4-2016

Incorporating Energy Related Concepts into EE and CS Laboratory Work and Coursework

Jack Adams
Merrimack College, adamsj@merrimack.edu

Christopher S. Stuetzle
Merrimack College, stuetzlec@merrimack.edu

Incorporating Energy Related Concepts into EE and CS Laboratory Work and Coursework

John Adams
Department of Electrical Engineering
Merrimack College

Christopher Stuetzle
Department of Computer Science
Merrimack College

During the course of this interdisciplinary effort, members of the Electrical Engineering (EE) and Computer Science (CS) departments collaborated on energy related curricular efforts. Initially work was carried out to develop and utilize an inexpensive, open-source system for measuring, storing, and displaying energy related data from across campus. Hardware and software components chosen were open source or free for educational use. A low power Linux server was utilized. The LAN-enabled Arduinos included sensors to measure energy related quantities such as power and temperature. EE and CS students were engaged in various aspects of the project – EE students focused on the hardware, CS students focused on the programming. EE junior students worked with clients to implement real world measurement and display solutions. A CS student project focused on developing a JavaScript-based web page that visualizes sensor data by leveraging CanvasJS and JQuery packages. This web page development project will continue in spring 2016 as the work is significantly incorporated into the CS department's Software Engineering and Information Technology Systems classes. Most recently, EE junior projects (fall 2015) emphasized collaborations across a wide variety of disciplines: projects include wetland environmental factors (Biology), greenhouse environmental factors (Biology), pump energy usage (ME), weather monitoring (Physics), and classroom temperature monitoring (Facilities).

Corresponding Author: Jack Adams, AdamsJ@Merrimack.Edu

Introduction/Background

In this paper we start with a "background and motivation" section: what were the original goals of this work, how did they evolve over time, and how eventually did the department of Computer Science become involved? We continue with a "Methods" section describing the underlying technology for much of the work. Web-enabled Arduino microcontrollers are used to send sensor data to a LINUX server for storage and display. We next describe the Computer Science and Electrical Engineering (CS/EE) collaboration. Some of the project results follow, including CS/EE results. A section on "Assessment" follows. The final section includes conclusions as well as a discussion of ongoing and future planned work.

Background and Motivation: The work described in this paper began as an EE initiative and evolved over time to include faculty and students from the CS department. The EE initiative began with the creation of a course with significant power, renewable energy, and project content titled "Energy Generation, Conservation, and Technology" first offered in spring 2013. The course was developed for several reasons, including:

- As a response to weaknesses identified in the EE department's 2012 program review – in particular weak debugging skills, and less than optimal preparation for the Senior Capstone project,

- the department's determination that a junior level project-oriented course is needed,
- and the first author's strong interest in Energy Efficiency and Renewable Energy (EERE), and belief that EERE is an important and timely topic.

The course has a significant emphasis on renewable energy, and the comparison between renewable and fossil based generation. With its emphasis on renewables the course also addresses ABET criteria on ethics (f), global impacts (h), and contemporary issues (j).

There are of course numerous energy monitoring solutions on the market, both commercial and open source. Lucid technologies has a solution with an excellent GUI[2]. There is the Open Energy Monitor project, which uses Arduino boards with energy monitoring shields to sample and report a wide variety of energy related quantities[3]. The Open Energy Monitor project is a complete system, with various server options, Arduino shields geared towards energy measurement, and full software distributions for both the web and database server and the Arduino. However, from the perspective of the current work, each existing solution has its drawbacks: for example the Lucid technologies approach is purely a GUI and database solution, and the client needs to provide the data. The Open Energy Monitor Project is not configured for instructional purposes in a way that is suitable for a junior-level EE project course.

Methodology

Methods – General

The primary technology used in this project is the energy monitoring platform described in this section. The EE/CS collaboration centered around this platform. A high level sketch of the hardware/software platform is shown in Figure 1. In this case, a wind sensor is shown, although any of a wide variety of sensors could be used. The figure is from a student project implemented by Mr. Shawn O'Shea[4].

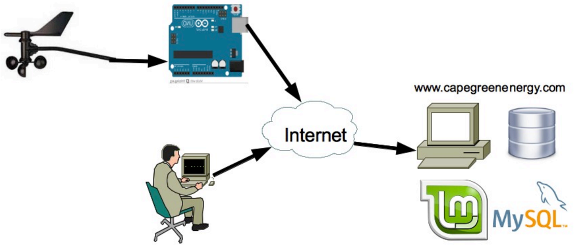


Figure 1: Project overview

The sensor is connected to an Arduino microcontroller. The Arduino takes sensor readings and connects to the www.capegreenenergy.com website via the Internet. The website is housed on a Linux server. Traffic from the Internet is directed to the router using the dyndns.com service. A webpage written in the PHP programming language takes the sensor values and saves them in a MySQL database devoted to the present offering of EEN3270 – each student is allotted a portion of the database. Once up and running, an end user with Internet access can view both real time and historical data.

