



Lab and Team Project Development for Engineering Problem Solving using MATLAB, with Emphasis on Solar Power and Engineering for Sustainability

Mr. Stanley W. Hsu, University of California, Davis

Stanley Hsu received a B.S. degree from University of California, Los Angeles in 2006 and an M.S. degree from University of California, Davis in 2011, both in Electrical Engineering. He is currently pursuing a Ph.D. in Electrical & Computer Engineering at UC Davis. His research is in the area of low power mixed-signal integrated circuits for energy harvesting sensor applications. He is also interested in high-speed wireline communication circuits, high-performance VLSI designs, and signal integrity.

Prof. Rajeevan Amirtharajah, University of California, Davis

Rajeevan Amirtharajah received the S.B. and M.Eng. degrees in 1994, and the Ph.D. degree in 1999, all in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA. From 1999 to 2002, at High Speed Solutions Corp. (later Intel), Hudson, MA, he developed high performance memory buses. He is currently an associate professor at the University of California, Davis, where his research focuses on low power microarchitecture, circuit and interconnect design, energy scavenging, and signal processing for wireless sensor nodes. He received the National Science Foundation CAREER award in 2006. He is also a visiting scholar at the Berkeley Wireless Research Center for the 2012-13 academic year.

Dr. andre knoesen, Department of Electrical and Computer Engineering, U C Davis

André Knoesen received his B.Eng. degree in electronics engineering from the University of Pretoria, South Africa, in 1980 and the M.S.E.E. and Ph.D. degrees from the Georgia Institute of Technology, Atlanta, in 1982 and 1987, respectively. He joined the Department of Electrical Engineering, University of California, Davis, in 1987 where he is now a Professor. He performs research in biosensors, and materials for high-frequency electronic and optoelectronic applications. Dr. Knoesen is a fellow of the Optical Society of America.

Lab and Team Project Development for Engineering Problem Solving using MATLAB, with Emphasis on Solar Power and Engineering for Sustainability

Introduction

Engineers apply fundamental scientific principles to invent and design solutions where the constraints in the problem are met. However, in an age where traditional energy sources are limited, engineers must design solutions that not only solve the problem at hand, but they must also be capable to design solutions that have minimal, or even zero energy footprint. Energy footprint is defined here as the total energy required to implement and to maintain a solution for a particular engineering problem. To engineer a minimal energy footprint solution, engineers must understand and be capable of making tradeoffs that reduce energy footprint; this requires an understanding of sustainability concepts, specifically the different modes of renewable energy, their fundamental operations and cost-effectiveness.

To effectively educate current engineering students in sustainability early in their program of study without modifying degree requirements, we revise the curriculum of an engineering problem solving course (Engineering 6, or ENG6) at UC Davis to incorporate topics in renewable energy. ENG6 is a required lower division course that teaches basic programming concepts to electrical, biomedical, mechanical, aeronautical, and civil engineering students. Because computer science and computer engineering students are not required to take this course, most students have little familiarity with programming. Some are also not motivated to master programming concepts, since they view software design as being outside the core competencies required for their chosen disciplines. Increasing student engagement was another motivation for developing the new curriculum, including the creation of hardware-based exercises and projects described in the sections below.

The course utilizes MATLAB programming exercises and projects to teach engineering problem solving skills. In a regular quarter, the course typically has 150 to 250 registered students encompassing freshmen to fifth year seniors, resulting in a wide distribution of the student demographic. ENG6 is a platform that allows the largest number of students to be reached in the shortest amount time.

In ENG6, four major components determine a student's grade: weekly homework assignments, weekly computer laboratory sessions, two projects, and a final exam. Homework assignments are designed to help student mastery of technical programming concepts and MATLAB language mechanics. The multiple choice final exam is employed to test student understanding of fundamental MATLAB programming concepts. Lab sessions in traditional offerings are optional and are usually used as extended TA office hours or open lab hours. However, lab sessions are required in our revised curriculum. We use it as a proctored assessment environment to test student understanding of the lecture material; in addition, we design the lab exercises around the theme of sustainability, to introduce those concepts and hopefully increase student engagement with the programming material as well. Example problems ranges from sizing a solar panel array based on expected demand, financing for a photovoltaic array, and modeling of solar panel.

Finally, the two projects are also sustainability themed. The first project is an individual project where students analyze weather data for a number of U.S. cities to evaluate alternative locations for renewable energy production while the second project is an open-ended final team-based design project where the problem statement and constraints are loosely defined, allowing students to exercise creativity. Courses that teach sustainable engineering tend to focus mainly on teaching sustainability concepts, e.g. through service learning⁷, instead of emphasizing the development of problem-solving skills. Butler⁸ describes a course that teaches sustainability concepts within the civil engineering and environmental science majors; however the focus was on educating students in specialized topics within the civil engineering field. Our approach is to use sustainable engineering to motivate students from a wide range of engineering disciplines to acquire problem-solving and programming skills.

