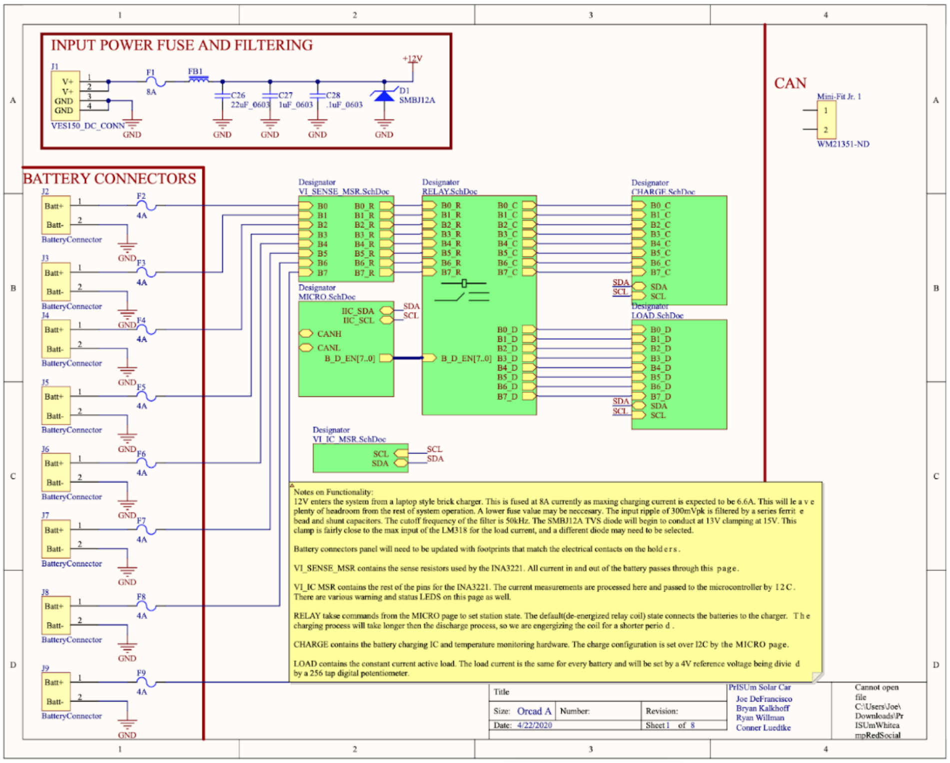
[1] Opus BT-C3100 Digital Battery Charger. "Opus BT-C3100 V2.2 4 Bay Digital Battery Charger." 18650 Battery Store. https://www.18650batterystore.com/Opus-p/opus-btc3100-v2.2.htm (retrieved April 25, 2020).

[2] Qian, J. (n.d.). Li-ion battery-charger solutions for JEITA compliance. Retrieved April 25, 2020, from http://www.ti.com/lit/an/slyt365/slyt365.pdf?ts=1587833638791

6.3 ACRONYMS

AC: Alternating Current
CAN: Controller Area Network
DC: Direct Current
IC: Integrated Circuit
I2C: Inter Integration Circuit, communication protocol
IEEE: Institute of Electrical and Electronics Engineers
JEITA: Japan Electronics and Information Technology Industries Association
LED: Light Emitting Diode
PCB: Printed Circuit Board

6.4 APPENDICES



Notes on Functionality:
12V enters the system from a laptop style brick charger. This is fused at 8A currently as maxing charging current is expected to be 6.6A. This will leave plenty of headroom from the rest of system operation. A lower fuse value may be neccesary. The input ripple of 300mVpk is filtered by a series ferrite bead and shunt capacitors. The cutoff frequency of the filter is 50kHz. The SMBJ12A TVS diode will begin to conduct at 13V clamping at 15V. This clamp is fairly close to the max input of the LM318 for the load current, and a different diode may need to be selected.

Battery connectors panel will need to be updated with footprints that match the electrical contacts on the holders.

VI_SENSE_MSR contains the sense resistors used by the INA3221. All current in and out of the battery passes through this page.

VI_IC MSR contains the rest of the pins for the INA3221. The current measurements are processed here and passed to the microcontroller by I2C. There are various warning and status LEDS on this page as well.