The server: Physically, the server is a net top based unit that draws less than 15 watts maximum. The server is running what is commonly called a LAMP (Linux, Apache, MySQL and PHP) stack. Most readers will know that Linux is an open-source operating system that comes in many distributions, some free and others commercial. For this project, the free, open source distribution called Linux Mint is used. This distribution was chosen for its popularity, and has among its benefits a deep and instructional website, as well as many excellent tutorials on YouTube that include, for example, start to finish LAMP installation. The components of the LAMP stack provide the components for a free, open-source system that can provide dynamic, data-driven websites.

The visualization software chosen is CanvasJS[5], a JavaScript charting library that can work with JavaScript or Ajax to enable database access. The webpage itself is written in HTML, and includes one or more CanvasJS "containers," one for each plot to be displayed. CanvasJS was chosen for several reasons; first, although not itself open source, it is free for educational use under the Creative Commons Attribution-NonCommercial 3.0 License. Second, the CanvasJS website has many examples of how to incorporate its functionality into web pages, many of these live demos allow user interaction. Finally, CanvasJS is an actively managed distribution: questions posted on the software website are answered in a timely manner, and software updates are posted on a regular basis.

The microcontroller: The microcontroller platform employed is the Arduino UNO. Arduino is a highly popular open source hardware platform. It is well suited to be a core component of an instructional system: it has a fully-documented, easy-to-program, open source programming environment, and there are great tutorials available both in written form and in the form of YouTube videos.

The Uno is based on the Atmel ATmega328 processor/controller, an 8-bit microcontroller that incorporates an Arduino boot loader. Particularly useful for this project, the Arduino has built-in 10-bit analog-to-digital converters for sensor interfacing. While the standard Uno has no built-in networking capabilities, there are Ethernet shields based on the Wiznet W5100 Ethernet chip that enable full Internet connectivity, and these were employed in the current project.

Sensors: A fairly wide range of sensors were employed during the three offerings of EEN3270. Most projects employ temperature sensors, a common choice being the LM34DZ precision Fahrenheit IC. Students were familiar with the LM34DZ having used it in their sophomore course on embedded controller design. Another common choice for temperature sensing is the 1-wire Dallas Semiconductor DS18B20; these are highly accurate, multiple ICs can be placed on the same 1-wire bus, and there are existing Arduino 1-wire libraries.

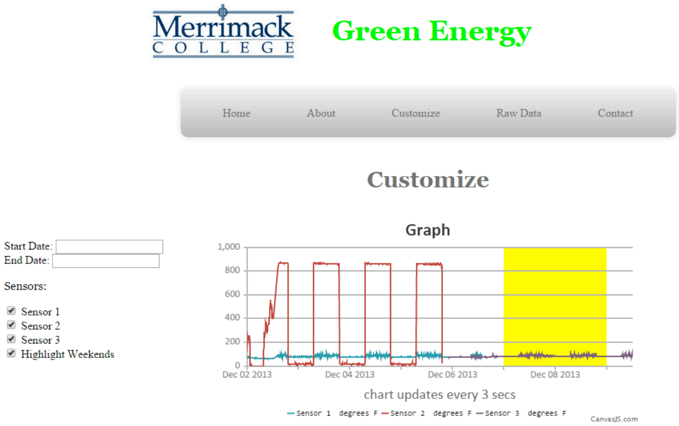
If the voltage is known, and the current is known, the power usage as a function of time is readily derived. A split core current transformer such as the SCT-013 is interfaced with the Arduino by utilizing code available in the Open Energy Monitor[3] website. Another popular sensor is the DHT11 temperature and humidity sensor; the code and example applications are available from Adafruit[6]. Other, less common sensor choices included photoresistor light detection circuits, the wind sensor, and a distance sensor (see Figure 1).

Methods – CS/EE collaboration

The CS/EE collaboration over the course of this work has taken two forms: EE faculty consulting with CS faculty to improve the software aspects of the platform, and then jointly directing a CS senior student in implementing new features during an independent study. While the initial work was an EE initiative, it was quickly realized that consultation with CS students and colleagues would be crucial. For example, in the mind of the first author, AJAX (**A**synchronous **Ja**vaScript and **X**ML) connotes a household cleaning product rather than an advanced web programming technology. Over time, consultations have including server setup, Arduino microprocessor programming, MySQL assistance, and more. This part of the collaboration has resulted in a productive working relationship between the departments.

