Seattle
Making Value for Society

122nd ASEE Annual
Conference & Exposition
June 14 - 17, 2015
Seattle, WA

Paper ID #13942

Letting students learn through making mistakes: Teaching hardware and software early in an academic career.

Dr. Derrick Rodriguez P.E., Colorado School of Mines

Dr. Rodriguez is a Teaching Associate Professor at the Colorado School of Mines in the Mechanical Engineering Department. He has taught over 30 courses and specializes in Thermal/Fluid Systems.

Prof. Jenifer Blacklock, Colorado School of Mines

Dr. Jenifer Blacklock is the Assistant Department Head in the Mechanical Engineering department at Colorado School of Mines. Jenifer is active in the Undergraduate Curriculum in the Mechanical Engineering department and is an advocate of using hands-on-learning tools to help develop strong math, science and engineering foundations.

Dr. Joel M Bach, Colorado School of Mines

Associate Professor, Mechanical Engineering, Colorado School of Mines

Letting students learn through making mistakes: Teaching hardware and software early in an academic career.

Introduction

It is difficult in today's society to teach students that it is indeed acceptable to fail at something, as long as you can learn from that initial (or many) attempt(s). Academia is also criticized that not enough effort is spent teaching students hands-on skills in addition to the traditional lecture offering. In response to this and other factors, the Department of Mechanical Engineering at the Colorado School of Mines (CSM) decided to reach out to students earlier in their academic career and offer them an introductory course in mechanical engineering at the sophomore level. This new course replaced a legacy course at CSM that would not always necessarily offer a design project that was of interest to a mechanical engineering major. All of the engineering students basically attend the same general courses freshman and sophomore years. As such, the Department does not receive mechanical engineering students until they are juniors. The primary reasoning behind an earlier mechanical-based course stemmed from an effort to better coordinate courses across the Department for these students and to teach material that would be utilized in later courses within the Department. These efforts resulted in the creation of the new course that was just launched in the Fall 2014 semester. Our second offering is currently ongoing this Spring 2015 semester.

Due to the fact that there was not a previous course or any prior information to draw from in creating this course, a committee was assembled that had varying experiences in design, hardware and software. Although there are other departments with classes based on hardware and software[1-3], we wanted to design our course specific to the direct needs of our Department. The overall goal of the committee was to create a new course for the students to really have fun participating in through hands-on learning and to try as much as possible to stay away from the traditional model of class lecture, homework and exams. Obviously the traditional method is still successfully utilized in upper division courses, but the goal was to depart from the norm and really focus on sparking the interest back in the students after their first year of general education courses. The course was a part of a new bulletin for their major and was also made a requirement for all sophomore-level students majoring in mechanical engineering. The Department of Mechanical Engineering is the largest department on campus and the numbers for the course have been approximately 150 students across 3 sections. In addition, we have seen a slight increase in enrollment this semester due to juniors and seniors taking the course as well. The increase in enrollment this semester was due to positive word-of-mouth communication to the juniors and seniors in and outside of the Department. Although we appreciated the increased attendance, it has proven to be problematic due to limited resources in capital costs for the course and staffing to teach the additional sections. In addition, sophomores are required to have this course completed prior to enrolling in a future course in the Department. Junior and senior-level students have first priority when registering and have pushed the sophomores to the waiting list for the course. Efforts are currently being made to possibly add a course-fee to cover the hardware costs and to limit the course to those that are within the Department and need it for follow-on courses.

Course Content

The course coverage is split between learning software, learning hardware and then finally using the software to control the hardware and associated sensors. The course content is divided into the following sections:

Introduction into Mechanical Engineering

- What do engineers do and what skills are expected of engineers?
- Discussion of engineering practice (design and analysis)
- Basics behind materials, machines, building and moving parts
- Tours of engineering building, machine shop and design lab
- Faculty and graduate students presentations about design/research throughout the semester

Introduction to problem solving with Matlab

- Programming principles and introduction to pseudo code
- Applied examples and mini projects
- Data analysis
- Graphical representation

Mechanical Design Projects

- Students will learn the fundamentals behind mechanical engineering and design
- Introduction to experimentation and data collection with a microcontroller
- Design, analysis, testing and results
- Understanding the importance of the design of an experiment
- Understanding of various types of sensors and how they work
- Team work, presentation of results and technical writing
- Integration of probability and statistical analysis into design projects through data analysis
- Problem solving with Matlab (solving a system of equations)

The general outcome for the course is to better prepare the students for their follow-on courses in the Department. As this is a new course, we do not yet have the data set to show that we have met this requirement in their advanced courses that will be taken their junior year. The specific outcomes for the course that we list on the syllabus are listed below.

