AC 2012-4011: PROGRAMMING IS INVISIBLE OR IS IT? HOW TO BRING A FIRST-YEAR PROGRAMMING COURSE TO LIFE

Dr. Beverly K. Jaeger, Northeastern University

Beverly Jaeger, Susan Freeman, and Richard Whalen are members of Northeastern University’s Gateway Team, a group of teaching faculty devoted to the developing and enhancing the First-year Engineering program at Northeastern University (NU). They also each maintain a close affiliation with the Mechanical and Industrial Engineering program at NU, bringing expertise from their majors to the first-year classroom. The focus of this team is to provide a consistent, comprehensive, and constructive educational experience that endorses the student-centered, professional, and practice-oriented mission of Northeastern University. Each of the authors has won multiple engineering teaching awards.

Dr. Susan F. Freeman, Northeastern University
Dr. Richard Whalen, Northeastern University

©American Society for Engineering Education, 2012
Programming is Invisible – or Is It?
How to Bring a First-year Programming Course to Life

Abstract
Our engineering students have often asked why they need to learn computer programming and at times they miss the association that many aspects of our daily existence are dependent upon software running hardware. It is not enough to tell students that some of the required attributes of becoming a good engineer involve being proficient in the application of algorithmic thinking and problem solving. Therefore, our program at Northeastern University decided to implement a hands-on, low-cost, easily integrated component in the first-year programming course to emphasize the importance of understanding how software and hardware are interlaced, and to engage the students meaningfully in their first college programming class. The hardware-application approach that was initiated at Northeastern University contrasts some of the more traditional methods used to teach algorithmic thinking and problem-solving skills to first-year engineering students, yet follows an emerging trend. Computing projects that are used to control physical hardware were added to the Engineering Problem Solving and Computation course in order to make strong connections to the many embedded computing applications used in students’ everyday experience and in society in general. Moreover, witnessing computer instructions produce light, sound, and motion engages the students’ senses and provides the sort of immediate feedback essential for constructive and memorable learning.

In order to accomplish the goal of introducing hands-on laboratory experiences to over 600 students, Northeastern University teamed with the not-for-profit company Machine Science Incorporated to conduct a pilot study using a custom kit of electronic components in the Spring semester of 2010. The kits were comprised of inexpensive, highly functional – and in many cases reusable – components combined for the purposes of demonstrating the capabilities of programming to generate tangible and observable results from programs created by the students. This was supplemented by online tutorials, instructions, and projects provided by Machine Science Inc. to guide the learning process. The success and lessons of the pilot study resulted in a full rollout to all first-year engineering students enrolled in the programming course in the following spring.

To assess this initiative, students were surveyed prior to implementation of the full iteration of the ‘machine science’ module to provide a baseline of their impressions of what was to come. Later, they also provided feedback on several aspects of the module including the learning outcomes related to this part of the course. Results strongly showed that student participants find that these labs demonstrate the capabilities and value of programming for any engineering major. They reported that it was engaging and that they learned more than just programming, and the course and its material gained credibility with the majority of the student population – an improvement over previous years’ feedback.

Introduction and Background
At Northeastern University, the first-year engineering curriculum is common for all majors and the general engineering courses typically have 20 to 25 separate sections of approximately 30 students each. The College of Engineering requires an Engineering Design course during the entering semester in which learning principles of engineering and design is accomplished through “hands-on” tasks for students in areas such as problem formulation, creativity stimulation, construction work, and associated reporting in relation to projects that students produce in teams. There is a strong emphasis on applying technical
knowledge in a practical way, developing analytical problem-solving and decision-making skills, and demonstrating resourcefulness. As a result, students tend to be more vested in the learning process, appreciate what they have learned, remain engaged, and retain more of the material. In addition to student benefits, it has been shown that infusing active and problem-based learning modules into a course results in statistically significant increases in student ratings of instructor effectiveness and overall course satisfaction.

In the second semester of the first year, a course titled ‘Engineering Problem Solving and Computation’ centers on the practicality and applicability of logical solutions to real-life problems using software tools such as Mathworks’ MATLAB and the C++ programming language. In terms of pertinence and engagement, this particular course presents educational challenges not seen in the Design course.

**Challenge #1: Connectivity and Relevance.** This second semester ‘programming’ course had not fully made the connection between software written to solve a practical problem and how it might be used to drive hardware/devices in a visible experiential way. As a result, students were skeptical, expressing a disconnect with real-world and career applications. This weak cause-and-effect association at times resulted in a somewhat uninterested learning population. It became apparent that students did not deeply understand the importance of writing code in relation to engineering problem solving. We as instructors saw an opportunity to take a role in bridging this gap.

