

An Introductory Engineering Course for Freshman Students- From Programming to Implementation

Utayba Mohammad, D.Eng.

Department of Electrical and Computer Engineering

University of Detroit Mercy

Detroit, MI 48221

Email: mohmmut@udmercy.edu

Abstract:

Freshman engineering courses play a critical role in a student's carrier selection and have a long-term impact from a curricular perspective. This paper describes the design of a one-credit hour course that serves as an introduction to engineering at the freshman level. The course is designed with Matlab® and Arduino® at its core and with four main objectives: educate students about different engineering fields and applications, teach an engineering oriented programming language that has simple syntax and yet powerful tools, teach the concept of problem abstraction and problem solving, and finally apply the problem solving concepts to engineering problems from different fields and disciplines. This paper shows how Matlab and Arduino are integrated to achieve all the aforementioned objectives and provide a successful engineering experience for freshman engineering students. The material provided here has been developed gradually since 2011, and was offered in its current form in an introduction to engineering course at the University of Detroit Mercy in 2013.

Introduction:

The course described here is a mandatory 1 credit hour course in the first semester for Electrical, Robotics, and Mechanical Engineers at the University of Detroit Mercy. Students registered for this course have background in pre-calculus, chemistry, and in some cases physics. Only few of the students have prior programming experience and mainly with graphical programming.

Many efforts have been directed to develop introduction to engineering courses that would help student retention and are good presentation of the engineering field. Some of these methods followed a project oriented approach^{1,2,3}, in which students are presented with a problem statement and are asked to develop a solution for this problem and produce a prototype of their solution. This approach has great advantage when it comes to retention, as students get to experience the cycle of product engineering. However, this approach has a down side from a curricular point of view. Due to time restrictions, the specific problem statement, and the very basic knowledge of Freshman students the project complexity and the solution bag-of-options become very limited, which in turn might have negative impact on the students' perception of engineering as a carrier.

A second approach adopted programming as a platform for teaching introduction to engineering^{4,5}. Since programming requires having good understanding of the problem and developing an algorithm to solve that problem, it provides a very good vehicle to teach the principles of problem solving. However, using programming in such abstract way does not capture the product development concept in engineering. Engineering students are usually more interested in building a physical system rather than developing an algorithm in their Freshman year.

A third approach combines the previous two approaches by using software to control physical systems, such as robots, hence getting the best of the two worlds^{6,7,8,9,10,11}. Most of these effort were recently directed at integrating Matlab® or Labview® with Lego Mindstorms® to perform some tasks with a robotic system. With both products, software and hardware, being produced by leading companies in the field, the results are quite impressive in terms of achieving the goal of a predefined project. This approach achieves all the expected outcomes for a freshman introduction to engineering course, but it stands alone as an isolated block in the curriculum. This paper presents some modifications to this model so that the introduction to engineering course is a first step in a progressive and integrated curriculum for electrical, robotic, and mechanical engineering students. In the following sections, the overall structure of the course is described and then the individual sub component are discussed and elaborated.

Course Design:

The course is designed with the following objectives:

- Introduce the basic programming code blocks that are available in any programming language.
- Introduce an engineering programming platform that can support future mathematical and engineering topics in the curriculum.
- Provide examples of system modeling and problem solving using programming.
- Teach the basics of a general-purpose microcontrollers kit.
- Illustrate the engineering approach of problem solving using an integrated model of software development and hardware construction.

In order to achieve these objectives three topics are covered in the course: Matlab programming, basic electric circuit concepts, and microcontroller architecture. Matlab is the glue to all of these topics and hence a basic level of Matlab programming is mandatory for teaching the remaining topics.

Matlab Programming:

Matlab® is one of the main engineering software packages used in industry and academia, it offers a simple script language and a wide range of scientific and engineering libraries. The advantage of Matlab over other languages lies in being an interpreted language that need not be compiled, has automatic type detection, and offers large library of functions and visualization

tools that allow engineers to get quick answers and intuitive understanding. Furthermore, Matlab is used in many courses in electrical and mechanical engineering to perform simulation or actual implementation of related projects such as signals and systems, circuits, communication systems, image processing, control systems, mathematics, and robotic modeling^{12,13,14}. Accordingly, Matlab satisfies all the requirement of this course and is chosen to be the programming language of the course.