The second form of the collaboration involved both EE and CS faculty advising a senior CS student in an independent study to design and implement the project's associated web site. The student explored and practiced with a variety of graphics packages and languages, ultimately landing on the CanvasJS and AJAX frameworks for use on the site. She designed and implemented the entirety of the site, with optimization and input from EE and CS faculty as appropriate. A major goal of this independent study was to produce a webpage that allowed a user to customize a visualization of the sensor data (housed on a local server). An example page, the Customize page, is shown in Figure 2. The page allows users to choose a start- and end- date for the data to be displayed, which datasets are visualized, and whether or not weekend dates are highlighted (here, seen in yellow).

The page was designed using jQuery selector functionality in conjunction with CanvasJS's charting capabilities, and all pages are built on the server using PHP. Data are imported to the page using JSON objects. The pages can be dynamically updated as sensor data are added. All of these technologies are freely available, as is the HTML/PHP source for the sensor data display pages.



Start Date:
End Date:

Sensors:

☑ Sensor 1
☑ Sensor 2
☑ Sensor 3
☑ Highlight Weekends

Figure 2: A screenshot from the Customization page. Users can select the date range for the data, which datasets are displayed, and whether weekends are highlighted (they can be seen in yellow in this figure).

Analysis

With the EE junior-level having been offered four times as of this writing (Spring 2013, Fall 2013, 14, and 15), over 40 projects have been carried out. For the most part the projects have been successful, with students getting to the point where they are uploading data to the server using their sensor-enabled Arduinos, for visualization via website. Example projects have included:

- Monitoring temperature, light, and humidity in the Mendel Science Center greenhouse.
- Monitoring temperature in the Chemistry Department’s NMR lab, so that the room temperature can be monitored off site to ensure at all times that is low enough for safe operation of the NMR system.
- Monitoring light in a Biology lab where fish experiments are run, to ensure that lights are turned off and stay off at nighttime, so that the fish

circadian rhythms are maintained as they are in nature.

- A weather station project, including wind speed and direction sensors

Example results from student projects are shown below.

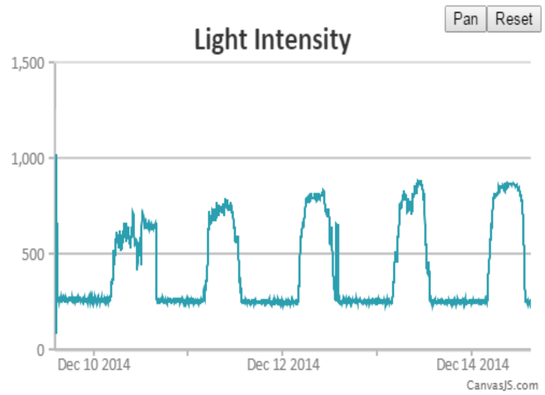


Figure 3: Office light intensity during a several day period in Dec. 2014

The "light intensity" shown in Figure 3 above displays light sensor readings over a several day period in an office experiencing wide temperature swings. The cloudiest day is December 10th, and the days get sunnier as time progresses. The daytime "notches" in the plots represent periods of cloudiness. In the same room, the temperature is shown for a two week period beginning on Dec. 12, 2014, in Figure 4. Hotter days can be seen in the earlier part of the period. The occupant observed sustainable practices, turning the heat down over the holidays. Zoom is a nice feature of CanvasJS, and in the plot to the left the temperature "detail" is shown for Dec. 13, 2014; the electric heat can be seen cycling on and off.

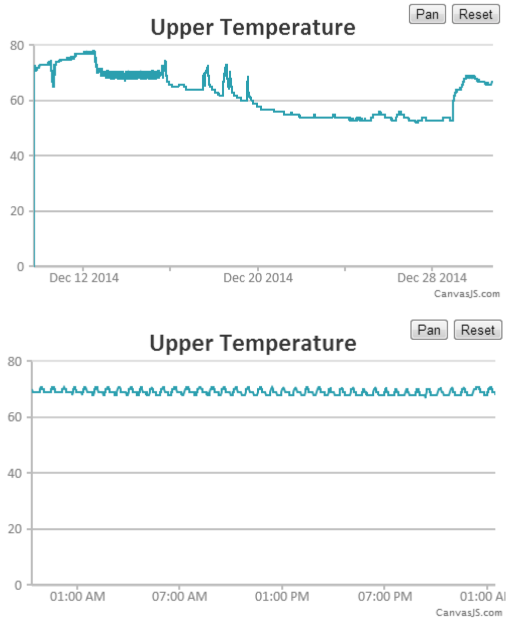


Figure 4 Office temperature for a two week period (above) and zoomed in (below) to show Dec. 13

In another project, the client was a Biology professor, and lab light intensity was measured using a photoresistor circuit. Representative data is shown in Figure 5: for this period the on/off light cycle was preserved in a manner that supports the natural circadian rhythm of the fish being studied.

