# Abstract

# 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 renewable energy sources like wind and solar. Many undergraduate and graduate students are interested in battery electric vehicles, plug-in hybrids, and renewable power systems that utilize battery energy storage systems.

The electric drive train 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 there high energy densities compared to other battery chemistries. For acceptable electric driving ranges this requires very large battery packs that represent a very significant percentage of the total vehicle cost. Because of the packs high cost its life time substantially affects the affordability of the vehicle.

Cycle life, energy density, and cost are key factors in the successful application of battery

energy storage. In many applications, the aspect of cycle life has not been addressed as

thoroughly as performance or cost. One driving reason for this is that life cycle tests can

require months or years to perform at room temperature and require very expensive load

banks, chargers and data acquisition systems. The cost of these systems is prohibitive for

graduate studies unless significant funding is available from a grant or existing assets can be used. 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 a barrier to rapid progress in energy storage systems. Therefore 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. It is hoped that this will lead to further discussions and exchange of ideas about how to develop and apply open hardware in a variety of educational settings.

# Our Solution

## The Battery Cycler

In response to the difficulty of obtaining the capital equipment necessary to conduct battery testing, we have created the battery cycler project. A solution that integrates all four elements of testing as a commercial solution would, data acquisition, cell load and charge banks, environmental control, and a host PC software. The system is implemented in dedicated but open hardware that is vastly scalable to meet the specific type and size of testing the user plans on conducting.

The scalability of our system means it is capable of testing full size and readily available production cells, as well as smaller coin sized test cells.

Built around the Arduino embedded platform the Battery Cycler provides a common USB interface and expandability within the hardware.

## System Capabilities

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. In addition we provide an interface for defining cell load profiles and testing sequences by the user. 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.

Data is reported and logged at 10Hz providing Cell voltage, current, temperature, cumulative energy in and out of the cell, 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,

Each system as currently designed provides two cell channels per system that have configurable charge and load bank power and power supply configurations. This makes system cost even easier to control for systems that do not have a demand for high power for testing full size cells. In the event that more power is required both channels of a system can be combined for use on a single cell.

The load and charge bank configurations are scalable to meet the user’s needs.

|  |  |
| --- | --- |
|  | C:\Documents and Settings\tng5461\Desktop\IMG_0007.jpg |
|  | |

## Development Philosophy

One of the most important elements of the project has been to keep the bar to entry as low as possible. By using free development tools such as Arduino, Exlipse, and Eagle we are able to ensure that everyone that would like to participate in the project will have the ability to.

Arduino has set the standard for open hardware, taking the success of open software and bringing it to the hardware development world.

Arduino has done this by building on the wiring and processing platforms that use abstracted C and Java programming allowing people who do not necessarily understand how computers and microcontrollers work to use them. We look to emulate this model of an open platform in the battery cycler, thus using the Arduino platform within it was a natural choice.

Battery development attracts a variety of backgrounds, many focused in material sciences or chemical engineering. Thus employing free and easy to learn development tools is essential. The assumption that users will already have access to and experience with them cannot be made.

Unfortunately, open hardware does not mean free hardware. None the less, battery testing equipment is a market where volume is small and margins are high, there are great savings to be had by building a kit solution like the battery cycler. We estimate that we can cut the cost per watt of testing equipment to nearly a tenth of many high end commercial systems with our open design.

-Eagle

The battery cycler relies to a great extent on hardware external to the arduino development boards. The system also requires load and charge banks to manipulate the cell. Developing the hardware to do this requires schematic entry and board layout capability. Eagle by cadsoft provides a professional grade solution for free for nonprofit uses. Supporting eagle allows us to publish schematics and board design in a format that other contributers can easily access and change for free.

JAVA-

JAVA is an excellent software platform for open source projects. JAVA is available free under the GNU General Public License, which keeps it free (with a few exceptions in the core of the framework of the JVM). ~~SUN offers the JAVA Development Kit (JDK) for free through their website.~~ JAVA is heavily standardized and extremely well documented. SUN specifies a standard for code comments which makes JAVA source code easy to understand and implement, and compiles web-page-like document known as JAVADOCS. ~~SUN provides a tool for JAVA which automatically takes the comments in the code (comments following the SUN JAVA Commenting standard) and compiles them into a web-page-like document known as JAVADOCS.~~

JAVA is based syntactically on C and C++, two of the most commonly used programming languages. The main difference is JAVA uses object-oriented paradigm and JAVA comes with a Garbage-Collection system, which eliminates memory leaks and makes programming significantly easier. The object-oriented paradigm adopted by JAVA inheirently allows for the growth and upgrade of libraries without having to directly change them through inheritance. Inheritance means that a programmer can create a new JAVA class, and say it inherits from another class. This gives the new class all the functionality of the parent class as well as whatever new features the new class offers. JAVA also has tons of libraries available, offering many advanced features to even novice programmers.