Due to the difficulties that many freshman student face with programming, a two phase concept reinforcement approach is adopted. In the first phase Matlab is used to teach the basic programming structures and some of Matlab's mathematical and engineering specific capability. The topics taught in this phase are adopted from textbook^{15,16,17}, and are reinforced with engineering software examples developed in this paper. These topics are organized in 2-hour units as follows:

Unit 1: provides an introduction to computer architecture. It described the internal components of a computer and draws analogies between the way a computer works and how humans process data and make decisions. The individual components of a computer (processor, Input/output devices, and memory) are described and presented to students and demonstrated on an open PC platform. In this context, the concepts of data storage, binary data representation and measuring units are introduced and discussed.

Unit 2: Matlab is presented as an Integrated Development Environment (IDE). The concepts of variables and data storage is demonstrated using Matlab's workspace, then simple calculations and data manipulation are presented through Matlab's command window. These topics are used to reinforce the discussion of the computer architecture in Unit 1. Matlab editor, debugging and error messages are then presented and good practices of editing programs are discussed. A general presentation-through-example of Matlab's capabilities are then demonstrated, including finding the roots of polynomial, solving a set of linear algebraic equations, plotting, and manipulating audio files.

Unit 3: this unit covers vector manipulation in Matlab, including mathematical operations, dot-product, cross-product, element-by-element operations, and vector concatenation. These topics are reinforced by presenting the concept of sampling. An overview of the digital signal processing field is offered and Matlab plotting is revisited as a point-to-point relation. An example of plotting a sinusoidal wave is used to explain the concept of sampling, in which the sampling period is varied and the sinusoidal wave is plotted. Furthermore the Matlab "sound" function is used to play the the waveform so that students can relate the digitally plotted sinusoid to a physical quantity that they can hear. Students are then asked to use concatenation and element-by-element multiplication to synthesize different sound bites with a proper sampling frequency.

Unit 4: extends the concept of Matlab vectors to Arrays. Students are first introduced to two dimensional arrays and then to multi-dimensional arrays. The field of image processing is used to

reinforce the multidimensional array manipulation. Student start with creating multicolor pattern images as shown in Figure 1 in preparation for the more in-depth examples in Unit 5.

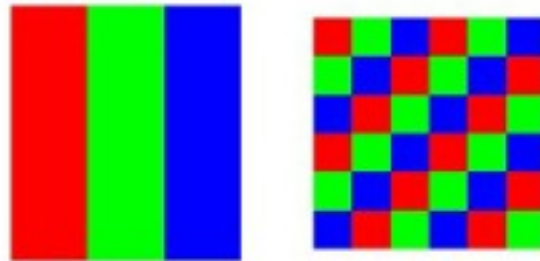


Figure 1: Multicolor Image Patterns

Unit 5: as students get comfortable with manipulating multi-dimensional matrices, the subject of Steganography is presented. The concepts of data codes and data types are presented in an abstract way. Student are then introduced to a simplified algorithm that hides a text in an image every other 'n' pixels of the different color planes. They are then challenged to retrieve that text properly. Furthermore, students are asked to improve the encryption algorithm so that the visual changes in the image are not abrupt and hardly captured with the naked eye. Figure 2 shows an example of hiding text in an image.

