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

March 28, 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 8-10 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
    - 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 state of charge on every battery cell before it goes into a parallel module. This allows for the grouping of similar performing batteries into each module. This process ensures that the batteries in a module will charge and discharge at similar rates improving the longevity of the pack. PrISUm Solar Car currently does not have a reliable method for performing this characterization.

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.

1.4 Requirements

Main Controller:

Will provide power to each characterizer node and utilize a serial communication to transfer information. The main controller will have an internet connection to send all test information to a central database. At this point, we are still evaluating if we should design this unit or buy an off the shelf unit such as a Raspberry Pi.

This unit will also have hardware-implemented error handling, such that if any of the batteries enter an unsafe state, hardware and software will independently be capable of shutting down the test program.

Module Unit:

This is where the testing will take place. The current goal is 8-10 batteries per module that will be characterized at once. 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 local test controller.

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 8-10 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. Hobbyists: Many hobbyists 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 necessary to match similarly characterized cells in parallel.

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 10 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 test data.

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 send the data to our database for storage.

2.2 Design Analysis

After quickly concluding that the original project would not be viable for the client, we have recently changed the scope of our project. Due to this change, we have done very little design work.

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. **Error! Reference source not found.** 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.

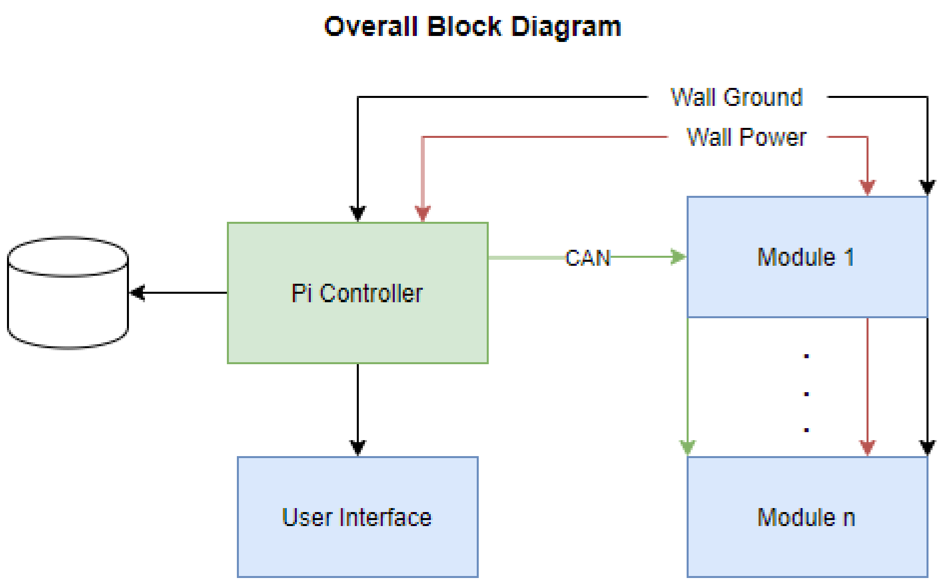

Figure 1: High-Level Block Diagram.

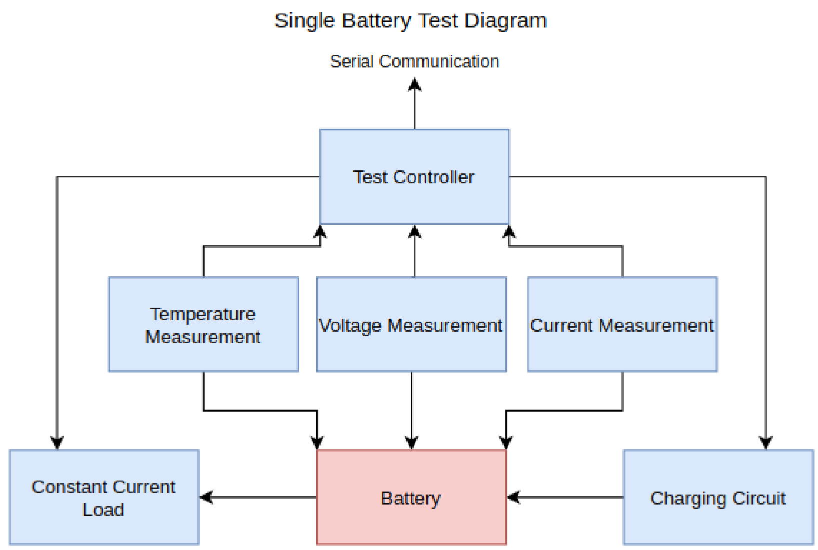

Figure 2: Battery Test Diagram

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. 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. Concerning 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

Power Management Section

- The system will need to be able to use a transformer to provide the necessary power for running the test.