Although inspiration for the curriculum is drawn from topics in sustainable engineering, our current effort emphasizes solar energy for its potential to become a low cost energy production technology due to its steadily rising efficiencies. The Shockley-Queisser limit puts the theoretical efficiency of a p-n junction silicon solar cell at 33.7%¹ while the highest reported efficiency today is at 27.6%². However, the efficiency of more complex multi-junction concentrator solar cells is around 44%². While solar energy most likely won't completely replace conventional energy sources, it is an essential technology that improves sustainability. An example is the Robert Mondavi Institute for Wine and Food Science at UC Davis which utilizes a 120,000 kWh³ photovoltaic array to complement other means of renewable energy generation.. This initial course development effort starts by incorporating technical solar energy related topics for the previously stated reason. However, additional technical topics to be incorporated into the curriculum include (but are not limited to) hydroelectric and wind energy generation. Ecological and economic issues due to sustainable engineering will also be incorporated as the curriculum evolves based on the assessment process.

Course Flow and Programming Exercise Development

Inspirations for the weekly lab problems come from the Robert Mondavi Institute for Wine and Food Science at UC Davis. The institute is a LEED Platinum certified teaching and research facility. LEED stands for Leadership in Energy and Environmental Design, and is a certification awarded by the U.S. Green Building Council. Some important sustainable features of the building includes rainwater and process-water capture and reuse for landscape irrigation and toilets, renewable energy production from roof photovoltaic panels, daylight “harvesting” and occupancy sensors, CO₂ sequestering, and others³.

The curriculum for this class has three main student-centered learning objectives. First, to introduce students to an engineering approach to problem solving (e.g., extracting what is the ultimate desired result from a problem statement, determining known and unknown quantities, and proposing and executing a solution to the problem). Second, to familiarize the students with using computer programming to develop problem solutions, as well as the mechanics of software design and implementation for addressing engineering problems. MATLAB is chosen because of its relatively clean syntax and wide use in the engineering curriculum as well as in academic research and industry. Third, to introduce students to sustainable engineering concepts around the theme of renewable energy.

The course starts by spending the first week on one-dimensional arrays and the next two weeks on two-dimensional arrays, and then moves into flow control, loops, and logical operators for two weeks. This order is chosen because vector and matrix processing concepts are fundamental to the MATLAB syntax and build on mathematical foundations students should have already seen in physics and mathematics classes that touch on linear algebra. The second half of the course begins with one week on custom functions, cell arrays, and strings, and one week on numerical computing and curve fitting. Finally, the remaining three weeks in the quarter is distributed between graphical user interface (GUI) design, object-oriented programming (OOP), and pattern matching using regular expressions. These final topics can be adjusted at the discretion of the instructor, for example to include block diagram simulation instead of pattern matching.

Because this course is designed for freshmen, some of whom are taking calculus concurrently, and has no prerequisite programming courses, the curriculum only assumes knowledge of trigonometry and no prior programming experience. The introductory nature of the course thus limits the mathematical depth of the homework exercises, in-lab assessment problems, and projects. One-dimensional arrays are taught by designing problems that require the student to perform algebraic manipulations to compute an unknown at multiple input values. Two-dimensional arrays are taught by having students solve problems using built-in MATLAB functions that operate on rows, columns, and diagonals of matrices. Flow control, loops, and logical operators are introduced to the students by having lab exercises where the output of the program/script depends on user input. Problems on custom functions require students to implement simple algebraic expressions as standalone/anonymous functions. To teach basic numerical computing built-in functions that implement the quadrature and trapezoidal method of numerical integration are introduced. Curve fitting is also demonstrated. Concepts in GUI and OOP are taught by having the students work with the predefined **arduino** class object; having students define an object from scratch and then use that object to perform a task, all within a single lab session, is too ambitious for this course's audience. Cell arrays and strings are emphasized throughout the course by asking students to display information about the result of a specific problem.

Each weekly in-lab assessment exercise contains two problems. The first problem challenges the students by requiring them to utilize techniques they should have mastered from previous labs to solve a new problem, while the second problem teaches a new concept/technique that the student should add to his or her skillset. Problems that teach a new concept/technique are tutorial-like, with step by step instructions on constructing each line of code that culminate in a working MATLAB script, while problems that challenge the students are posed as general problem statements without step by step instructions; students are expected to utilize their growing technical knowledge and problem solving skills to identify the required steps in solving the problem and implementing those steps in MATLAB code. Table 1 below shows the programming language (MATLAB) topics and the corresponding sustainability problems designed.

