AC 2011-2268: AN OPEN HARDWARE, OPEN SOURCE ELECTRONIC LOAD BANK AND DATA ACQUISITION SYSTEM FOR EXPANDING THE NUMBER OF SCHOOLS AND STUDENTS RESEARCHING BATTERY ENERGY STORAGE

Tyler Grandahl
Gianfranco Biondi
Clark Hochgraf, Rochester Institute of Technology

©American Society for Engineering Education, 2011

An Open Hardware, Open Source Electronic Load Bank and Data Acquisition System For Expanding the Number of Schools and Students Researching Battery Energy Storage

Abstract

There is growing student interest in renewable energy and electric vehicles. Energy storage is a critical technology for electric vehicles and to some extent for intermittent renewable energy sources. Battery life, performance, and cost remain a barrier to widespread use of battery energy storage. Increasing the number of students researching these technologies would increase the likelihood of progress.

The process of testing batteries to characterize performance and durability typically requires the use of specialized equipment that may be beyond the budget for many educators. Hence, many interested students may not have access to the tools to perform these experiments.

To remedy this, we have begun the development of an open source battery-testing platform that can be built at a fraction of the cost of commercial systems. Furthermore, the development of the testing platform provides educational opportunities for students.

Introduction

In their 2010 National Energy Policy Recommendations, IEEE USA identifies energy storage as a critical technology for electric vehicles, the Smart Grid, and for intermittent renewable energy sources like wind and solar[1]. Many undergraduate and graduate students are interested in battery electric vehicles, plug-in hybrids, and renewable power systems that utilize battery energy storage systems.

In the case of electrified vehicles, the electric drivetrain has proven itself to be extremely reliable and efficient. However the electric vehicle must be economically viable in addition to clean and efficient. Typically Li-Ion cells are used for energy storage because of their high energy densities compared to other battery chemistries. For acceptable electric driving range, this requires a large battery pack that represents a significant percentage of the total vehicle cost. Because of the battery pack's high cost, its lifetime substantially affects the affordability of the vehicle.

In some applications, the aspect of battery cycle life has not been addressed as thoroughly as performance or cost. One reason is that life cycle tests can require months or years to perform at room temperature and require expensive load banks, chargers and data acquisition systems. In industry, the high cost of these systems reduces the number of tests that can be run. The situation is even worse for undergraduates who want to study energy storage, as typically even less funding is available to support them, regardless of their talent, enthusiasm, and dedication.

We believe the lack of affordable electronic load banks, chargers, data acquisition systems, and software to run these systems is one barrier to rapid progress in energy storage systems. In 2009 we began a project to develop an affordable open source, open hardware system for performing life cycle measurements on energy storage systems including batteries and ultracapacitors. This paper describes the system design philosophy, design choices, the initially targeted load cycle, and the integration of students into the development of the open source system.

System Requirements

The battery cycler is intended to work with individual cells of voltage ranging from 1.0 to 4.2 volts. This range covers popular battery types such as NiMH and Li-Ion. It is intended to allow for high current charge and discharge of cells 2- 5 amp-hours. Ideally, the cycler can be expanded to test from 1 to 100 cells. The cycler can also heat the cells to an elevated temperature and keep that temperature regulated with less than 1 degree C error.

Commercial Equipment Options

Several life cycle testing commercial solutions are available and offer a good set of features for data acquisition, selection of charge/discharge profiles, and high or low current discharge capability. However, the cost of these systems is high. Unfortunately, these systems can often not be scaled down to a single channel, so that entry cost is prohibitive for a single cell test.

Creating a Lower Cost Option – The Open Source Battery Cycler

In response to the difficulty of obtaining the capital equipment necessary to conduct battery testing, we have created the battery cycler project. The cycler integrates four elements similar to a commercial solution; data acquisition, cell load and charge control, environmental control, and host PC management software. It is a complete system designed to be open hardware that is scalable to meet the specific users needs of cell type, number of cells, and power. Because of this, the system architecture is capable of testing readily available production cells, as well as smaller coin sized test cells.