1) Provide an introduction of the breadth of mechanical engineering disciplines
2) Develop a practical knowledge and appreciation of Matlab for engineering design and analysis problems
3) Develop skills for addressing open-ended design problems
4) Understand the notion of multiple solutions (parametric studies)
5) Understand how to solve multi-variant problems
6) Learn how to use programming to communicate and control mechanical hardware as a working system

7) Experience working individually and in teams and communicating technical information

It is intended for the students to use what they learn in this course and carry it throughout the remainder of their program. Several future courses in the students' major currently require (or want to require) the use of Matlab and it is the working assumption that the students have already learned what they need to know in previous classes. This fact is not always the case and this course is an effort to provide a consistent baseline in software and hardware for these future courses. Prior to the creation of this course, students typically had taken a programming course, e.g., C++ and a hardware course, both taught outside of the Mechanical Engineering Department. It is not necessarily the content of these courses that causes the shortfall, as much as it is the timing of when they are taken and the direct application to specific mechanical problems. In creating this course, a lot of collaboration was also done with the CSM's Mathematics Department to adapt their numerical methods course in order to carry on from the introduction course. The numerical methods course also utilizes Matlab.

The content of the new course differs from a traditional Matlab offering in that the students are started off with basic interactive lectures and examples on the primary functions of the code, but then the course shifts to the incorporation of the hardware. A toolbox was setup for each student group that included a circuit board, breadboard, analog sensors, resistors, LEDs, a DC motor and some basics tools. SparkFun RedBoards were chosen as the circuit board for the class due to their low cost, reliability and location in Boulder, Colorado. A multitude of inexpensive analog sensors were also chosen including a bend sensor, stretch sensor, and light sensor to name a few. Each student group consisting of 3 to 4 students was responsible for bringing the toolboxes to every lecture. In addition, the students brought their laptops to class. The bulk of the course revolved around three different projects. The instructors were very hands-off for the projects; thereby allowing the students to make mistakes and learn from these mistakes. Basic instruction was given to the students regarding how each of the sensors worked and the minimal amount of Matlab code to get them running. The students were responsible for learning how to control the hardware with the various given Matlab snippets of code. One of the first tasks of the students was to use the software and hardware together to get a LED to flash, see Figure 1 on the following page. Keep in mind, that the student had no prior knowledge of Matlab or circuits prior to this course.

For the first month of the course, we limited the content only related to the software portion. We do not teach this course in a computer lab and students are required to bring their laptops or portable devices to class. The primary reason for doing this is so that the students will have and hopefully utilize Matlab for classes whenever it is needed. In developing this course, we were unsure if it was financially feasible to require the students to have laptops for the course. We strongly suggested that student bring either a laptop or a portable device to class and for the first offering of the course, this turned out to not be a problem. In the current offering of the course this semester, we have had one student that did not own a laptop or a portable device. For the first month of the course, this particular student was able to borrow a loaner laptop from the Department during class. In future offerings of the course, we are working with the administration in order to require students to have laptops and/or a portable device and to have their electronics be currently updated to the latest version of their operating system. The latter requirement stems from the issue we have been witnessing this semester with students not keeping their laptops current and having compatibility issues with the version of Matlab we require for the course and the code provided in class for the sensors.