Table 1. Topics and corresponding types of problems designed.

Programming Language (MATLAB) Topics	Types of Problems Designed
1D arrays	Solve for average temperature given degree-day data, for multiple cities.
2D arrays	Modeling a solar panel using rows and columns of photodiodes.
Flow control, loops, logical operators	Compute the total cost of a solar panel array while incorporating volume discount.
Custom functions	Write a function that computes the power of a solar panel given the open circuit voltage and the short circuit current of each photodiode, and the number of PD in each row and column.
Cell array and strings	Display information about the result of a problem.
Numerical computing	Newton's method for solving a nonlinear equation such as the current-voltage equation of a photodiode.
Curve fitting	Comparing performance between fixed and tracking solar panels by extrapolating the curve fitted from voltage and current data.
GUI and OOP	Gather luminance data using the Arduino Solar Module (see section below).
Plotting	Display results. This topic is emphasized throughout all labs.

Arduino Solar Module

The purpose of the Arduino Solar Module (ASM) is to give students hands-on experience in hardware control and data gathering. Having experience handling and controlling hardware increases student interest and engagement in class materials. Large quantities of modules are required to accommodate the large class size of ENG6, requiring the module to be low cost. Details on the hardware and software design, including the bill of materials are outlined in this section.

Hardware Design

The ASM consists of a custom daughter board (from the Arduino Proto Shield kit) mounted on top of the Arduino UNO. The module communicates with a laptop by emulating a serial port using the USB interface, and it is also powered through the USB interface. The daughter board contains a connector that connects to a 24 mm by 22 mm monocrystalline silicon solar cell, a digital potentiometer, an open-circuit jumper, and headers. The headers interface to the pins on the Arduino UNO. The solar cell output connects to the DS1803 digital potentiometer, which acts as the load for the solar cell. The DS1803 chip has two 10K potentiometers, which are connected in series for a total possible load resistance of 20 kOhms. Each resistor is controlled by an 8-bit code via a 2-wire interface. The open circuit jumper disconnects the load from the solar cell

allowing the open circuit voltage to be measured. Figure 1 below shows a top view of the module with different components labeled, and Figure 2 is the connection diagram of the module.

The cost breakdown of the ASM is shown in Table 2. Each unit costs around \$46.38 where a large portion of the cost comes from the Arduino UNO. It is possible to replace the Arduino with a lower-cost microcontroller platform, such as the TI Launchpad, which is based on the MSP430 microcontroller. Replacing the Arduino with Launchpad reduces the unit cost to \$17.72. Additionally, to increase the mobility and versatility of the solar module, the wired USB interface can be replaced with a wireless interface, such as the TI EZ430-RF2500T wireless target board. Adding the wireless interface and a 9V battery brings the unit cost to \$39.70.

One concern is the maximum solar cell current. The solar cell is able to produce up to 20mA of current under direct sunlight while the maximum continuous wiper current of the DS1803 is only 1mA. As a result, one of the potentiometers in the DS1803 must be used to limit the solar cell current if the module is to be used outdoors. An alternate solution is to use a potentiometer that can carry a larger continuous wiper current, such as the AD5161.

Due to class time limitations, the modules are provided directly to the students with no assembly required. However, the circuit components and prototyping area are left exposed to allow the option of modifying the hardware for later projects.