**Challenge #2: Resources.** A further challenge relates to resources: How can we demonstrate the value of programming and problem solving in an engaging and cost-effective way? It is clear that other programs have invested a prohibitive amount of capital. Space is another limited resource. Without lab benches and storage, we would need to be creative in developing a hands-on equipment-based module. The final resource to be considered is time. This precious commodity would also make it challenging to start from square one to develop a curriculum to teach hardware-driven program development. These conditions inspired a search to determine how to make programming come alive for our students given the limitations described.

**Challenge #3: Experience.** While there is a core of instructors responsible for the majority of the 20+ course sections from year to year, there are also some who visit the team to cover sections each term. Across both of these teaching sets, there exists a vast array of background expertise. How do we overcome any deficiencies in the instructors’ collective knowledge base in software/hardware integration? Not everyone has experience in electronics, circuits, or embedded platforms.

**Some of the Other hands-on Computing Initiatives**

The introduction of hands-on programming experiences to a first-year problem-solving course is not a novel idea. In fact at Northeastern University, the College of Engineering has offered a special section of our first-year course called High Tech Tools and Toys since the early part of the last decade. In this specialized lab setting, undergraduate students are given the opportunity to work with advanced engineering instrumentation such as digital oscilloscopes, ultrasound imaging devices, stepper motors, digital cameras, signal conditioning and A/D converters. The initial thrust of this course was to expose students to problem solving with software and hardware integration early in their education. The dedicated facility for this course can only accommodate 24 students at a time and requires full-time lab technicians to maintain and set up the weekly experiments. While the advanced electronics are inspiring for the students, the expense of the equipment and customized lab space make it impossible to expand to a population of over 600 students. Admission to this special section is set up by having students self-select
this course as an option and then are enrolled on a first-come-first-served basis. The students selected for these sections generally have strong academic standing and are told that much of the learning about programming is done independently; it is less guided instruction than the general sections. A certain number of seats are reserved for students holding a particular scholarship at Northeastern University and on average less than 15% of the 600 or so first-year students will be able to take this course.

Other universities seeking inexpensive tools for project-based programming instruction have turned to popular robot toys such as LEGO® Mindstorms and Parallax®. In its ‘Living with the Lab’ initiative, Louisiana Tech’s engineering students complete a project-based curriculum based on the Parallax® BOE Bot. Here students build robots to move around a track or follow a light, and use a microprocessor to control servos and read data, allowing students to quickly develop skills in programming and even electrical circuit prototyping. Student engineers purchase the kits in lieu of a textbook and the component inventories for the projects are managed by faculty members. A newer version of the ‘Living with the Lab’ program was implemented at Portland State using ‘Arduino’ which is a compact, inexpensive, open-source electronics prototyping platform built around an Atmel AVR microcontroller. In addition, Portland State just recently redesigned a first-year electronics course to include a hands-on element using items such as a DC motor and a motor controller chip which is used to accomplish tasks such as powering a scale-model car or turning the drawing knobs on an “Etch A Sketch” toy.

Educators at the Milwaukee School of Engineering have organized their first-year programming course for mechanical engineers around mechatronic challenges, which provide students with immediate concrete visual feedback and demonstrate the importance of programming in mechanical engineering work. Similarly, the Center for Engineering Education Outreach at Tufts University has pioneered the use of LEGO® Mindstorms robots to teach undergraduate engineering and computer programming.

At Penn State Abington College, traditional lectures in their introductory programming courses have been largely replaced with team projects to collaboratively design and implement algorithms in the C language for autonomous mobile robots. The team projects have been developed to “teach” critical programming concepts. Years back, others integrated Labview®, Lego® Bricks, robotics, and data acquisition into their first-year courses. Finally, at Drexel University, first-year engineering students are introduced to several microcontroller-based learning modules, one such being the design of a smart house in which students control a thermoelectric cooler, a heater, and an attic fan.

Development of the Machine Science Program

At Northeastern University the first-year engineering entering class has grown from about 300 students at the beginning of the last decade to over 600 in this past year. In the second semester, the college strives to offer its students more than a theoretical education, as they have in the first semester Engineering design course. In addition, Northeastern University is distinguished by its cooperative education program and is focused on providing a practice-oriented education. This second General Engineering course titled ‘Engineering Problem Solving and Computation’ is centered on the practicality and applicability of logical solutions to problems in real life using available software tools such as Mathwork’s MATLAB or the C++ programming language. It is this notion of functional knowledge and its communication that the student engineers acquire as one of their most valuable competencies to take with them into their professional working lives. Also, like many universities, the students expected to develop and demonstrate advanced proficiency in both verbal and written presentation, and the programming course is no exception.
The second-semester programming course has undergone many transformations over the years. A major change to the course was the elimination of the final exam in favor of a substantial integrated project. After developing, implementing, and assessing the integrated project, it is not only a better alternative, but it is currently the most representative final assessment method for our first-year programming courses. Also of significance, this integrated project focused on a real-world programming application, tied to the major of the students where possible. With the structure of the course using two complementary programs, students also learn to solve problems across software platforms, demonstrate how two programming applications communicate with each other, and learn how to select the "best tool" for the job within each project according to the features of each software language.