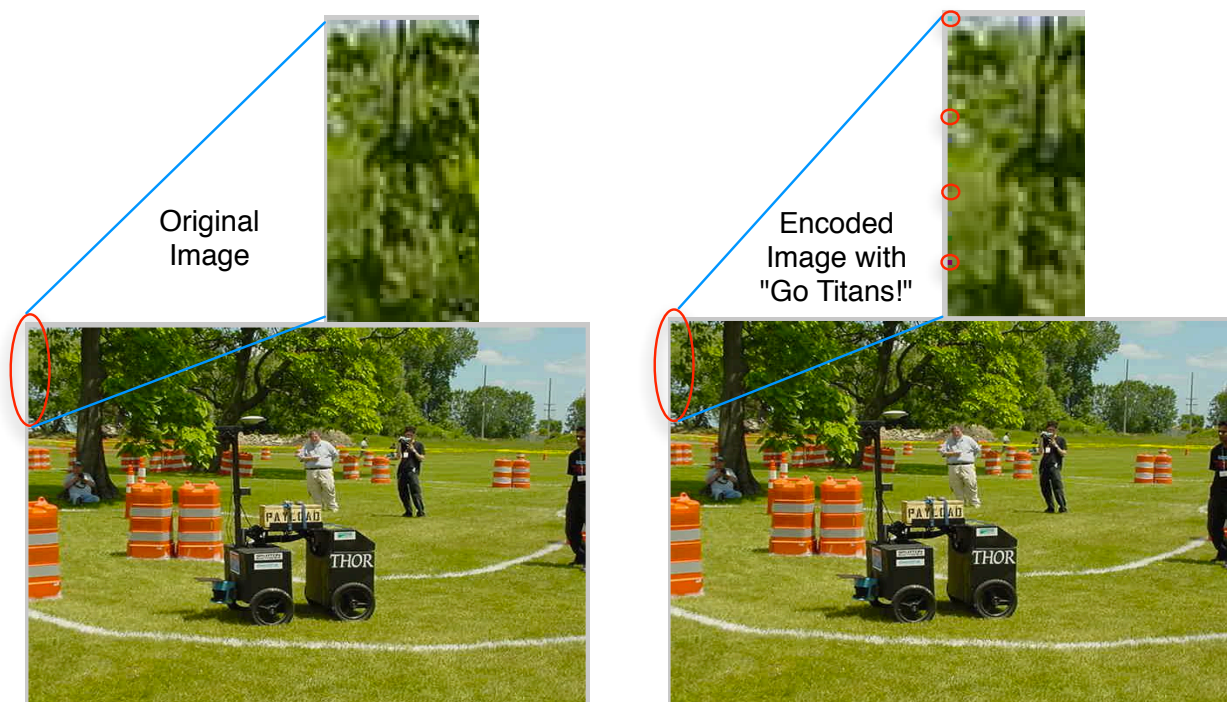
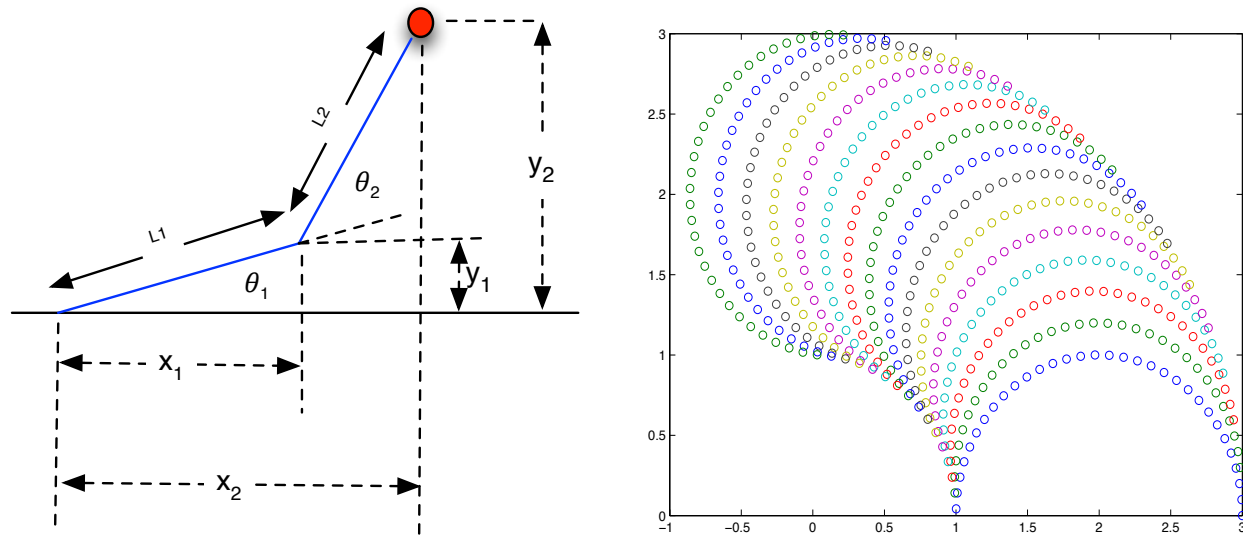