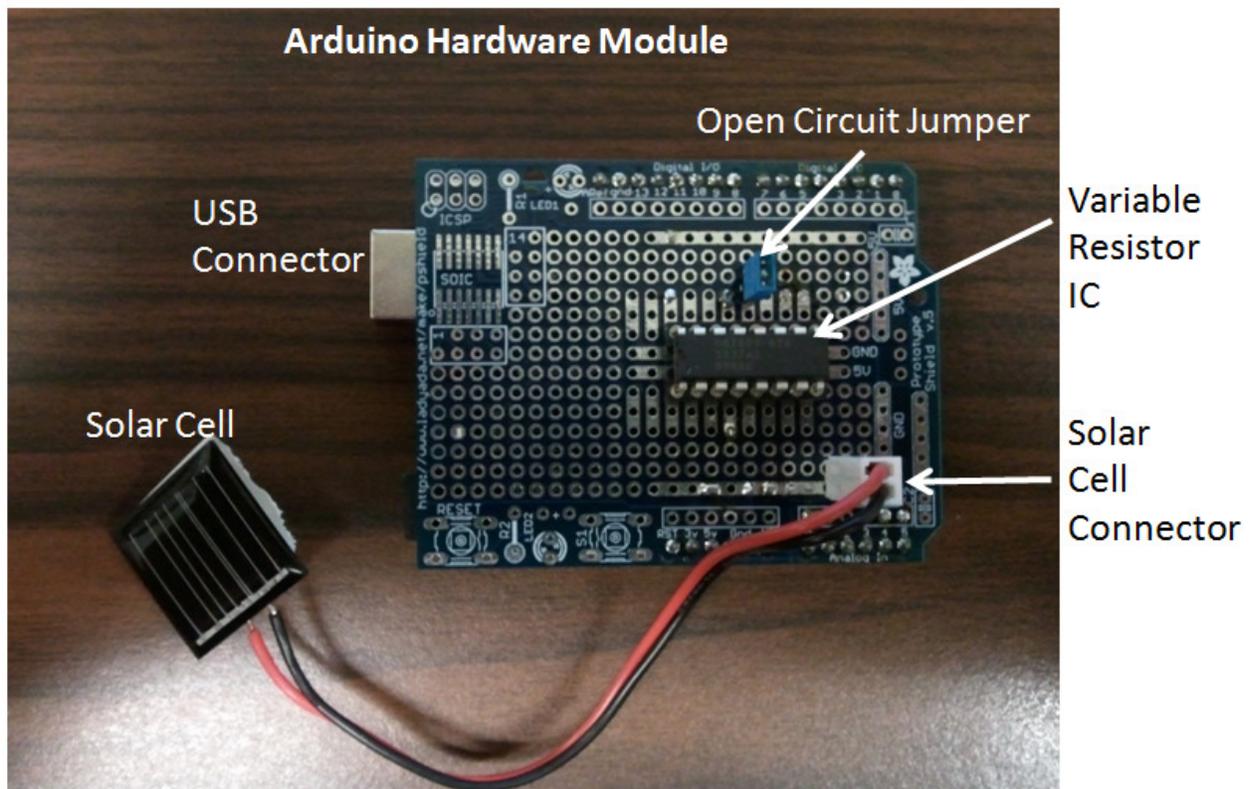


Figure 1. Arduino Solar Module with parts labeled.

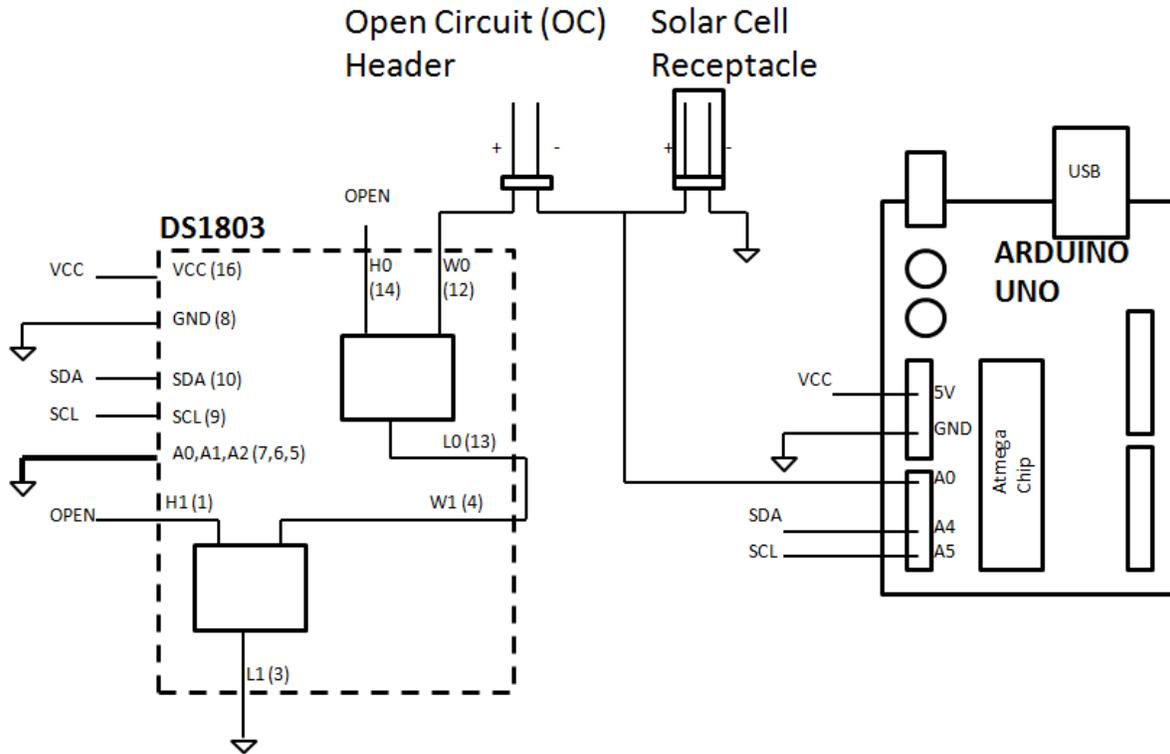


Figure 2. Connection Diagram of the Arduino Solar Module.