As noted, what was still lacking in the course was a connection between the software developed to solve a practical problem and how it might be used to run hardware. Once the decision was made to implement another change to address this insufficiency, we were faced with the primary challenges described earlier: connectivity, relevance, resources, and experience. In order to accomplish the goal of introducing a low-cost minimally invasive hands-on laboratory to over 600 students Northeastern University became affiliated with the not-for-profit company Machine Science, Incorporated. Machine Science, with many years of experience working at all levels, elementary schools through large universities, was a great fit.

The contents of the kit created with Machine Science are shown in Figure 1. It includes a solderless breadboard, an ATmega168 microcontroller, an LCD text display, button switches, LEDs, a piezo-speaker, a liquid crystal text display, a light sensor, a temperature sensor, resistors, capacitors, and various wires and connectors. The toolkit makes it easy to prototype circuits, since the components are modular and connect without soldering using the breadboard. A complete set of components costs under $100 which is less than the cost of a textbook and in some instances half the price of other robot kits. The components shown in Figure 1 can all be packed into a single 11" x 6 5/8" x 2 3/4" Sterilite® Small Clip Box for easy storage. This compact form is critical, since programming courses are typically taught in classrooms with computers and monitors and as in our case, very limited table space. In addition, the ATmega chip is compatible with both MATLAB and the C programming language. This provides instructors the flexibility to use it with either or both software applications.

Figure 1. Breadboard Components, which are packed in a clip-closed Sterlite® container.
Advantages of Partnership

In partnering with Machine Science, Northeastern University was able to use the existing web-based tutorials created to introduce the various kit components. This provided a dual advantage. First, having all of the material online streamlined the orientation for instructors who never have used any of these components; they were able to self-learn and experience the kits just as a student would. Second, since all of the material was on the internet, instructors could use the online lessons to introduce the kits in class, saving many hours of instructional development and allowing them to focus on project development and creative learning. In addition, the engineers at Machine Science are accessible, and have provided personal help, training instructors and attending classes, supplying parts and answering questions in a timely manner. They have helped develop new exercises and challenges when asked, serving not only as technical support, but as colleagues who assist in improving the course and the learning experience for the students.

Figure 2. Web page illustrating circuit setup on Machine Science breadboard.

Figure 2 shows an example of a tutorial web page. In general the pages include a schematic wiring diagram, photos of components, placement schematics for the breadboard, and Flash videos illustrating key steps. The tutorials also can be used as prelab assignments. This allows students to prepare on their own for more advanced work in class. Most importantly, each tutorial is developed with a real-world application in mind, making a connection with the components so that the students see their programs doing something which in many instances they themselves have seen or used.

Figure 3. Accessing the programming window using Firefox web browser.
Machine Science has also developed a compiler and file storage system that is accessible from any Java-enabled web environment as shown in Figure 3. The environment allows students to write C code, compile their code, and download it directly to the microcontrollers. Because files are stored on line, students have access from anywhere the internet is available. The only locally installed software is the Java download and driver for the USB interface. This greatly simplifies the system setup and removes the need for embedded drivers, a local compiler, and configuration of each machine. Therefore, it can work seamlessly with most university computer systems. Since the kits are portable, the students can take them home to work on their own computers –try doing that with a lab setup with an oscilloscope or other high-end equipment!

Integration into the Classroom

The flexibility of the Machine Science toolkit makes integration into the classroom easy. Since the Microcontroller is programmable in both C and MATLAB, at Northeastern University instructors can choose to introduce the kits early in the course whether or not they start the course teaching C/C++ or MATLAB. In the case of the pilot study conducted in 2010, the kit was successfully presented after the introduction of the C++ programming language and students had been exposed to loops and logic tests.

Students were first taught about using an electronic programmable microcontroller in order to experience some hands-on hardware programming which is part of practically every aspect of their daily lives. For them this easily maps to devices such as cell phones, computers, modern thermostats, lighting controllers, stereos, TV’s, cars and airplanes, to name a few. They develop an appreciation that none of these devices would be able to function without these controllers. The students were then introduced to each kit component in order to gain facility with each. They used the online tutorials and “programming challenges” in configuring the board in small increments to learn about individual components’ functionality and how they are programmed.