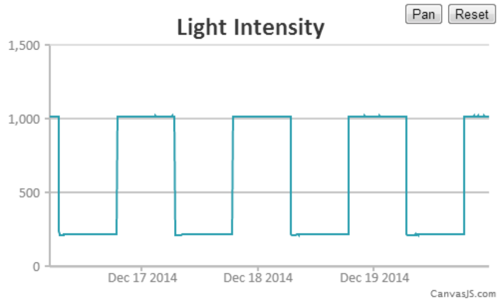


Figure 5: Fish Lab light intensity vs. time

Assessment methodology and results

There were various assessment methodologies employed throughout this work. These are described below.

EE assessment methodology: all EE courses require assessment of the course using "points of learning" methodology, which includes both instructor assessment and self-assessment by the students. Also in Fall 2015, evaluation by 3 EE faculty members, and 3 Advisory Board members of the final project presentations.

EE assessment results:
Course objectives were all met. Students can better identify and work with clients – there need to be deadlines for achievement of client objectives. Students can better identify the specific "problem" being solved in their initial project scoping and their presentations. Having presentations by junior students like these is very helpful (feedback from two alumni who were reviewers)

CS independent study assessment methodology: faculty review of student presentation. The student gave a final presentation seen and reviewed by EE and CS faculty. A user manual and project write up written by the student were reviewed by EE and CS faculty.

CS independent study assessment results:
EE and CS faculty were very satisfied with the amount and quality of the work done. The effort was seen as contributing significantly to the EE/CS energy monitoring and display "knowledge base". Most importantly, the foundational work developed in this first independent study has laid the groundwork for multi-semester cross-disciplinary work in the form of senior capstones and semester project, described in greater detail below in our plans for future work.

Conclusions

The project portion of the course has been a success, with significant room for improvement. In the EE program, first semester junior students still are used to predefined labs, and have trouble orienting themselves to a project environment. While most students are able to complete their work, they generally are finishing up within a day or two of the final deadline. In the latest course offering, all students had a "client".

Going forward, as a result of its most recent Program Review the Electrical Engineering department will be incorporating significantly more experiential learning projects both in and beyond the classroom into the curriculum. In addition, the department is committed to a significantly increased level of interdisciplinary work. A planned Senior Capstone robotics project that includes Computer Science, Mechanical Engineering and Electrical Engineering students will incorporate both the interdisciplinary and experiential attributes. Arduino based sensor networks with wireless functionality are a likely component of the Capstone, and the junior-level project described in this paper can be modified and extended to prepare students to be strong contributors in a robotics senior project.

The collaboration between CS and EE worked well, with significant functionality added by a highly motivated CS student within the parameters of a one semester independent study. The web page and sensor set-up and maintenance methodologies utilized in this work are publicly available, but in the future they will be readily accessible to any and all institutions that wish to repeat the procedure, including detailed documentation and specific instructions on setup for customization of individual web-pages. One goal is to automate installation of the web page via an install script which would allow users to simply run the installations script, answer a few questions, and have a working database and web page automatically.

Finally, available open source solutions are improving significantly; for example in the two or so years since this project began, the

Open Energy Monitoring[3] solution discussed above has become significantly more robust and deep in sensor and GUI options. The ongoing addition of new sensors, database and GUI implementations to the public domain will likely provide an increasingly rich set of tools available for students to draw upon in their projects. However, the system used in EEN3270 going forward will still be education-specific, and kept as simple as possible so that junior students can understand most, if not all, of the components.

There remain significant opportunities for EE/CS collaboration in the future. What is in place now is a "system" that works well. However, the system has been designed with the constraint that improved solutions for parts of the project that can be "swapped in" to improve the system as a whole. For example, CanvasJS works well, and perhaps a better solution emerges in the future - the technology for this type of software changes regularly. In spring, 2016, there is an ongoing project in the CS Capstone course that includes several improvements to the existing combination of technologies. For example, the database solution implemented by EE is not scalable: each sensor has its own associated PHP scripts, and its own table. This is OK for 10s of sensors, but not 100s or even 1000s that we envision over time. The students have “improvement of database solution so that it is scalable” as one of the deliverables.

We conclude by noting that, in terms of the value of and accomplishments of the interdisciplinary collaboration in this work, a solid foundation for future projects has been created. Students and faculty are benefiting. The authors consider this work to be ongoing and multiyear in nature.

References

1. Faculty Led Initiative: Powerful, Elegant, and Inexpensive Open System to Measure, Store and Visualize Energy Data, spring 2014.
2. Lucid Technologies /www.luciddesigngroup.com/
3. Open Energy Monitor Project: http://openenergymonitor.org/emon/
4. Shea, Shawn Arduino Data Logger with Davis Wind Sensor, Final Project Report for EEN3270 Merrimack College, 12/2013.
5. This software is free for educational use. User manual and software at www.CanvasJS.com
6. For this sensor see: Adafruit http://www.adafruit.com/product/386