JAVA was designed with Threading, distributed computing and security in mind. JAVA is an interpreted language and as such JAVA code is highly portable thanks the Java Virtual Machine (JVM). JAVA code can run anywhere a JVM is supported with little to no modification. The JVM significantly simplifies programming by handling things such hardware drivers and memory management for the programmer, which reduces the complexity an average programmer has to deal with and allows them to focus more on the functionality of their code rather than the operation of the hardware running the code.

There are many free and open development environments. Eclipse is one of the most feature rich and popular, and it supports JAVA development as well as many other languages including C/C++ and web-applications. Eclipse supports many plug-ins which make development for many different platforms easy. Eclipse supports the use of SVN through plug-ins and allows for check-ins and check-outs directly through the program’s interface. Eclipse supports advanced coding features such as auto-formatting and executable generation. Both the environment and plug-ins are very well documented.

Google Code –

Google runs a free project hosting service for all Open-Sourced projects, Google Code. The hosting service provides revision control supporting both Subversion and Mercurial, revision control is a crucial tool to provide the foundation laid to the masses of people looking to implement the open source hardware and software. This service makes collaboration within the team convenient, but also will make collaboration on a larger scale possible. The website also provides a bug-tracker system, a wiki for documentation and a file-download feature.

For an open platform to be truly successful it needs to be allow its users to make improvements and share those improvements with the rest of the open source community. We believe that working within a google code repository will help us achieve this.

DOCUMENTATION

A great amount of time has been spend documenting the architecture of the system and the layout for software development, this has been essential to fostering a successful collaborative project and paving the way for the work that needed to be done. Integrating this into a wiki within the provided google code structure allows us to take this documentation itself to a collaborative position.

# Involving students

As an integrated solution by way of a multidisciplinary project, the Battery Cycler has allowed the students that need the hardware to participate in the development of it.

This process began back in 2009 as a proposal for a summer undergraduate research project developing the foundations of the Battery Cycler project. 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

It was immediately clear that the project needed to be broken into several major components, Device software development, Host PC software development, and device hardware development.

During the course of the summer three students joined the project, a graduate Software Engineering student jointed and formed the initial framework for the host PC software discovering many of the major challenges that would be encountered communicating without hardware. In addition two electrical engineering technology undergraduate students began work developing the load and charge banks that would interact with test cells.

A great deal of time was spent defining constrains for the many different sub systems that are involved. Learning how to develop a efficient and robust communication protocol as well as dealing with many of the headaches of building a system that is robust enough to run for months without crashing.

While students move on, the project is refreshed with new students from different programs that are interested in participating as independent studies and payed graduate work. The project continues to include students with a broad range of interests and experience that are willing to contribute and learn.

# Ideal for education

The battery cycler is a diverse project that provides not only an excellent platform for the testing and education of battery technologies, but many other engineering disciplines as well. An education environment can benefit from the projects as it offers many classic engineering examples in an applied setting.

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 system class. The load and charge banks provide an excellent project for an electronics class exploring power supply design, and loop analysis contributing to the systems stability, total bandwidth, and optimizing to meet the systems needs. 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.

# What we have done so far, what comes now.

We have drawn the framework for a powerful multidisciplinary project that reaches across many disciplines and provides a solution to a choke point in battery research for educators.

We have grown the project to revision one, a point where it meets our original goals and we can begin the testing that originally inspired the project. We are extremely excited to have reached these goals and be able to release our work into the wild.

With the amount of work that was required to bring the battery cycler to rev 1, its only natural to be confronted with countless ideas for improvements and additions to the system along the way. We are excited to further grow the platform with new innovations and collaborate with other students and institutions as the project grows.

The need

Arbin, Bitnnnaaaa, and others make excellent battery testing systems and have dominated the market for some time. However the exorbinate cost of such systems leave them out of reach of

* Possible approaches
* Choice
* Related design decisions
* Labview vs. arduino

Teaching benefits ,,,

* Writing paper

Labview vs. arduino, is mostly a cost issue, everything that has been done in the arduino platform could have done within a labview VI. Our approach mimics the design of a product, and provides more practical design experience that is invaluable to students that are about to be entering the Job market.

Practical design experience

* Something that is invaluable