Instructors developed two distinct approaches in introducing the students to the kits. In the immersive profile, the introduction and activities occurred in the span of about a week, some with a day or two extra. In this case, students worked solely with the machine science kits in that time frame. In the integrated profile, the machine science module was phased in over a number of weeks as additional weekly assignments along with more traditional programming problems, tied to the material being covered that week. The topics introduced to the students first in both tracks were basic electric circuits and working with schematic diagrams. Then they worked on setting up the microcontroller on the breadboard and interfacing with it through the computer.

Once the initial interface was established, students were commissioned to work on programming challenges and add various components that were utilized them in unique ways, always mapping back to a familiar device and relating to their majors. This is another byproduct of the kit flexibility –the ease with which the materials can be applied to problems from various engineering majors. Here at Northeastern University this gives the kit a significant advantage over using other platforms since we have a common first-year curriculum and must expose our students to a wide variety of engineering problems drawn from all majors. Once students had the necessary background using the kit, a small project was assigned which was an open-ended design challenge using all of the primary kit components.
Topics and Progression

Once students created their online user accounts, they were then able to start the interactive web-based tutorials. Each tutorial introduced the students to a real-world application, wiring schematics to set up the breadboard and then programming challenges to make it work. Instructors were also free to add additional programming challenges and applications as homework to accompany the tutorials to free up much needed class time. Topics, along with some additional challenges were divided into 4 blocks as set out below.

1) **Basic Introductory Topics**: Setting up the board and downloading programs to the Microcontroller.

   *Sample Instructions*: Log in to your account on the Machine Science web site and complete the ATmega Projects “Getting Started”. By the end of the period you should be able to work through:

   - [Using the Breadboard](
   - [Understanding Schematics](
   - [Building the ATmega Board](
   - [Programming the ATmega Board](

2) **Introducing Basic Kit Components**: Light Emitting Diodes (LEDs), Buttons and Piezoelectric Speakers.

   *Sample Instructions*: Log on to the Machine Science web site and complete the ATmega Introductory Projects:

   - [Controlling an LED](
   - [Using Button Switches](
   - [Controlling a Speaker](

   You should demonstrate the following with your programming board:

   a. Have your LED blink on for 2 seconds then off for 1 second
   b. Have your LED turn off when the button is held
   c. Have your LED turn on/off along with the speaker making a siren sound
   d. Make a melody of your choice using toneout( ) function

3) **Introducing Basic Kit Components**: Light Sensors, Liquid Crystal Displays and Temperature Sensors.

   - [Using a Light Sensor](
   - [Controlling the LCD](
   - [Using a Temperature Sensor](

   You should demonstrate the following with your programming board:

   a. Using the light sensor, vary speaker tone with light intensity
   b. Using the light sensor, have the LED Bright and alarm sounding when in dim light (fluorescent room lighting will not be picked up so it will think it is dark), then turn LED off in bright light using a flashlight and stop alarm from sounding
c. Have the LCD continuously scroll on each line one at a time but in opposite directions “a message of your choice”

d. Using the temperature sensor, have the LCD continuously display room temperature in degrees C then show degrees F when a button is held.

4) **Introducing the Final Machine Science Design Project:** Multiple options were set out for students to demonstrate their competency and creativity with the kits. The following is an example of what students received for the machine science project:

Select one scenario from the choices below – or propose your own for approval by [date]. Then build and program your kit to accomplish the tasks needed. You will demonstrate your resulting system in class along with turning in a proper design report with schematic wiring diagrams, program development and operating instructions.

**Sample topics:**

1. You have taken your brilliant engineering skills and decided to design your own Smart Home. Your home system needs to detect many conditions that might cause you concern. For example, if the door was left unlocked you would need an alarm. If rooms were not kept in a reasonable temperature range, damage can occur, so this needs an alarm. When you are not home, and a light goes on in a room, or a flashlight, this would trigger an alarm. Also, an intruder may step into the house and trigger an alarm. Consider anything that would make your home safer and smarter. Your project is to prototype this design with your kits and your own original programs.

2. Imagine that the Museum of Fine Arts has opened a new wing and gallery. This gallery has many famous paintings and sculptures. This engineering firm called GE 1111 Problem Solvers has been hired to design the security system for the gallery. Your goal is to detect a risk to the gallery and artwork. Your project is to prototype this design with your kits and your own original programs.