Unfortunately, open hardware does not mean free hardware. Nonetheless, battery testing equipment is a market where volume is low and margins are high. There are great savings to be had by building an open solution like the battery cycler. Our current proposed system budget suggests that we can cut the cost per watt of testing equipment to less than one quarter that of commercial systems.

The current status of the project is as follows: The system requirements and architecture have been defined. Hardware has been designed for data acquisition, load bank/charger, and the environmental control chamber. A communications specification for interfacing the Host PC to the battery cycler has been completed. Serial communication test software has been developed and we are currently in the process of validating prototype hardware. We are also looking for additional students to continue the development as current contributors graduate.

System Design Details

The initial design concept considered using a Labview based system. However it became clear that this course would make meeting our low cost goals from a software and hardware perspective difficult. Following our design philosophy we have chosen to build a system that is solely built around off the shelf components available from many online distributors. Avoiding test equipment all together and integrating its functionality into the loadbank/charger, environmental chamber, and data acquisition systems has allowed us to meet our cost and performance goals. Table 1 below shows the specifications for the baseline system being developed.

Table 1. Specifications of Loadbank/Charger and Data Acquisition System.

Parameter	**Minimum**	**Nominal**	**Maximum**	**Units**
Channels		2		
Charge current per channel	0.005		20	Amp
Load current per channel	0.005		20	Amp
Profile step resolution		100		mS
Data Sampling Rate		10		Hz
Data Recording Rate		1		Hz
Commanded current resolution		5		mA
Measured current resolution		2.5		mA
Measured voltage resolution		250		uV

System Architecture

To allow for scalability and to cater to individuals with small volume needs, the battery cycler systems are built as an array of independent test systems. Each independent system connects to a common Host PC for control and data acquisition via USB. Each system functions on its own without requiring other units, or a central power supply. Figure 1 illustrates this topology.

Each battery cycler device uses its dedicated microprocessor to perform all data acquisition, charge and load control, environmental control, and communications with the Host PC. This leaves the Host PC with only the responsibility of recording acquired data and occasionally sending devices commands to take a specific action.