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

Joe DeFrancisco -- Team Lead and Hardware Designer. Hardware-level system design and integration.
Bryan Kalkhoff -- Digital Designer, microcontroller selection and hardware interfacing.
Ben Kenkel -- Embedded Developer, software test control and communication.
Ryan Willman -- Hardware Designer, develop safety monitoring hardware.
Kyle Czubak -- Embedded and Database Developer,
Connor Luedtke -- Hardware Designer, develop charging structure.

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.

3.5 Project Proposed Milestones and Evaluation Criteria

Key milestones would be the completion of the different subsystem designs, communication, charging, load circuit, sensing, and fault logic.

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 is implemented by using a Gantt chart seen in Figure 3.

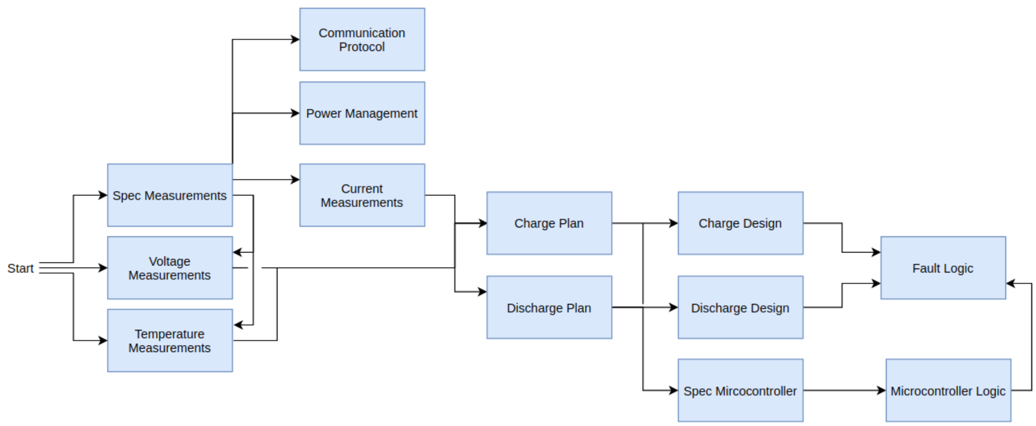

Figure 3: Project Gantt Chart

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.
TBA		
TBA		
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.
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.
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

As no design has yet been made, no testing has yet occurred. As the team creates sub-systems and when the team begins integrating sub-systems, we will be testing our 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.

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.

5.3.3 Hardware: Constant Current Electronic Load

The electronic load will be fairly simple to test. The op-amp, voltage reference, and current source will be supplied with X V(need to determine power rails). A multimeter will be placed in series with the 1Ω resistor to verify current accuracy. We will also verify device stability with an oscilloscope under steady-state, and switching conditions.

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.

5.3.5 Hardware: Temperature Sensing

5.3.6 Hardware: Digital Fault Monitoring

5.3.7 Hardware: Digital Communication

5.3.8 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.

5.3.9 Software: Web App

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.

5.4 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

– Explain how each method indicated in Section 2 was tested

– Flow diagram of the process if applicable (should be for most projects)

5.6 Results

– List and explain any results obtained so far during the testing phase

- – Include failures and successes
- – Explain what you learned and how you are planning to change it as you progress with your project
- – If you are including figures, please include captions and cite it in the text

  • This part will likely need to be refined in your 492 semesters where the majority of the implementation and testing work will take place

-**Modeling and Simulation**: This could be logic analyzation, waveform outputs, block testing. The 3D model renders modeling graphs.

-List the **implementation of Issues and Challenges**.

6. Closing Material

6.1 Conclusion

At the time of this report, our team has been primarily focused on project requirements, specifications, and timeline. We are slightly behind schedule due to the change of our scope, but we are confident in our abilities to create a functional design for this semester. Our team has a good understanding of what needs to be completed, and what we need to consider as we design and implement our proposed project solution.

6.2 References

This will likely be different than in the project plan since these will be technical references versus related work/market survey references. Do professional citation style(ex. IEEE).

6.3 Acronyms

AC: Alternating Current
IEEE: Institute of Electrical and Electronics Engineers
PCB: Printed Circuit Board

6.4 Appendices

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout, etc. PCB testing issues etc. Software bugs etc.