**Some guidelines to follow are:**

- You need to detect changes that signal problems. Some examples of these are light (from an intruder’s flashlight or a machine flashing a problem, an engine light on), temperature (too hot or too cold), and motion (simulated by hitting a button or some other method if you are creative). You have components in your kit that can aid in this detection. The ranges for temperature alarms are when it is greater than 85° F and less than 50° F.
- You need to detect incandescent light, not fluorescent.
- You can attempt to detect motion using the light sensor and a light source.
- After detection, you must report any conditions you have detected, with alarms and information. For this you can use noise, lights and messages. Use these in a combination that makes sense and is documented and clear.
- Consider having different alarm types for different events; such as combinations of light, sound or messages depending on the event detected.
- You will write the programs using the code from Machine Science you have already seen adding original coding as needed to accomplish all tasks.
- Review the tutorials, and electronics projects; these will be very helpful.
- Also review the schematic for your ATmega microcontroller, using many different types of ports will take planning.
Methodology and Assessment

A priori and post-evaluation of this new course component was essential in order to assess the value for the students and for improving subsequent iterations of the new course format. The surveys described below were administered to 413 students of which 88% were first-year students and 80% were male. There were 7 instructors, who had been trained on all kit components and had additional support available to help with questions and debugging both inside and outside of actual classes. After training, the instructors used their own kits and completed the tutorials themselves. Also, course evaluations on record at the university were evaluated on the following factors before and after the integration of machine science into the courses: The learning effect of course activities and overall amount learned.

Pre-Survey. At the outset of the machine science segment of the course, students were issued their machine science kits and given a brief overview of the upcoming module. They were then surveyed as to their initial thoughts concerning the related activities. See Appendix A for the Machine Science Pre-Survey. This was done to gauge the students’ receptiveness and apprehension –if any– for an approach that combined equipment that was relatively unfamiliar with procedures that were also new to them. Basic demographics were collected –academic year and gender– in addition to any background students may have with this type of activity. The machine science component was then implemented as described, immersive or integrated.

Post-Survey. Reflection, review, feedback, assessment, and iteration are familiar elements to students in the Northeastern University Engineering Program. As such, following completion of the machine science activities in each section of the course, students again completed a survey to evaluate specific components of the machine science initiative. Likert-style and open-ended questions focused on amount learned, skills acquired, pace of instruction and the learning experience, quality of support materials and tutorials, potential applications, the prospect of continuing the module in future course offerings, and suggestions to improve implementation. Appendix B contains the full questionnaire.

Results and Discussion

Pre-Survey Results. Before starting the machine science modules, 77% of the students reported that they had little or no experience with electronics, circuits or breadboards, 19% reported they had some experience and 4% had a lot of previous use. Students were also asked in the survey about how they felt about the upcoming projects. Figure 4 shows the before and after of the similar questions – the before question asked if they were excited about it, OK with it or dreading it, and the after question was if they were glad they had worked with it, OK, or wished they had done other projects. There were not a lot that were dreading it before, and even fewer were wishing for otherwise after, as shown. Overall the majority of students were glad they worked with the kits. The pre-survey comments showed that the students were really interested in how circuits worked, in learning about the hardware/software interactions and how to program the microcontroller.
**General Outcomes.** In the past, the learning outcomes had been measured in this course through weekly projects, quizzes, and the final integrated MATLAB/C++ project before the addition of the machine science module into the course. After the addition of the machine science kit use, students performed equally well on all of these measures and their integrated projects were consistently good. Instructors reported that grades were consistent with previous years’ grades, students had a better sense of the practical application of programming and they were motivated to use programming on these real world problems. In addition, as described below, the students reported learning or reinforcing several essential concepts principally related to the machine science work. By integrating the modules while still teaching the programming concepts through weekly homework assignments, classroom practice, quizzes and the final integrated programming projects, more is learned, as reported by the students.

The machine science integration raised the level of the programming course, as reported by the instructors. These instructors have all been teaching this course anywhere from 2 to 15 years, and when reporting on results and observations, a number of points were made. First, the students seem to like the course more now than in the past (or at the least they dislike it less!) It provides the instructors the opportunity to authentically demonstrate what programming can do, without hand waving, or continually saying “Programming is in everything, really, it is important, no matter what major you are…”.

Another unanticipated set of outcomes relates to fostering a community spirit. The students talk about the course more, they are carrying their kits around campus and other students are asking them what they are doing. They have more conversations and more to show –there is a sense of camaraderie. There is also a feeling of accomplishment, as they report below, that is visible in class to everyone. The instructors remarked that students rarely “high five” over a good program but are often seen getting quite excited when the lights go on, the speakers make music, and when other successes are created with the programming boards.
A further benefit of this program is its ability to connect more undergraduates to the engineering program. This initiative enabled the college to involve several undergraduates in working with the labs and their development. Interested students were able to help create and test new projects, help in classes as peer tutors, and the college was able to hire students to work on inventorying the kit components, developing MATLAB modules, and helping develop new designs and robotic projects. As we continue with this hands-on work, we anticipate using more of the experienced students as peer tutors and to help the instructors, especially as the kits are introduced.