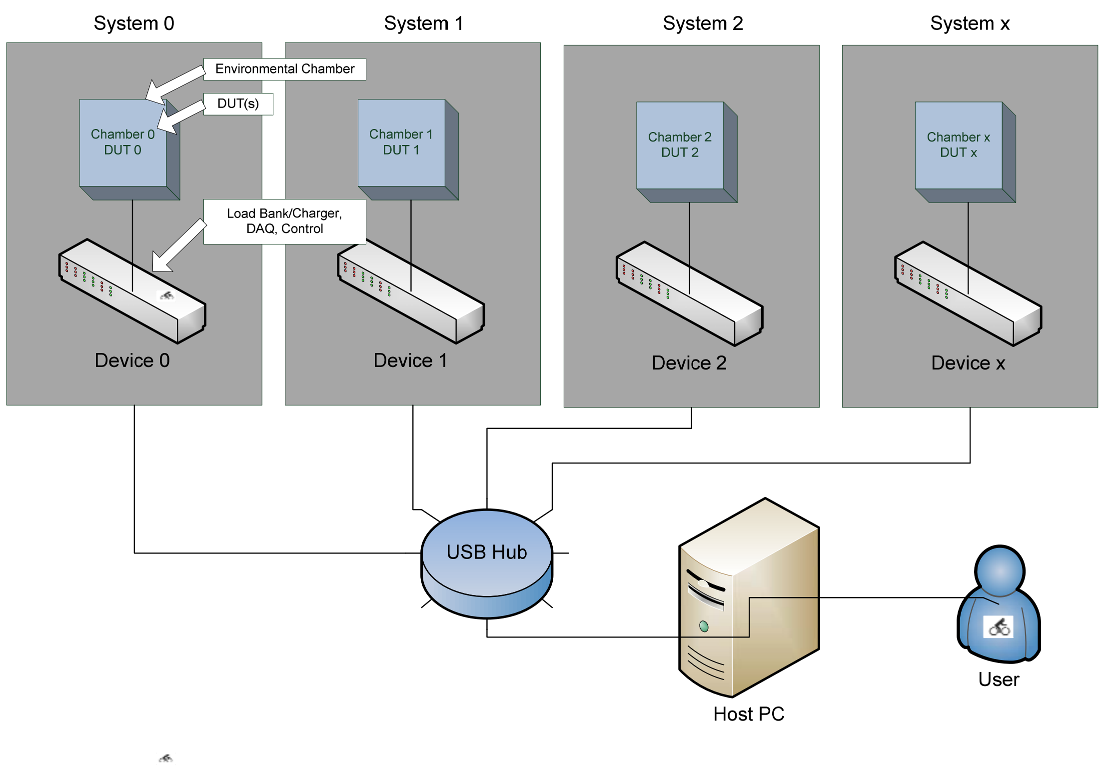

Figure 1 System architecture.

Channel Scalability

Many commercial testing solutions are scalable, however only within the limits of their base systems power supply and controls. For example, a minimum system may require buying a base capable of 8 channels. Often when only small scale testing is initially desired, the cost of the base system alone may be prohibitive.

The battery cycler fully integrates the charging power supply and charging control within each device. Each device only requires power and USB connectivity to a Host PC for control. Thus multiple cyclers can be connected through a USB hub.

Power Scalability

Battery experimentation occurs at a number of different power levels. While the initial intentions of the battery cycler platform are to focus on common production cells in the 2Ah to 5Ah range, it is scalable above and below this. Each cell channel is constructed from a number of current sources in parallel that make up that channel's current handling capacity. Having an open system allows a user to precisely define the capacity of a channel depending on their resolution and power needs. In addition, multiple channels within a single system can be combined in parallel to increase power for a single test.

Environmental Control

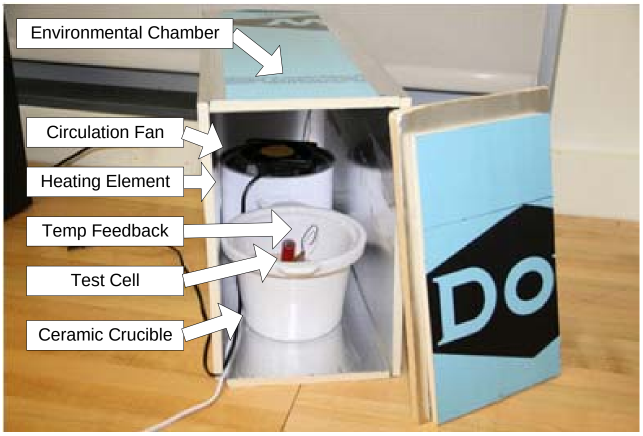

Figure 2 Photo of the environmental chamber, showing the cell under test.

Life cycle testing can often take months to years to perform at ambient temperatures. Because of this, cells may be tested at elevated temperatures to accelerate degradation mechanisms. Because of this environmental control is a core function of the battery cycler. The cycler provides an integrated PID control loop, temperature feedback, and triac to switch heating elements and closely regulate temperature. The environmental chambers construction can be tuned to suit the needs of a user. Figure 2 shows an example small thermal chamber with a 100W heating element, circulation fan, and ceramic containment vessel for cells under test.

Safety

Safety is paramount when considering testing that will be pushing high energy density cells to near failure. A failure mode analysis was performed to illustrate the considerations required for failsafe performance of the system. A robust communication protocol, as well as hardware watchdog, cell fusing, temperature monitoring, and software definable electrical and thermal limits are in place to protect cells from a certain range of potential failures. In the event of failure, cells are tested in a large container of sand to contain any potential thermal energy release.

Data acquisition

All measurements are sampled at 10Hz and reported and logged at 1Hz providing measurements of cell voltage, current, temperature, cumulative energy in and out, as well as a state of charge and test progress indicators. This raw data can be post processed to determine important cell characteristics such as ESR.