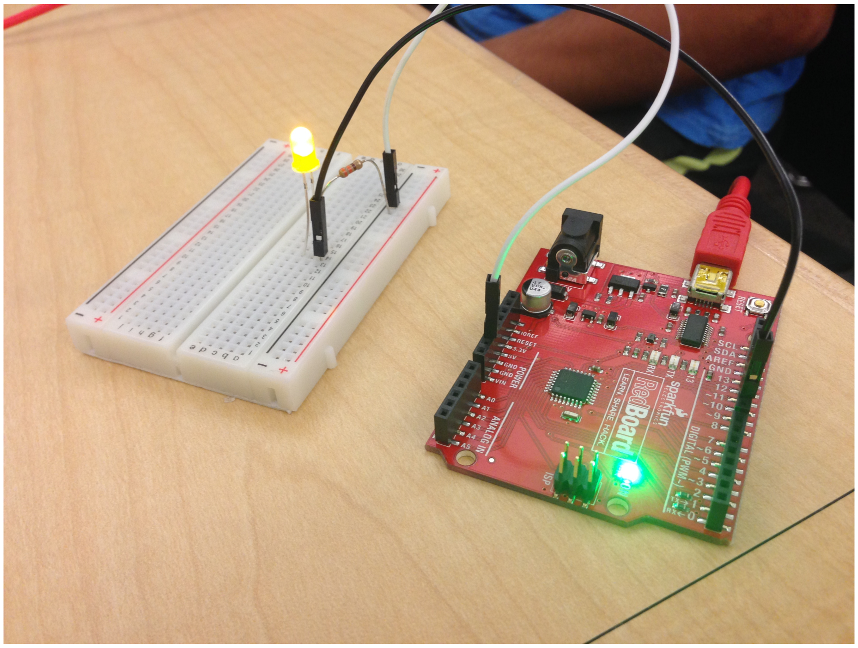

Figure 1. Student demonstration of working LED.

Three homework assignments were given to the students to help facilitate their learning of Matlab during the first part of the course. In addition to the homework, quizzes were also given. During the first offering of the course we only assigned one homework assignment related to the software portion of the course and the students provided us feedback that more assignments were needed to boost their confidence prior to the first project. The first homework assignment was just to have the student purchase the software and get it running on their laptop. Although this assignment may sound trivial, it was not. For both semesters that the course has been offered, we have realized that there are a large number of students that are not experienced in installing technical software on their machines. Issues involved with this have included virus-protection software, administrative permissions and current operating system incompatibilities to name a few.

Group Projects

The first project for the students in the class consisted of a multi-task solar panel business model that was modified from an educational project provided by Mathworks.com. The students had a large number of programming tasks that covered everything from financial calculations to various input/output user operations. Our Department conducts research involving solar panels and we thought this topic would be relevant to what they may work with in future classes or see on campus. Our primary goal with the first project was to try and balance enough exposure to different programming constructs with also not becoming too overwhelming to the students with their first attempt at programming. After completion of the first project, we received positive feedback from the students that we had met our goal.

The second and third projects of the course involved using the software to control the hardware. We started off the second portion of course with a hands-on demo of the hardware being utilized to calculate the time it takes a cylinder to roll down an inclined plane. We had three types of sensors hooked up at the top and bottom of the incline. These sensors included a force sensor, a light sensor and a flex sensor. We provided the students a computer with the Matlab code required to run the sensors, but did not tell the students which sensor was actually connected. We divided the students up into groups and told them to 1) determine which sensor was connected, 2) determine which sensor was the best for the demo and 3) determine the time it took the cylinder to roll down the incline. The first semester we taught the course, the faculty were more involved with the demo and the students seemed to be less inclined to get involved. This semester, we decided to give a brief introduction and let the students control the demo. Letting the students be more hands-on worked very well and the students were better engaged. We are going to build more demos in the next offering of the course so that we can keep the group sizes smaller.

After the completion of the demo, the students were assigned the toolboxes that contained all of the hardware they would be using for the course. The primary components of the boxes included the circuit board, wiring, sensors and basic tools. A hardware-based homework assignment was given that required the students to research what was in their toolbox, what its primary function was and where it could be purchased. We provided the students the basic Matlab code that was required to connect and initialize each sensor, but the students had to determine any code required beyond that, e.g., code for sensor calibration. The next two projects required the students to connect the software with the hardware. Without going into the exact details of the projects, the second project required the students to hookup sensors to a helmet to provide head position feedback for a Paralympic cyclist during a race (see Figure 2) and the last project required the students to build a working weather station.

In order to prevent one student from dominating the projects, it was made clear early on to the students that we would be randomly asking any student in each group to show us that a particular sensor was working, or anything about how a particular sensor, or snippet of code worked. As is sometimes the case with groups, the student that is most comfortable with the subject matter tends to dominate the working of the group. We feel that it is very important for all students to take responsibility for learning both the software and hardware, not just the piece they are more comfortable with as a result of their possible prior history in dealing with a particular piece.