**Teacher Rating and Course Evaluations (TRACE).** As at any university, all courses are formally evaluated by students at the end of the term. Since machine science was the primary hands-on activity introduced into the course in recent years, TRACE scores were analyzed in two aspects of the course to consider the effects of this new component: students’ responses to “The activities in the course helped me to learn.” and “I learned a lot in this course.” were statistically examined to detect any potential differences. By design both of these questions eliminate the Instructor aspect of course evaluation, focusing on potential influences of the coursework and level of new learning.

![Evaluation of Potential Learning Effects of Machine Science from Course Ratings of Record](image)

**Figure 5.** University Course Evaluations on record for learning within the course.

As seen in Figure 5 above, the mean across multiple instructors for “Class activities help me learn” was statistically higher following the introduction of the machine science kits into the course, $p<0.002$. Similarly, the reported levels of “Overall learning in the course” showed a significant improvement following Machine Science initiatives $p<0.001$. The scores were on a 1-5 scale, with 5 being the highest/most. The potential confounding effect of teaching improvement over time was in part counteracted by two facts: (1) Any statistical improvement over time due to instructors’ teaching experience before Machine Science would have obscured any detectable differences because of the
existing trend, and (2) all instructors providing data had been honored with first-year teaching awards prior to Machine Science introduction.

**Competencies Gained.** Responses from the students tell the story best. Several graphs illustrate the feedback collected after machine science projects were completed. One set of questions focused on the skills or knowledge the students felt this project had improved. Figure 6 below reports the percent of students selecting each listed competency developed through machine science.

The highest reported competency, *knowledge of electronics* is not a surprising result, but the selection of so many additional skills was more than hoped for, given that the machine science module is a limited portion of the course. In particular, *problem solving, experimental application, and thinking logically* are objectives of the course that are challenging to instill and measure. By the end of the course, students can write working programs, but the competencies listed in Figure 6 are the life skills or engineering skills we are focused on through the coursework and in our engineering program of study.

![Figure 6. Percent of students selecting competencies developed with machine science.](image)

**The Learning Experience.** As seen in Figure 7, 64% of the students reported a sense of accomplishment, despite the struggles to make things work—or perhaps as a result of the struggles they overcame. Furthermore, students’ comments on what they enjoyed further reflect the positive contributions of the machine science projects. A sampling of representative open-ended comments asking what the students enjoyed about the machine science work mention the following categories of competencies gained:
Design, real application of theory, finished product work, making a physical project, learning circuits hands-on, learned how components work together, figuring out the programs, seeing sensors work, application of hardware and software, how they made me think, tangible engineering, making songs (music), and real-world applications.

In addition to the feeling of accomplishment, students report that this experience was challenging, enjoyable, and that they learned a lot. This is apparent because the work can be difficult and at times frustrating, but they prevailed. This cluster of outcomes is exactly what we hoped for. There are statistically significant differences between the first six outcomes to the left in Figure 7 and the last four on the right. It is clear that very few selected the last 4; each is under 10%. “Too hard” is only 3%, so even with the struggle, it is achievable. Educational research on the concept of grit associates the ability to persevere and overcome obstacles as a precursor to success. The challenge of these boards provides an opportunity to develop grit. If it were easy to accomplish, then it wouldn’t be as meaningful. Work that is challenging and frustrating—yet possible—yields a sense of accomplishment and a lot of learning, a valuable lesson in life.
**Software-Hardware Association and Comprehension.** Figure 8 shows the results of the machine science initiative in demonstrating how programs control hardware. These results clearly show that almost all students recognize that they are learning and they understand how their programs control applications and devices. Some students recognize that there is more to learn, and that they have much to learn. Recognizing that you have more to learn is certainly an encouraging outcome as it has been shown to inspire self-directed learning. A consideration for the instructors, though, is that this lack of knowledge may in part be a result of the tutorials allowing for a large amount of cut and paste, and the students have given us feedback that this feature is not desirable. We are working on adjusting our methods and the tutorials to reduce this aspect of the machine science procedures.

**Figure 8.** Student self-reported understanding of hardware and software associations.

**Figure 9.** Student assessment of concepts learned.
**Conceptual Mastery.** Figure 9 reports student self-assessment of the concepts presented. This outcome gives us impetus to consider changing some elements of the module in future iterations. If we tie this to some of the open-ended comments, the students want more insight and more depth about the programs and functions. The instructors may need to preface the modules to clarify the concepts the students will be learning and map them to the skills and concepts being taught in the general class modules. Allowing the students to see more of the underlying functions and subroutines may provide more depth in their learning and mitigate some of this perceived shortfall.

As shown in Figure 10, over 75% of the students report that this is more than just nice to know; it is good theory, good practice, and they see themselves potentially using this skill set and body of knowledge, with only 7% labeling it as ‘useless’. As first-year students, they have limited knowledge of their future as engineers, so it is very encouraging that they see this as useful.