Supported Load Profiles

The system provides the capability to stress and characterize critical parameters of cells. To aid this effort and due to the system being developed with testing for hybrid electric vehicles in mind, the system is designed to implement standardized testing procedures such as those defined in the DOE Battery Test Manual for plug-in hybrid electric vehicles[2]. These include Charge-Sustaining and Charge-Depleting cycle life testing profiles, Hybrid Pulse Power Characterization testing profiles, Constant power discharge, and standard charging profiles.

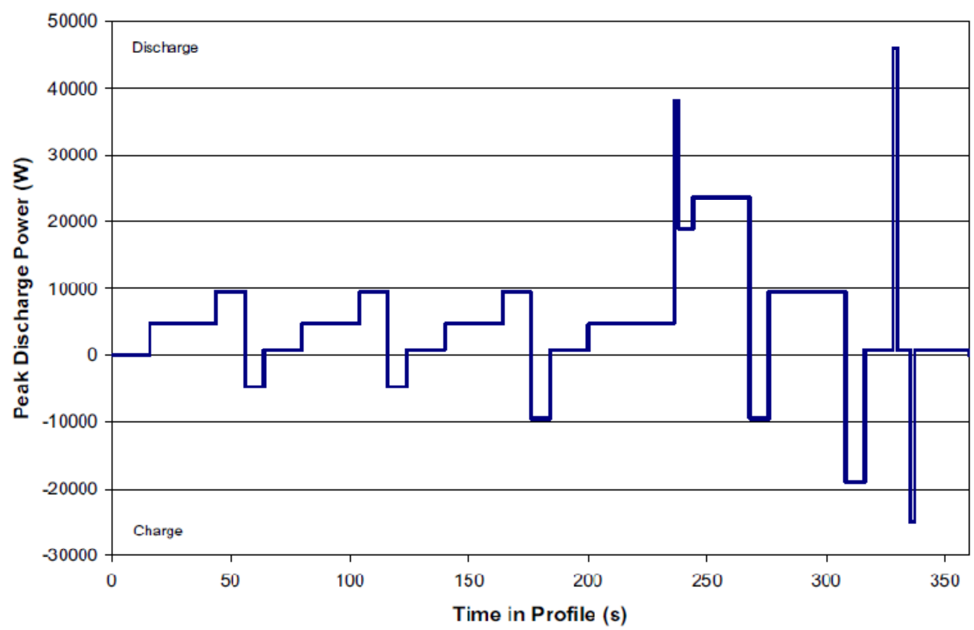

Figure 3 Charge-Depleting Cycle Life Test Profile for the Maximum PHEV Battery from INL Battery Test Procedures[2]

These standardized charge depleting (Figure 3), charge sustaining, and characterization tests (Figure 4) are built into to the devices firmware. Performing cycle life testing following these standards is merely a matter of defining scale factors, safety limits, and duration.