Figure 2: Sample Image With Embedded Text and Without Embedded Text

Unit 6: in this unit the concepts of functions and variable scope are presented and the procedural approach of software development is discussed. An example of a two-joint robotic arm, Figure 3, is presented. Students are asked to derive the mathematical equations that



calculate the end location of the robotic arm based on the angle movements of both joints; then write a Matlab function that implements those equations. Matlab is also used to plot the location of the robotic arm when only one joint is allowed to move, and then when both joints are allowed to move, hence producing a full representation of the arm motion space as shown in Figure 3.

Figure 3: Two-Joint Robotic Arm And Matlab Plot Of The Its Positions Based On The Different Joint Angle Combinations.

Unit 7: in this unit conditional Statements, “if”, “if-then-else”, “if-elseif-else”, and “switch” are presented. This unit uses textbook examples to explain the use of conditional statement. The robotic arm example in Unit 6 is then modified to represent an optimization problem. Each joint is assigned different time and energy costs and students are asked to write a function that takes three inputs: the current joints angle, the target arm position, and an optimization criteria; and then outputs the most energy or time efficient joint motions that will lead to the target position based on the optimization criteria. This example covers a full unit by itself, however it reviews almost all the topics from unit 1 to unit 6.

Unit 8: this is the last unit of the first phase of the class, and it teaches the loop blocks in Matlab (“for” and “while” loops). Textbook material is used in this unit to cover the different usages of loops such as, iterating over elements, calculating mathematical series, and providing repetitive action under specific conditions. The reinforcement material for this unit is described in the second phase of the course.

Basic Electric Circuits:

In order to use microcontrollers basic knowledge of electric circuits is required. Hence, Unit 9 is dedicated to provide students with general knowledge of electric circuits. It covers different topics at a very basic level, including: voltage and current, using a breadboard, using a power supply, building an LED circuit, using switches, and using potentiometers.

The Arduino Microcontroller Platform:

After unit 9 students will be ready to learn the basics of microcontrollers. Related work has been mainly focused on using the LEGO NXT Mindstorms brick^{8,9,10}. Although the NXT is very suitable for a freshman course, it does not have the potential to be used in other courses of the curriculum due to its proprietary sensor/actuator devices. Hence, this course sought a general purpose microcontroller kit that will provide a platform for future projects and courses throughout the curriculum. The Arduino® platform offers that versatility, and multiple efforts have demonstrated using Arduino in a classroom^{18,19,20} and many others have already taken place in many courses in electrical engineering, mechanical engineering, and robotics among others.

Arduino uses its own IDE with a C++ based programming language. This made it popular among hobbyists who had prior experience with C++, but not necessarily all its potential crowd. In this freshman course, the time has been invested in teaching Matlab due to its long term benefits in the curriculum, hence teaching another programming language, Arduino, would have been overwhelming to students and unrealistic. Fortunately, Matlab has recently offered a supporting library to Arduino using the serial port communication. A server program is installed on the Arduino board and a Matlab object is created in the Matlab environment. The communication is then initiated over the serial port using a USB connectivity between Matlab and the Arduino board. Matlab maintained all the function names of Arduino's software package, hence if a student desires to learn Arduino's native language, the transition from Matlab to Arduino, and vice versa, will be very easy, and the material taught in this course will have a constructive effect.

The Matlab-Arduino setup enables students to use what they learned in the first nine units and apply it to an actual hardware, hence providing a second phase of material reinforcement in the context of a microcontroller-driven hardware. Two units are used as introduction to microcontrollers; Units 10 and 11 cover the basic circuit and coding interfaces to the Arduino Mega 2560 board, and they are designed as follows:

Unit 10: this unit covers the digital circuit interface to Arduino, digital read and write, including reading switches and controlling LEDs. Multiple exercises are used in this unit to revisit the control and loop structures in Matlab including: flashing an LED for 'N' times (loops), controlling an LED with a switch (conditional statements), and creating different flashing

patterns with multiple LEDs based on multiple switch inputs (combining loops and conditional statements).