![Select the potential usefulness of these projects that you might see in your future](image)

**Figure 10.** Student reported project usefulness.

**Continued Course Integration.** As seen in Figure 11, a majority of students reported they think that classes should continue incorporate the machine science module in this course. Only 8% disagree. It was suspected that only certain majors would feel that way. Figure 12 stratifies by major, showing that Industrial and Civil engineers feel the strongest that these modules are not needed. Educational work on connecting the programming boards to each of the majors is ongoing. This helps focus our continued efforts during the first year to connect engineering students to their future majors.
In terms of students missing the relevance of the machine science work, our original hypothesis was that it would be Chemical engineers who may resist this part of the course, but that was not the case. In the future the instructors will map to these majors more clearly, for example, that Civil engineers will use programs to design and simulate systems, and that Industrial engineers will use programming to control processes, design component-based products, and also to simulation systems. As in any course, the method by which concepts are presented can often influence how the students perceive and embrace it.

Figure 11. Student opinion on future use of machine science kits and module.

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neither</th>
<th>Agree</th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>4</td>
<td>10</td>
<td>20</td>
</tr>
</tbody>
</table>

Figure 12. Classification by major of student opinion on future use of kits.
As can be expected there were students for whom machine science projects was not their favorite portion of the course. Most of the comments on how to improve this module were focused on more time teaching and in class. Many students commented that they wanted the instructors to teach more, to explain more in class, to talk more about the applications, and to teach more of the programming. Several wanted to know the commands in the many functions, and did not want so much cutting and pasting of code. Others just were not interested, and felt that this is not an electrical engineering course, and did not enjoy this module. It is important to note that many students feel the same way about programming in general, so this is not a surprising result. Given the feedback on more teaching of the material in class, we are planning more time on task and developing lessons to address this.

Advantages of the Machine Science Approach for any University

This work and the results enable the strong recommendation for any university to consider using these kits in a first-year programming course. The kits are inexpensive, less than the cost of many textbooks, and can be purchased by students or by the program. They can be reused: Northeastern University is in its third year with the kits, needing only to purchase a few replacement components, and additional kits due to increasing enrollments. In the three years, only one or two kits have been lost (plus 1 Sterlite® cover!). Taking ownership –students take reasonable care of the kits and parts, and the parts are sturdy.

The machine science projects can be used in any lab or classroom, or even outside of the classroom as they only require a computer with an internet connection. No special equipment needs to be purchased other than some simple tools that instructors can provide such as pliers, scissors and a few spare parts. The kits are self-contained and can be customized. Many additional parts can be purchased from outside vendors and any sourcing is fine for most of the components. Importantly, any instructor can learn and teach this material. In the initial pilot development study, the team of instructors consisted of an Industrial Engineer with no experience and a Mechanical Engineer with some experience in using programmable microcontrollers. Both, equally able learners, were able to teach and provide help to students with confidence on first presentation with only minimal preparation.

A key component of success has been the excellent working relationship with Machine Science. Machine Science continuously is improving their product and the tutorials. Northeastern University has made numerous suggestions and these have been incorporated wherever it was appropriate and feasible. The kits have been enhanced with new capabilities and components over the first two years. This close work has created a signature experience which can be provided to any program. Machine Science, Incorporated is a company who can supply anywhere in the USA, and they have shown outstanding support with great personal service. The program provides online tutorials that walk the students and instructors through orientation, basic lessons and examples that make implementation a smooth process.

Conclusion

Several engineering programs have added large-scale costly laboratories with robotic projects and instrumentation to accomplish hands-on programming experiences in the first-year. However, the kit-based machine science approach can integrate into nearly any programming course within the existing classroom at a very low cost. It has exhibited strong educational outcomes, hands-on experiences, and opportunities for challenges and successes. Students reported a sense of accomplishment, being more motivated to work on projects than to just write programs, learned more about electronics and real world
applications of microcontrollers and for the most part reported enjoying the class more. Furthermore the instructors reported a more fulfilling teaching and learning experience and environment. The machine science integration raised the level of the programming course by providing the instructors the opportunity to authentically demonstrate what programming can do, without all the hand waving. Machine science kits are easily customizable, expandable, and require no additional capital investment to modify existing classrooms. They are a viable option for anyone looking to enhance a first-year programming course with hands-on real time programming challenges.

**Future Work**

On the strength of the initial pilot and the first full year with the machine science module Northeastern University is already planning to develop and implement additional microcontroller-based activities for their first-year engineering students. A concerted effort is being made to develop these projects such that they will map directly to a specific major. This will become an important component of the decision making process for many of our students who are undecided, at Northeastern University upwards of 40% of incoming first-year students are selecting their engineering field this first year.