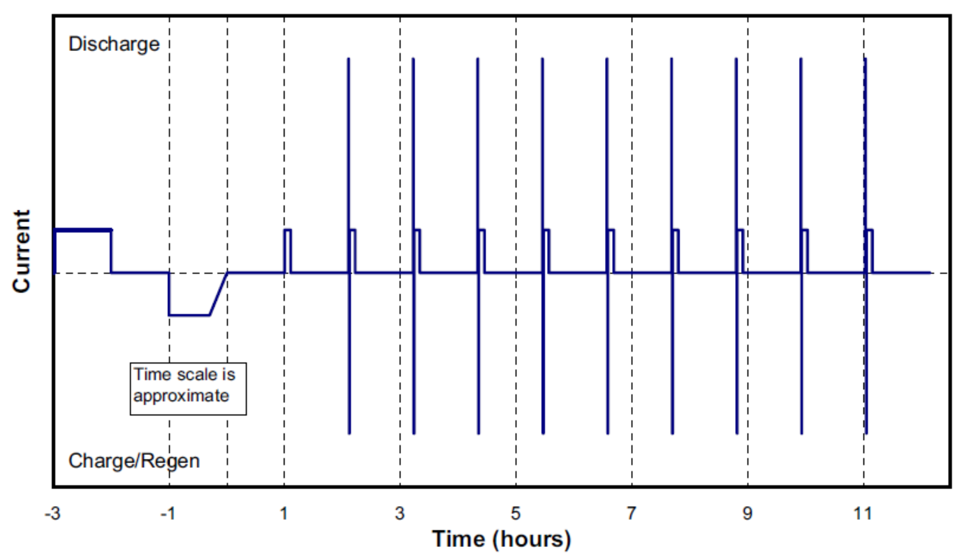

Figure 4 Hybrid Pulse Power Characterization Test (complete HPPC sequence) from INL Battery Test Procedures[2]

In addition, users are provided an interface for defining new cell load profiles and test sequences. The system is designed to operate in a variety of modes that can load the cell maintaining a constant current, voltage or power with definable limits of time or electrical conditions.

Design Philosophy: Open Hardware

Students interested in battery development come from a variety of backgrounds, e.g. material sciences or electrical, computer, and mechanical engineering. Because of this the assumption cannot be made that contributors will have access to or experience with certain development tools. Thus employing free and easy to learn development tools is essential to make sure that potential collaborators are able to take part. One of the most important elements of the project has been to keep the bar to entry as low as possible.

The development of the system requires specialty development tools in several areas. Both the Device and Host PC Software require complete development environments for debugging and compiling of program code. Because the device itself requires custom hardware to be developed, circuit schematic entry and Printed Circuit Board layout software is required. A central repository that is easily accessible to all collaborators for storing completed work and documentation is also essential.

By satisfying these needs using freely available development tools such as Arduino, Eclipse, and Eagle we are able to ensure that everyone that would like to participate in the project will have the ability to do so.

Arduino

Arduino is an open-source microprocessor platform that is at the heart of the battery cycler device. Arduino has set the standard for open hardware. It takes the success of open software and brings it to the hardware development world. Arduino's success stems from building on the wiring and processing programming language platforms that use abstracted C and Java programming, allowing people who do not necessarily understand how computers and microcontrollers work (artists, non-electrical/computer engineers) to use them. We look to emulate this model of an open platform in the battery cycler, thus building with Arduino for the embedded control of the battery cycler hardware was a natural choice.

JAVA + Eclipse

The Host PC software for control of the Battery Cycler is built on JAVA. It is available free under the GNU General Public License and is a convenient and common language for many technical majors. JAVA is often considered one of the easiest and most powerful programming languages because of its extensive use of vast libraries similar to Arduino.

Code comments can be compiled automatically into web-page-like document known as JAVADOCS. These documents allow collaborators to understand the functionality and working of others code. This is an important aspect when many collaborators may not be able to easily contact earlier contributors.

There are many free and open development environments that support JAVA. Eclipse is one of the most feature rich and popular. Its support for revision control (through plug-ins) allows many geographically separated contributors to share code by storing and merging all files in a central location through the Internet. The code sharing site is listed at the end of this paper. Users can easily work on different areas of the programs code simultaneously and submit them to a single final copy with ease. Revision control also automatically provides a history of submitted revisions.

Repository for Project Documentation and Code: Google Code

In a collaborative environment, having an accessible repository is just as essential as having good code development tools. Google runs a free project hosting service called Google Code that supports any Open-Sourced project. For an open platform to be truly successful it needs to allow its users to make improvements and share those improvements with the rest of the open source community. Working within a Google code repository helps achieve this goal.

The hosting service provides revision control supporting both Subversion and Mercurial. Revision control is a crucial tool making collaboration within the team convenient, but also making collaboration possible on a larger scale. It provides the ability for all collaborators to keep their work safe within Google's repository. The website also provides a bug-tracker system, a wiki for documentation and a file-download feature for distribution of released software and hardware documentation.