Table 2. Bill of Materials for one ASM unit.

	Unit Cost
Arduino UNO	\$26.96
USB Cable	\$1.67
DS18B10 Potentiometer	\$3.22
Solar Cell	\$3.15
Connector Receptacle, Header, and Crimp	\$0.15 + \$0.09 + \$0.06
Arduino Proto Shield	\$11.00
Jumper	\$0.08
Cost of One Unit	\$46.38

Software Design

There are two main software components in the system: code (“server”) running on the Atmega microcontroller on the Arduino UNO and the MATLAB arduino object class definition. Both components are modifications of the original code⁴. The Arduino language is written in C/C++ and has a set of lower level functions that allow the programmer to configure, read, and write to and from the pins. The arduino object class in MATLAB simply “brings” the lower level function in the Arduino language into MATLAB by first defining the function in MATLAB, then

transmitting a command (arbitrarily defined) and any input arguments to the server. When a command arrives at the server, the server calls the equivalent lower level function to execute a task. Since the goal of ENG6 is not teaching microcontroller programming or interfacing with hardware, all necessary code is provided for the students such that the only tasks they need to do is simply creating the arduino object and operating on the object to control hardware and sample data, all within MATLAB.

Modifications to the server and the MATLAB arduino object definition are required to interface to the digital potentiometer. On the server, the built-in Wire library is used to interface to the potentiometer via its I2C port. On the MATLAB arduino object definition, methods are added to allow the two potentiometers in the DS1803 to be individually controlled. A block diagram of the MATLAB software interaction with the Arduino hardware is shown in Figure 3.

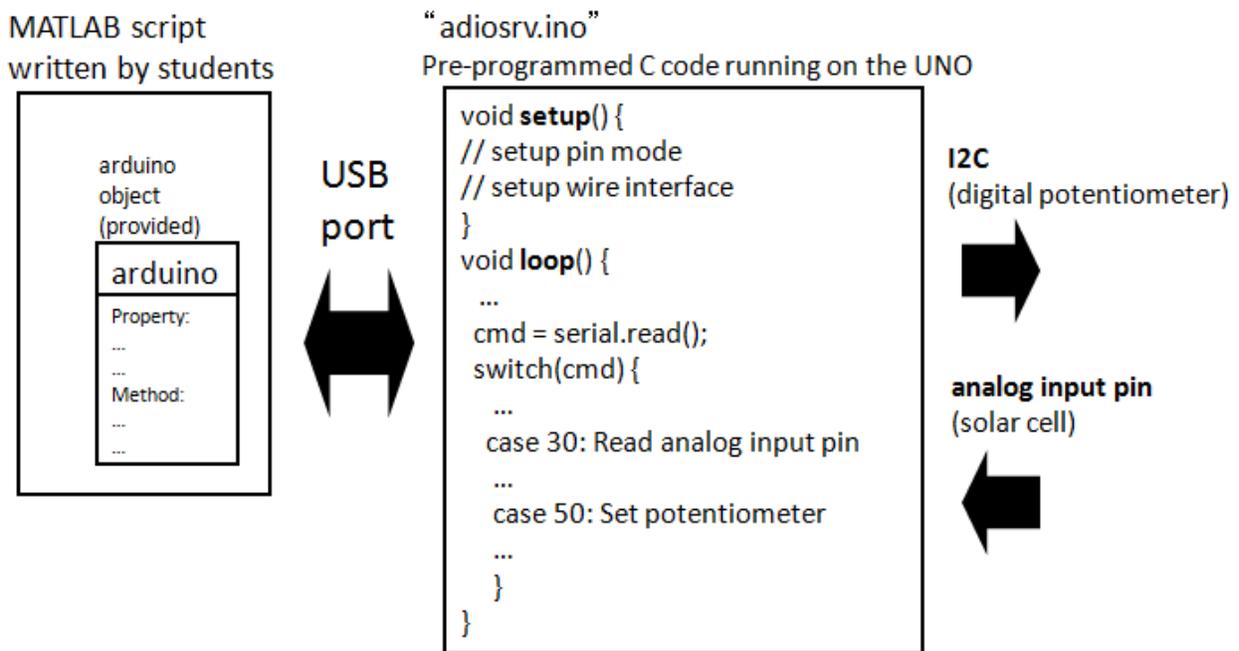


Figure 3. Software interaction of the whole system, MATLAB to the hardware.