Unit 11: this unit covers the analog circuit interface to Arduino (analog read and analog write), including reading potentiometers and generating sound with a piezo. This unit extends the exercises in Unit 10 to flash the LEDs at variable speed that is controlled by a potentiometer, and then associating an alarm sound frequency with the potentiometer input as well and using that frequency to drive a piezo. Figure 4 shows the a sample of the setup in Units 10 and 11.

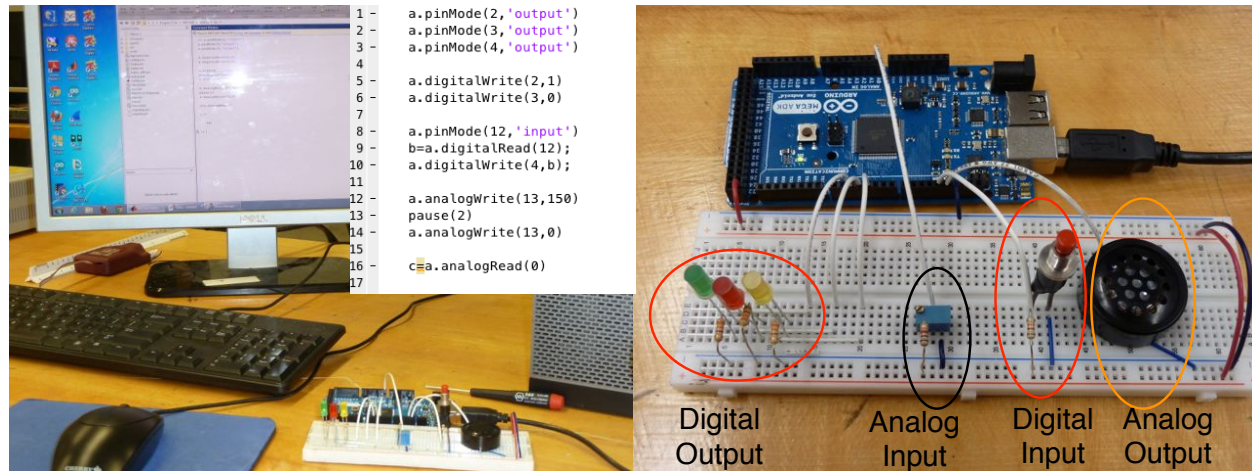


Figure 4: the overall basic setup of the Arduino and Matlab

Design Project:

After the students learn how to use the Arduino board with Matlab, a conclusive robotics projects that combines design, software, and hardware aspects is posed to the students. As this course veered away from the LEGO platforms, the Tetrix robotics ® platform from PITSCO® is adopted. Tetrix is made of heavy-duty aluminum blocks that can be attached together in many different angles, allowing for a flexible design while maintaining a sturdy chassis. Tetrix uses two powerful DC motors, 300 oz-in of torque each, as the main drive for the platform; and adds servomotors to control local part movements. The students are given the option of assembling their own robots the way they like or using a robot that is assembled off-the-shelf. Each robot is equipped with one ultrasonic sensor, two infrared proximity sensors, two DC motors and their H-Bridge drivers, a piezo, and a set of resistors, LEDs, and switches. Table 1, shows the exact part numbers, vendors, and price estimation per robot as of 2012 when these kits were purchased.

The activities in this project are divided to three Units:

Unit 12: in this unit students are asked to connect the sensors properly to the Arduino board and write functions to read the data from these sensors and render it in an understandable format. e.g. the IR and ultrasonic sensor return analog values that range from 0 to 1023, students are

asked to process the analog reading and render it in inches. Furthermore, they are asked to perform experiments on the sensors to verify the maximum sensing distance and field of view for every sensor. By the end of this unit, students decide where to place the sensors on the robot, based on their characteristics, and then run a test procedure to verify the functionality of the robot after all the sensor are mounted.