RELAY takse commands from the MICRO page to set station state. The default(de-energized relay coil) state connects the batteries to the charger. The charging process will take longer then the discharge process, so we are engergizing the coil for a shorter period.

CHARGE contains the battery charging IC and temperature monitoring hardware. The charge configuration is set over I2C by the MICRO page.

LOAD contains the constant current active load. The load current is the same for every battery and will be set by a 4V reference voltage being divied by a 256 tap digital potentiometer.

Figure 10: Circuit Schematics as of 4/25/2020-Main Page

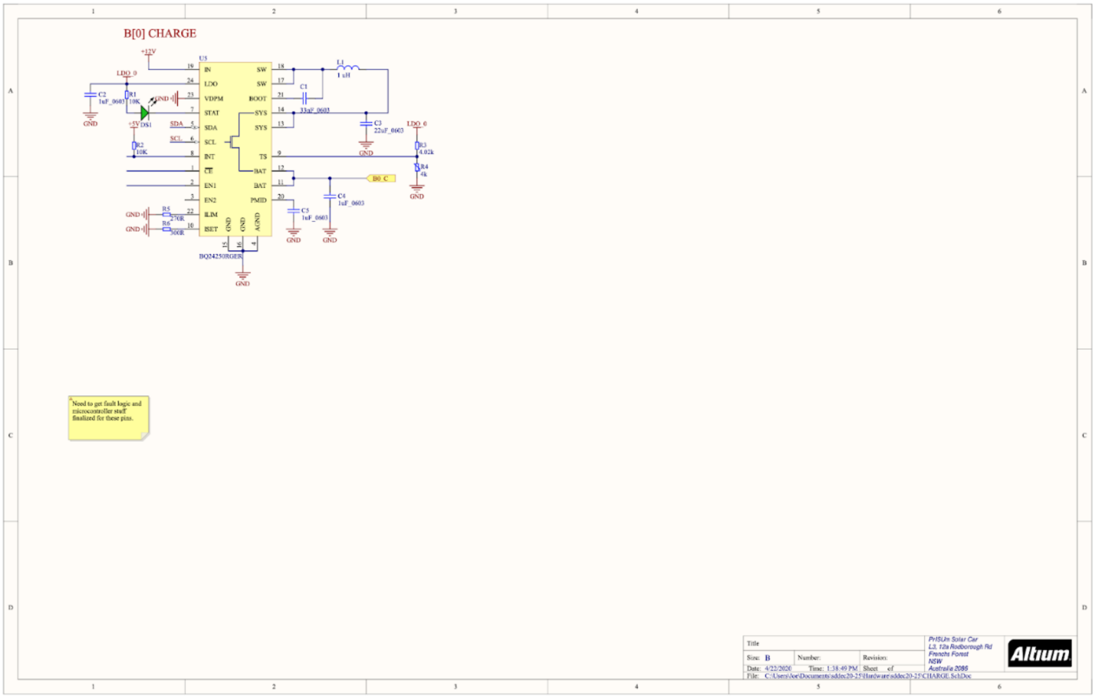


Figure 11: Circuit Schematics as of 4/25/2020- Charge

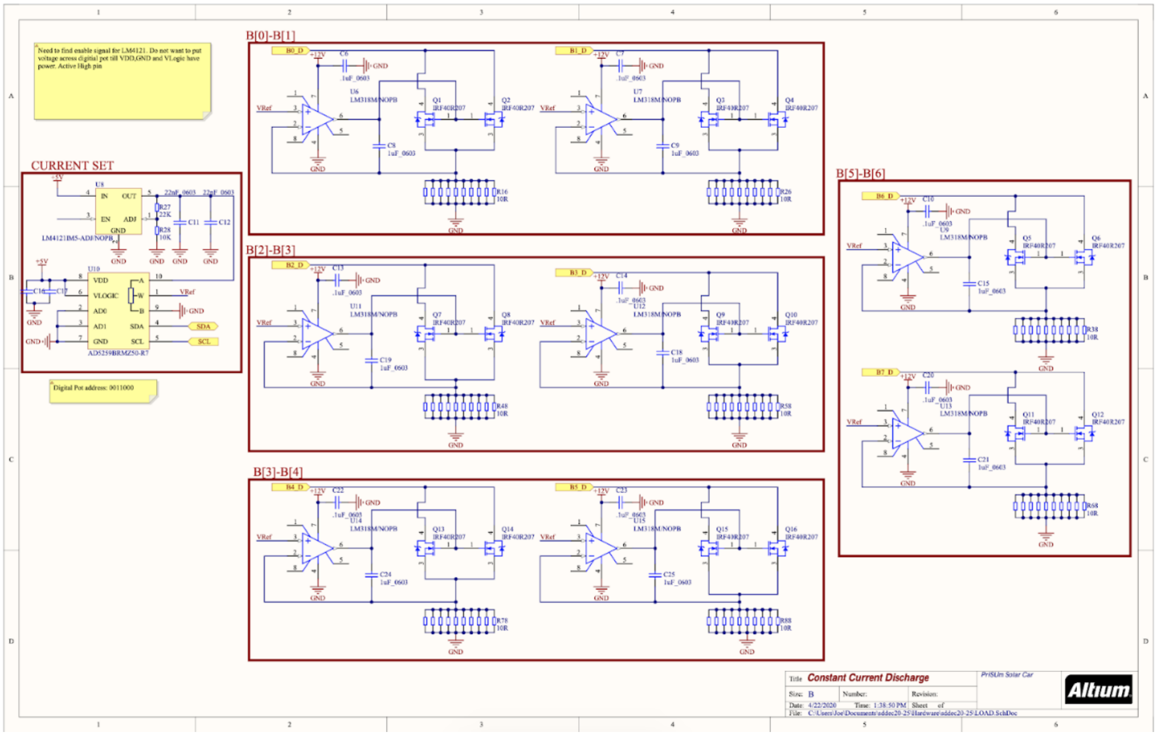


Figure 12: Circuit Schematics as of 4/25/2020- Discharge

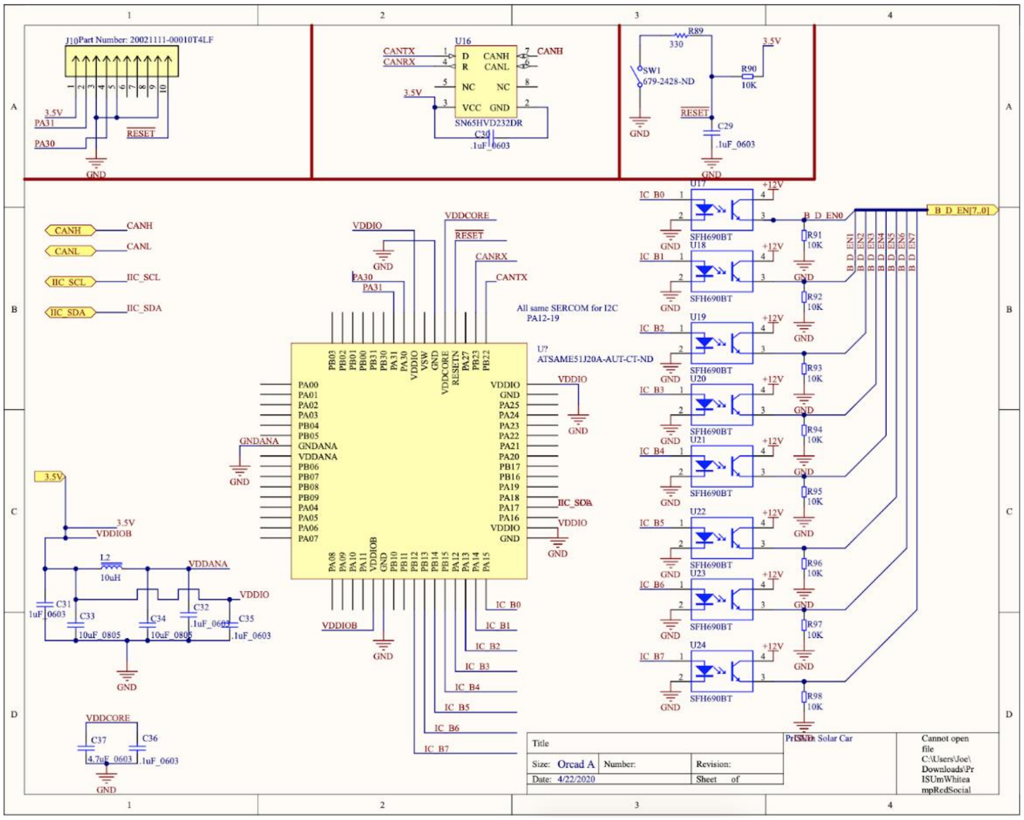


Figure 13: Circuit Schematics as of 4/25/2020- Microcontroller