Module Operation and Sample Lab Exercise

The ASM allows the load resistance to be kept constant or adjusted, allowing students to measurement light intensity, or characterize the solar cell by looking at its current-voltage (I-V) curve. A sample I-V curve is shown in Figure 4a below. The figure contains three curves at various illumination levels; the top curve corresponds to the greatest illumination, and curve shifts down as illumination decreases. Figure 4b below shows the solar cell’s output to a periodically flashing light.

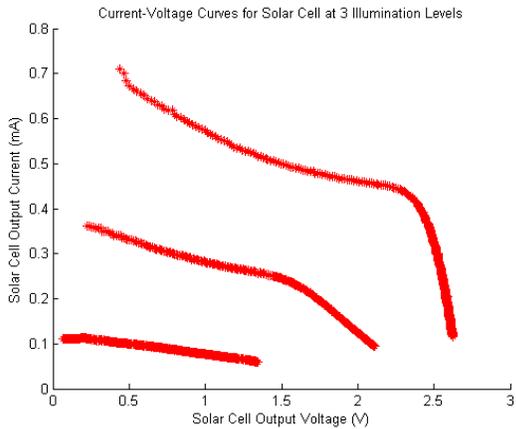


Figure 4a: Current-Voltage curve for solar cell at three illumination levels (decreasing illumination from top to bottom), obtained using the ASM.

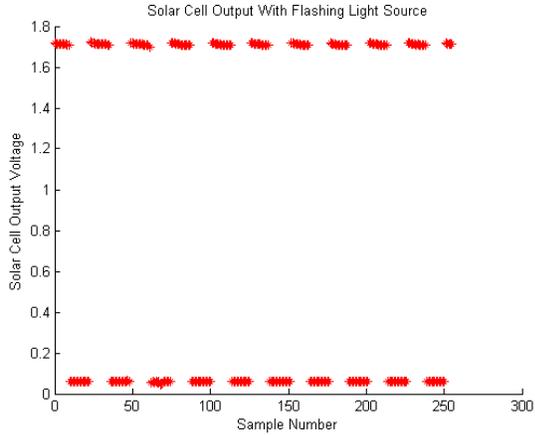


Figure 4b. Solar cell response to flashing light.

During lab sessions, each student receives an ASM pre-programmed with the server and the MATLAB arduino object definition. Each student is asked to obtain data using the module, at three different light intensities. For example, the first data set is obtained while a flashlight shines directly on the ASM (red curve in Figure 5). The second data set is obtained without the flashlight (blue curve in Figure 5), and the third data set is obtained with the ASM covered (green curve in Figure 5). Students are also asked to compute statistical properties of the three data sets.

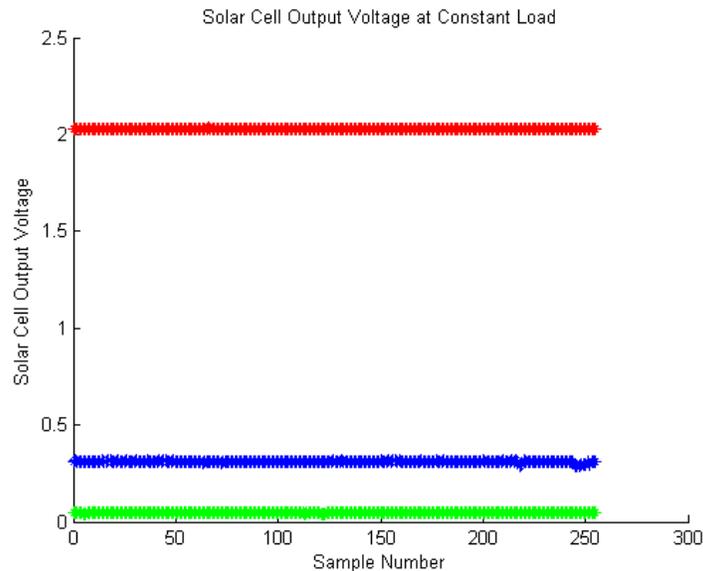


Figure 5. Solar cell output at different constant illuminations. (Red: with flash-light. Blue: without flash-light. Green: ASM covered.)