In addition to learning the software, hardware and the interaction between the two, the students were also required to present their projects in class to the other students. In the first offering of the course, we only required the students to present the final project. We currently are requiring the students to present two of their projects this semester. We added another presentation to their schedule in order to give them more practice in presenting technical work. We witnessed last semester that not all sophomores had had much experience in presenting and needed some extra practice during class.

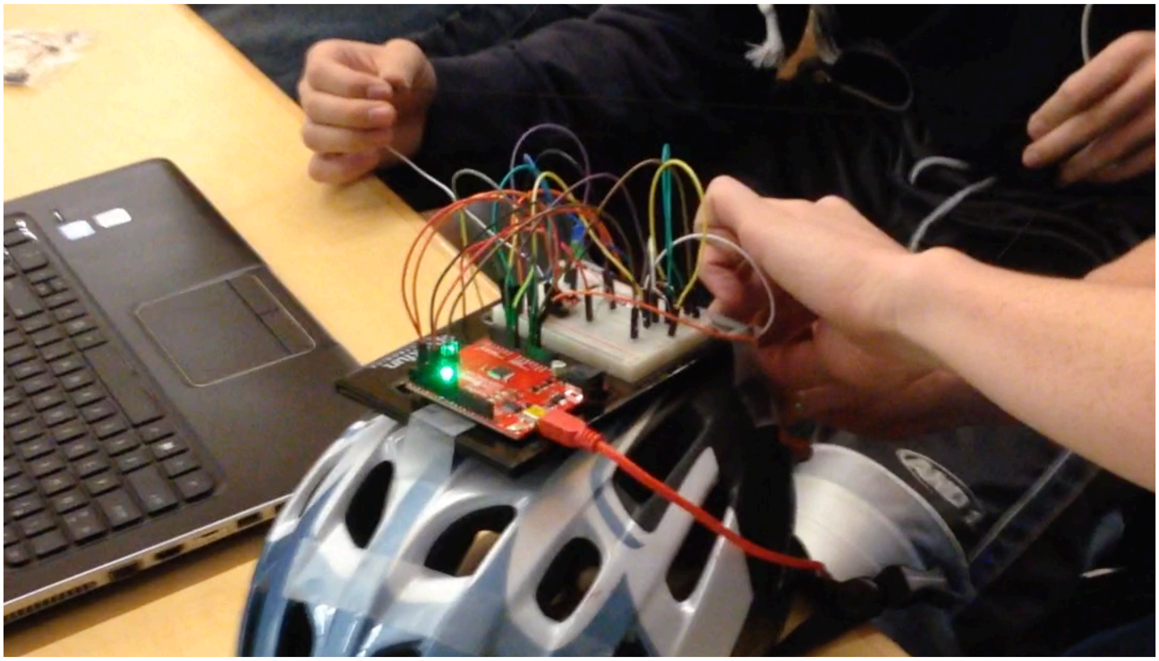

Figure 2. Multiple sensors controlled by Matlab in order to provide rider feedback on head position during racing.

Conclusions

This paper presents an overview of a new course that was developed for sophomore level mechanical engineering students. At the end of the course last semester, we asked the students to provide their feedback on the first offering of the course. The ability of the course to meet our general goal of better preparing them for their advanced mechanical engineering course will not be fully seen until the current wave of students take the future courses that will utilize the skills learned in this course. The evaluations of the course taught in Fall 2014 were very positive with most indicating that they felt like they understood both the software and hardware and their interactions together. Surprisingly there was a request for more homework in the course (discussed early in this paper) in order to better understand Matlab early on, prior to starting the projects. Obviously that request is not voiced by students very often and was new to the faculty that delivered the course the first time. The second offering of the course is still ongoing and we will have more data in the future once our current students complete the courses in their junior year. We are planning to give the same students that took our course this year an exit survey their junior year to determine how the introduction course helped them in their follow-on courses.

1) Recktenwald, G.W. and Hall, D.E. "Using Arduino as a Platform for Programming, Design and Measurement in a Freshman Engineering Course." AC 2011-2642. ASEE 2011.

2) Riofrio, J.A. and Northrup, S.G. "Teaching Undergraduate Introductory Course to Mechatronics in the Me- chanical Engineering Curriculum Using Arduino." Paper ID #7659. ASEE 2013.

3) Smaili A. “A Model for Integrating Mechatronics into Mechanical Engineering Education.” 2002-698 ASEE 2002.