Unit 13: In this unit students are asked to write the driver functions to the robot. Starting with functions that drive one motor only, the left or right, backward or forward at a configurable speed. Those functions are then extended and used in a hierarchal fashion to develop orchestrated motions, such as spinning left and right, backing up left and right, veering, and driving straight. By the end of Unit 13, the robot is ready to drive and an intelligence algorithm is the only block remaining in the project. Figure 5 shows the robot in its preliminary and final installment stages.

Table 1: The Robot Bill of Material as of 2012.

Part	Price	Quantity	Part #	Retailer	Total
Robot Base	\$399.00	1	W39137	Pitsco	\$399.00
Encoder	\$55.00	2	W35915	Pitsco	\$110.00
Battery	\$49.95	1	W39057	Pitsco	\$49.95
Charger	\$34.95	1	W39059	Pitsco	\$34.95
Arduino Mega 2560	\$56.35	1	RB-Ard-19	Robotshop	\$56.35
DFRobot XBee Expansion Board	\$13.80	1	RB-Dfr-32	Robotshop	\$13.80
DFRobot Xbee USB adapter	\$22.08	1	RB-Dfr-55	Robotshop	\$22.08
Breadboard Mini Self-Adhesive Black	\$3.56	2	PRT-08803	sparkfun	\$7.12
XBee 1mW Wire Antenna - Series 1	\$20.66	2	WRL-08665	sparkfun	\$41.32
MegaShield Kit for Arduino	\$16.16	1	DEV-09346	sparkfun	\$16.16
Ultrasonic Range Finder - Maxbotix LV-EZ0	\$25.16	1	SEN-08502	sparkfun	\$25.16
Infrared Proximity Sensor - Sharp GP2Y0A21YK 10-80 cm	\$12.56	2	SEN-00242	sparkfun	\$25.12
Dual VNH3SP30 Motor Driver Carrier MD03A	\$44.95	1	707	pololu	\$44.95
30mm Speaker: 100 Ohm, 0.15 W	\$1.79	1	1261	pololu	\$1.79
				Total	\$847.75

The final assigned task in the course takes place in Unit 14. The students are asked to write a Matlab script that will navigate the robot safely through an obstacle environment. To do so the Matlab-Arduino USB connection had to be converted to a wireless connection. ZigBee based serial-to-ZigBee XBee transceivers are used for this purpose. One transceiver is connected to the serial port of the robot and the other is connected to the compute's USB port though an adaptor. The XBee transceivers fully mimic the serial port communication and hence no modification in the software are needed. Figure 6 shows the final setup of the robot using the XBee transceivers.

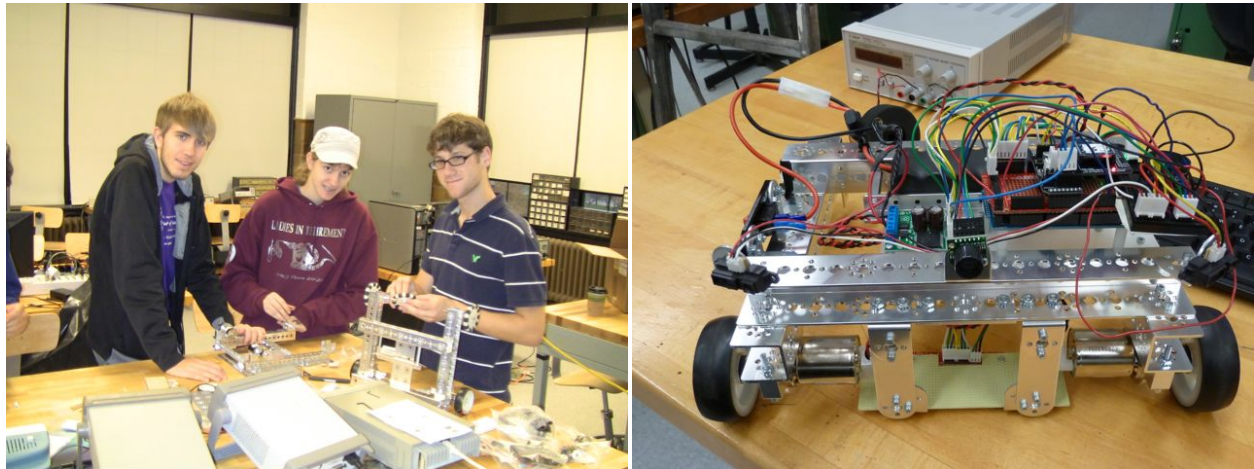


Figure 5: The robot platform in a bare chassis form and with its complete outfit of sensors.