Project Development

Individual Term Project

The first project in the course is an individual project that guides the student in writing code that imports and analyzes a climate database. The database contains weather and demographic information for 54 U.S. cities, for 12 months: monthly weather information such as wind speed, solar radiation, and precipitation, and demographic information such as population, land area, and city name are in the database. Sample project tasks include finding the number of unique cities, cities with the most solar radiation, highest wind speed, and most amount of precipitation. The project then moves onto comparing the suitability of each city for solar energy, wind energy, and rainwater collection. The project also asks students to create a program that finds the rank of each city for the amount of renewable energy from each of the three sources. The students output the data as a table, sorted based on the user's preference input.

In addition to the tasks outlined above, an earlier version of the project also asks students to compute the correlation between precipitation and solar radiation of a city. Additionally, students learn simple GUI development and user interaction. In one task, students are presented with a U.S. map and are asked to create a database of boundary coordinates for various states, using the built-in MATLAB function for graphical input 'ginput'. In a later task, students use that database to determine where the user has clicked on the U.S. map, and displays weather information for all cities within that state.

The goal of the term project is to reinforce MATLAB programming concepts such as element-wise operations on vectors, flow control constructs, and loops by guiding the students in completing each task in a specific manner. The goal is such that upon project completion, students will be able to easily recognize the appropriate MATLAB programming construct for solving engineering problems.

Team-based Final Project

This is an open-ended team design project where groups of three students design and implement a GUI-based program. The goals of the project are to further student's problem solving abilities, teach team work, and encourage creativity in engineering problem solving.

For many students, the final project may be their first exposure to working on a team to solve a design problem. Consequently, parts of two lectures are devoted to introducing the engineering design cycle as an iterative process that begins with developing a specification, followed by brainstorming possible solutions and selection of an initial design concept after discussion and tradeoff analysis. The next phase includes developing a prototype (primarily software in the case of this course), testing, and refining the design. After teams are formed and assigned to specific project topics, a preliminary report is required from each team member before they begin implementing their design. Two components are required in the report: team milestones and the individual's contribution that allows each team milestone to be achieved. For each team milestone, division of tasks and an expected completion date must be specified, and the associated contributions from each individual must be outlined. The objective of the preliminary

report is to force teams to complete the first phase of the design cycle (specification, brainstorming, initial design concept) before engaging in the second phase.

At this stage in the class, students are expected to be comfortable with designing and debugging in MATLAB, and are capable of utilizing web resources to learn about new topics. With the above expectations, the project topics are developed. Table 3 contains project topics previously offered:

Table 3. Project topics and description.

Project Topic	Topic Description
Renewable Energy Planning and Design Tool	Design a GUI that allows the user to drop various solar panel and wind turbine models onto a canvas containing the map of a city. The program should allow the user to save their work and continue later. The program should also create a bill of materials based on the design.
Solar Survey	Use the Arduino Solar Module to gather sunlight data at various locations, then create a contour plot of the data and overlay it on top of the corresponding map. Design two GUIs: the first GUI helps with data gathering by allowing its user to quickly record the location, time, and trial number. The second GUI allows its user to select the data set (specify time of day, trial number, etc) to be plotted, then plots the contour plot of the data.
Carbon Dioxide Emission from Traffic	Create a GUI that simulates traffic of 20 cars traveling at different speeds, across five lanes. The program should display the instantaneous fuel consumption, carbon dioxide emission, and speed of the 20 cars.
Global Solar Radiation Analysis	Create a GUI that allows the user to graphically input an arbitrary area/region on earth. The program then calculates the area of the region using different numerical integration methods. Finally, the program calculates the potential solar power available from that area given the specification of a solar panel and the global solar data ⁵ .

Upon the conclusion of the project, each team is required to submit a written report and a video presentation (via YouTube). The written report contains the team's design and implementation process, sample screen-shots, flow-charts, instruction on how to use the program, and work breakdown across the team members. The video presentation is selected here since live

presentations not a practical option in large sized classes. In the video, each team demonstrates core features of their program and discusses their algorithm and important parts of the code. Finally, a teaching assistant tests their program for correct functionality and grades the project based on the quality of the work submitted.

Assessment

The assessment process was inspired and modeled after previous work introducing active learning through hands-on laboratories utilizing low cost hardware platforms for controls courses, which were previously mostly theoretical⁶. Assessment of the new curriculum proposed in the present work was performed by asking the students to complete an online survey (Google Forms) at the end of the course. Students were asked to rank various objectives and outcomes, on a Likert scale relative to their experience in other engineering courses. Objectives and outcomes related to problem solving skills, engagement in course materials, experience with the solar module, and comfort with MATLAB were a few of the questions in the survey. 403 students from four quarters were surveyed and 105 responses were received. The survey results are summarized in Table 4 below. In general, the revised curriculum appears to be more effective relative to the students' (limited) prior experience, with little measurable change from quarter to quarter. To achieve more effective results, weekly lab exercises may need to be tailored each quarter such that the difficulty of the exercises does not exceed the student's skill set. Doing so ensures that students are challenged, but also remain engaged at the same time. Importantly, exercises should not be so difficult that students are unable to finish in the allotted time; this decreases their confidence in problem solving and reduces morale. Additional student self-assessment to test this hypothesis is planned for this quarter's offering of ENG6.