Examples of additional projects and components are being tested for students to more clearly identify a mapping to a possible major. DC motors can be coupled with servo motors, photosensors and a simple chassis to create working robots to track and perform simple tasks demonstrating mechatronics. A mechanical engineering project might include monitoring the temperature of a system using a temperature detector and controlling the system using heaters or thermoelectric coolers demonstrating simple feedback loops. Environmental engineering projects are possible by adding infrared receivers with a light source which might be used to measure Turbidity of a fluid. Civil engineering projects might use a strain gage in conjunction with a bridge circuit to measure mechanical strain on a beam. Chemical engineers might perform a salinity test using the analog inputs of the microprocessor and a resistor. Industrial engineers can use distance sensors and proximity sensors to control production processes, as any engineer would in a manufacturing setting. Some future projects being researched for development and inclusion include those above, plus programming a remote, a text messaging system, a location finder or detection of specified materials. Other high tech add-ons are available such as accelerometers, GPS, digital key pads, potentiometers, ultrasonic transducers and analog distance sensors. All of these innovations have the potential to make programming come to life in exciting ways for all engineering majors.

**References**


APPENDIX A:
Machine Science Pre-Survey
Administered with Kit Distribution

GE 1111 – Spring 2011
Machine Science Pre-Survey

Instructor: ______ Class Time: ______ Gender: ______ Year of Grad: ______

Please fill in descriptors, responses and/or thoughts below as applies to you.

1. I am ____________________________ (excited for, okay with, dreading, etc.)
   working with the kits, the next steps and labs. – use your own descriptor for this.

2. I have worked with electronics, circuits, breadboards before: ______________________
   (a lot, some, a little, never).

3. I think working with the kits and programming lights, messages and sound will be

   ____________________________________________
   ____________________________________________
   ____________________________________________

   In this module, I hope to learn ________________________________
   ____________________________________________
   ____________________________________________

   Comments and thoughts:
APPENDIX B:  
Machine Science Post-Survey  
Administered with Kit Return, Following Application Project

College of Engineering  
GE 1111 Engineering Problem Solving & Computation  
Spring 2011  
Page 1 of 2

Machine Science Projects Feedback

Provide careful responses to each item, and please provide us some feedback in the open questions!

1. Rate your understanding of how programs control and operate electronic and mechanical components after doing the tutorial based projects and final project design.

Complete understanding Good knowledge Some familiarity A small amount I have a lot to learn
1 2 3 4 5

2. The pace of the project work in class and out of class was:

Much too fast Fast, but I kept up A moderate pace Easy for me to keep up Too slow
1 2 3 4 5

3. Rate the tutorials on the Machine Science Website in terms of learning about electronic components:

Invaluable Very helpful Somewhat helpful Had to figure out things Confusing
1 2 3 4 5

4. Rate the tutorials on the Machine Science (MS) Website in terms of learning about programming.

Invaluable Very helpful Somewhat helpful Had to figure out things Confusing
1 2 3 4 5

5. I am not sure if I learned all of the concepts and skills presented:

Strongly Disagree Disagree Neither disagree/agree Agree Strongly agree
1 2 3 4 5

6. Rate how you feel about doing more advanced projects with the Machine Science kits:

Very unsure A bit shaky Ok, I can handle it Pretty confident Very confident
1 2 3 4 5

7. Select the potential usefulness of these projects that you might see in your future:

Useless Nice to know Some good theory Good practice I see myself using this
1 2 3 4 5

8. Which describes how your partner and you worked together on the project:

Equal contribution Most work by me Most work by my partner All work by me No partner
1 2 3 4 5

9. I think classes should continue to use these kits in GE 1111:

Strongly Disagree Disagree Neither disagree/agree Agree Strongly agree
1 2 3 4 5
Machine Science Post-Survey, page 2

10. Check any/all skills or knowledge the Machine Science tutorials and projects improved:
   - Programming
   - Problem solving
   - Debugging
   - Experimental application
   - Writing software
   - Electronic components and circuits
   - Reading schematics
   - Thinking logically

11. How would you describe the learning experience? (Check all that apply)
   - Enjoyable
   - Frustrating
   - A lot of work
   - Too hard
   - Learned a lot
   - Not worth it
   - Challenging
   - Boring
   - Not applicable
   - Sense of accomplishment

Please fill in the blanks with a description of your feelings:

12. Before working with MS kits I was _____________ (excited, OK with, dreading, ... ) the module.

13. After working with the kits I _______________ (am glad this was part of the class, wish we had done something else, still OK, indifferent, or other write-in ... ).

14. Given a choice on incorporating MS modules into GE 1111 would you recommend:
   - One week immersion for tutorials
   - Integrated with classes over several weeks
   - Other: _______________________
   - Please explain your choice above:

15. What have you enjoyed about the Machine Science projects?

16. What suggestions do you have to improve the experience for the future?

Demographic Information: Major _________________ Year __________ Gender: M F