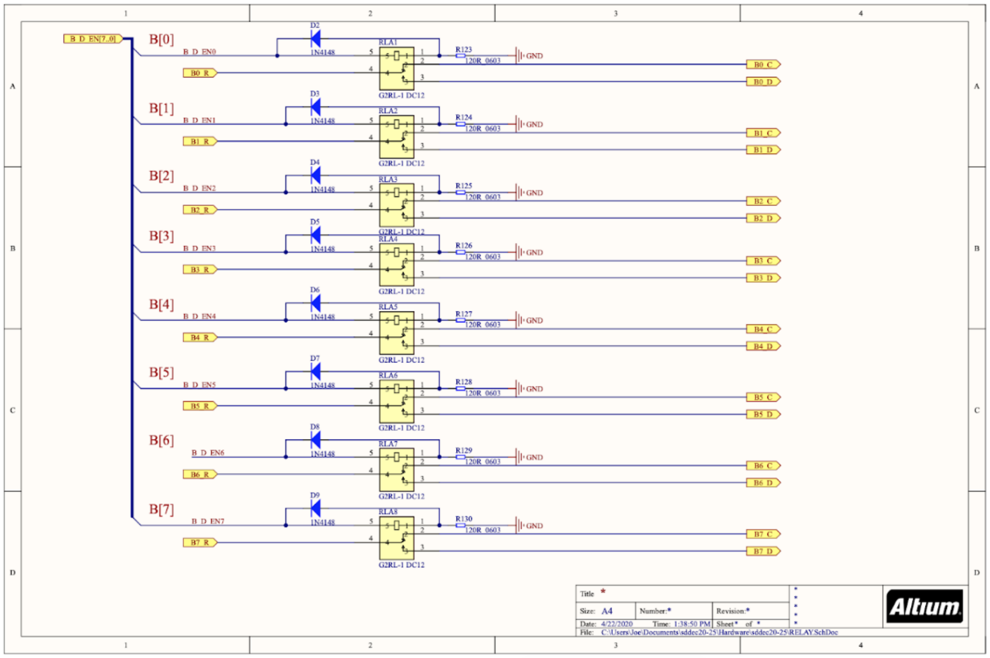


Figure 14: Circuit Schematics as of 4/25/2020- Relays

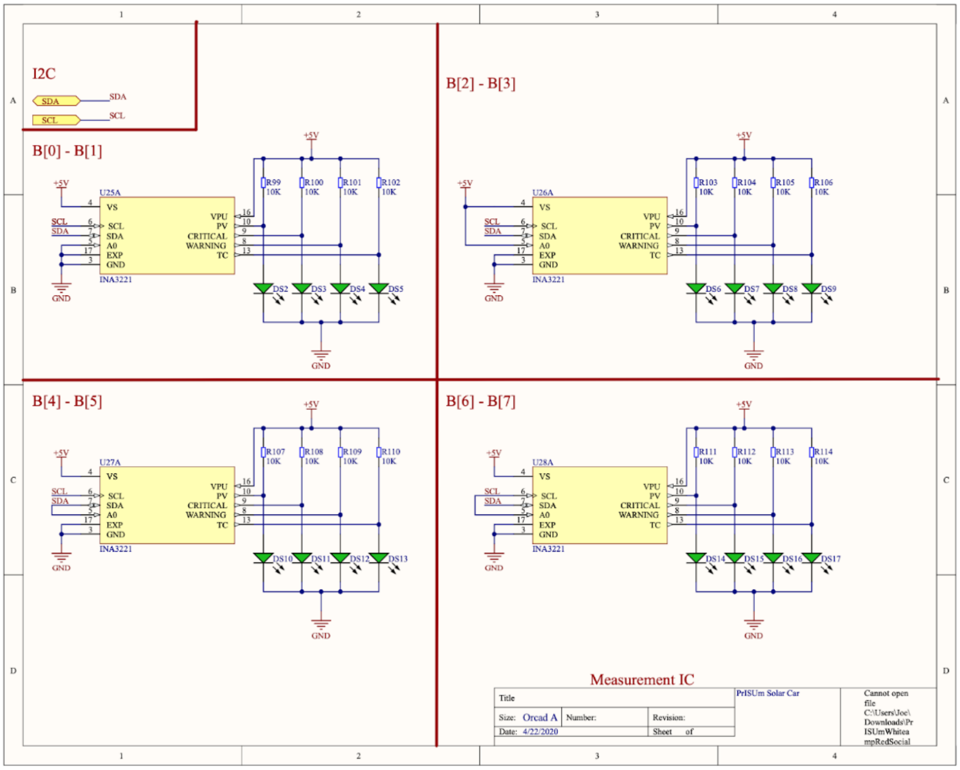


Figure 15: Circuit Schematics as of 4/25/2020- Measurement IC

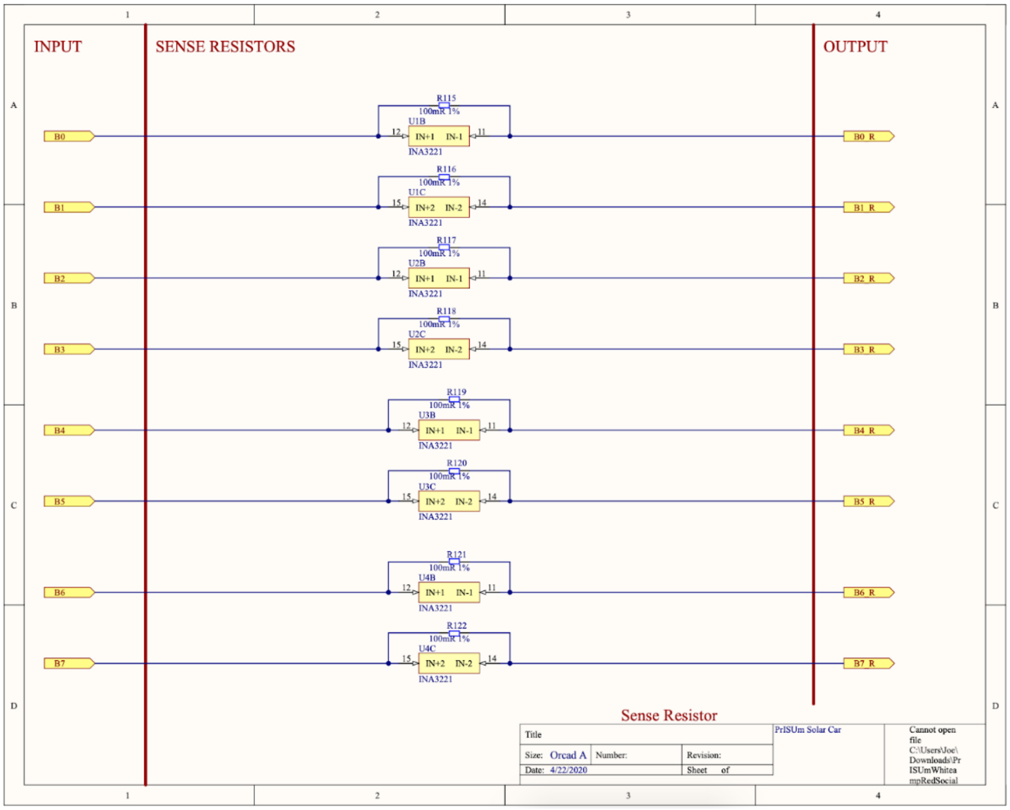


Figure 16: Circuit Schematics as of 4/25/2020-Sense Resistor