Table 4. Summary of Student End-of-Course Evaluations.

	1 - Much less effective	2 - Less effective	3 - Similar	4 - More effective	5 - Much more effective	
Development of problem solving skills.	8	9	31	34	23	Mean: 3.52 Std. Dev.: 1.15
Ability to design and implement an algorithm to solve a given problem.	12	11	27	35	20	Mean: 3.38 Std. Dev.: 1.23
Ability to define a problem and specification.	9	11	36	29	20	Mean: 3.38 Std. Dev.: 1.16
Engagement in course materials.	11	18	20	46	10	Mean: 3.24 Std. Dev.: 1.16

Hands-on experience with electronic hardware.	9	13	23	24	36	Mean: 3.62 Std. Dev.: 1.3
Comfortable using MATLAB to solve problems.	10	10	28	29	28	Mean: 3.52 Std. Dev.: 1.25

Summary and Future Directions

A new curriculum for an introductory engineering problem solving course is designed and implemented around the theme of sustainable engineering. Lab exercises and project topics are developed to increase student awareness of sustainable engineering, specifically in renewable energy, while developing problem solving skills. To give student hands-on experience in working with hardware, the Arduino Solar Module is designed. The hardware module allows students to perform simple data gathering and hardware control through MATLAB.

Based on the assessment results, the newly developed curriculum is on the right track but still need to be improved. To increase effectiveness, lab problems towards the beginning of the course need to be re-formulated such that more specific instructions are given. Additionally, lab problems should not be made too difficult to the point where students are unable to complete the exercise. We also plan to follow the students through upper-division classes that have ENG6 as a prerequisite to evaluate whether the revised curriculum has translated into improved student mastery of the topics introduced here.

Future upgrades to the ASM include replacing the Arduino UNO with the TI Launchpad for reducing cost. Additional modifications include moving from the current wired interface to a wireless interface using the TI-EZ430-RF2500T kit; a wireless module improves the versatility of the hardware and allows networking-related problems to be developed as lab exercises or project topics. Finally, sensors other than the solar cell could be incorporated onto the module if an analog-front end circuit is designed to adapt the dynamic range between the sensor output and the MSP430 microcontroller analog-to-digital converter (ADC) input range.

Acknowledgement

The authors would like to thank ENG6 teaching assistants in Summer 2011 and Winter 2012 quarters for help with deploying the new material. We thank Shane Austin for helping with lab and project development, and Karen Dunn-Haley and Mikaela Huntzinger from the UC Davis Center for Excellence in Teaching and Learning for help with the assessment plan. We would also like to thank Giampiero Campa from Mathworks. Finally, we thank Mathworks for supporting this work through a curriculum development grant.

Bibliography

- [1] Shockley, William and Queisser, Hans J., “Detailed Balance Limit of Efficiency of p-n Junction Solar Cells”, Journal of Applied Physics, Volume 32 (March 1961), pp. 510-519.
- [2] National Renewable Energy Laboratory, “Best Research-Cell Efficiencies”, http://www.nrel.gov/ncpv/images/efficiency_chart.jpg.
- [3] Nola, Julianne, “What Makes us LEED Platinum”, <http://greenrmi.ucdavis.edu/news/leed/what-makes-us-leed-platinum>.
- [4] Arduino Support from MATLAB, <http://www.mathworks.com/academia/arduino-software/arduino-matlab.html>.
- [5] NASA Surface meteorology and Solar Energy (SSE) Release 6.0 Data Set (Jan 2008) 22-year Monthly & Annual Average (July 1983 - June 2005), http://eosweb.larc.nasa.gov/sse/global/text/global_radiation.
- [6] Borgstrom, Per Henrik, et al. “Science and Engineering Active Learning System: A Novel Approach to Controls Laboratories”, ASEE Annual Conference, 2012.
- [7] Bielefeldt, Angela, et al. “Evaluation of Sustainable Engineering Education Via Service Learning And Community Service Efforts”, ASEE Annual Conference, 2010.
- [8] Butler, Elizabeth, “Changing Students’ Knowledge and Attitudes About Sustainable Development and Sustainable Engineering in an Introductory Science and Engineering Class”, ASEE Annual Conference, 2012.