Eagle

The battery cycler relies to a great extent on hardware external to the Arduino development board. The system also requires load and charge banks to manipulate the cell. Developing the hardware to do this requires schematic entry and Printed Circuit Board (PCB) layout capability. Eagle by Cadsoft provides a professional grade solution free for nonprofit use. Supporting eagle allows us to publish schematics and board design in a format that other contributors can easily access and manipulate for free.

Project Documentation

A great amount of time has been spent documenting the architecture of the system and the layout for software and hardware development. This has been essential to preparing setting up a collaborative project and paving the way for the work that needs to be done. For example, a user manual for the battery cycler has been created. It includes information needed for other developers to make additions and enhancements to the project. Integrating this into a wiki within the provided Google code structure allows us to make the documentation part of the collaborative process.

All these tools have allowed us to develop a project that is available to any and everyone that has an interest in it and is willing to contribute.

Integration of Students and Engineering Education

The battery cycler is a multidisciplinary project. It allows students who need hardware for battery testing to participate in the development of that hardware. It provides tangible design experience that is invaluable for some students preparing to enter the job market. To-date, students from software engineering and electrical engineering technology have participated.

One of the students was recently hired at his first choice company. He cited his experience on this project as the decisive factor in getting an offer and the one topic that all of his interviewers were interested in. Another student has received an employment offer from industry due to his experience on the project.

A graduate software engineering student created the initial framework for the Host PC software, discovering many of the major challenges that would be encountered in the USB serial communication. In addition, two electrical engineering technology undergraduate students began work developing the load and charge banks that would interact with test cells, as well as defining the external hardware needed for data acquisition, output control, and safety considerations.

The project began back in 2009 as a proposal for a summer undergraduate research project developing the foundations what is now the battery cycler project. This was done with the following goals in mind:

- Cost under $200/test configuration
- Independent control of cell charge/discharge rate
- Control of cell environmental conditions
- Ability to monitor cell response to loading and log cell health related data.
- Redundancy in case of power failure

The project was broken into several major subtasks: loadbank/charger hardware, data acquisition and system control hardware, Arduino software development, Host PC software development, and environmental chamber design. A great deal of time was spent defining specification for the subsystems and developing an efficient and robust communication protocol. This partitioning was done to allow independent work tasks.

The battery cycler is a project that provides a platform for learning about battery test methods, for doing research on battery cells, and for learning about the design of hardware and software systems, as used in the battery tester.

For example, the environmental control provides a classic example for tuning the PID control for different chamber and heating element designs, an excellent problem for a Control systems class. The load and charge banks are ideal for an electronics class exploring power supply design, and a controls class exploring Bode loop gain analysis of the system's stability, and total bandwidth. The embedded environment provides a project for basic microcontroller programming and studying the systems architecture. The host PC software is built in JAVA, a language that reaches across many disciplines in the sciences.

Students have led the overall system design, the partitioning of tasks, the subsystem design tasks, the development of documentation and the work to make the code available for open-source development. The faculty role has been one of consultation, on an as needed basis. The faculty set deadlines, participate in design reviews, and set deliverables for independent studies.

The Future / How to Join

We have created the framework of a multidisciplinary project that provides a battery research tool for educators.

Open systems can have drawbacks. It is difficult to maintain consistency in contributed work. In addition, the success of an open project is limited not only by the project's user base but also the participant's willingness to contribute and grow the project. How this will work out remains to be seen.

A considerable amount of testing and development is still needed to bring the battery cycler to release version one. There are many ideas for improvements and additions to the system but these are on hold until the basic system is released. We invite collaboration with other students and institutions to move the project forward.

For more information regarding the battery cycler project visit the project's Google code website (http://code.google.com/p/battery-cycler/). This website contains project documentation as well as contact information for current project members. If you would like to contribute, contact:

Dr. Clark Hochgraf

cghiee@rit.edu

References

1. IEEE-USA. "National Energy Policy Recommendations." (February 2010): 10-11. Web.
2. *Battery Test Manual For Plug-In Hybrid Electric Vehicles*. Idaho National Laboratory, INL/EXT-07-12536, March 2008. Web.