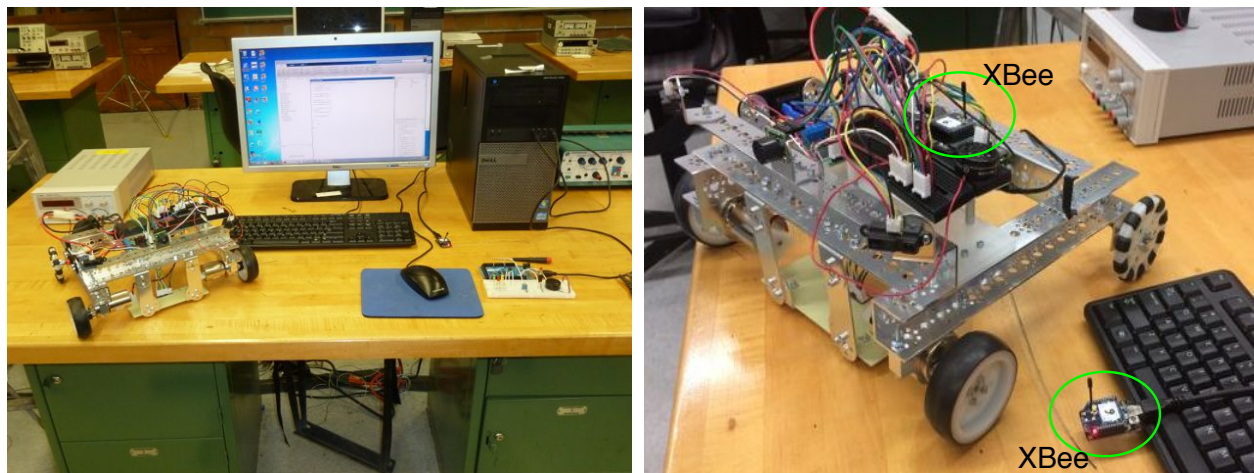


Figure 6: the robot setup with XBee transceivers.

Assessment:

The student performance in this course is assessed through weekly quizzes, homework, and in-lab activities. The homework is assigned 50% of the total grade as it is designed to be more extensive; quizzes are assigned 30% of the total grade and they are proctored at the beginning of every unit; while in-Lab activities are the exercises posed during class time and are graded based on completion. Although more statistical data is needed to confirm the effectiveness of this course, the initial feedback from student in the first offering of this course in Fall 2013, indicate the over 85% of students confirmed that they gained good understanding of the concepts and principles presented in this course and that they can apply the principles a concepts in new situations.

Future work:

In this paper a new approach of teaching for teaching introduction to engineering in the freshman year is presented. This approach uses a two-phase concept reinforcement system in which topics are reinforced as they are presented in class through direct examples and then they are revisited and reinforced at a latter stage in the context of hardware-based challenging project. The lecture material is drawn from different engineering fields, however, there is still room to incorporate more diverse examples in order to cover wider range of applications. The flipped approach is particularly suitable for this type of courses and can be integrated with the course in form of multimedia pre-class sessions.

Bibliography

1. J. Frolik and T. Keller, "Wireless Sensor Networks: An Interdisciplinary Topic for Freshman Design," Proceedings of the ASEE Annual Conference and Exposition pp. 1–7, 2005.
2. D. Ye, I. Brutman, G. Georgi, and L. Folan, "Freshman project: Autonomous underwater vehicle (AUV)," Proceedings of the ASEE Annual Conference and Exposition 2007.
3. R. E. Barr, P. S. Schmidt, T. J. Krueger, and C. Y. Twu, "An introduction to engineering through an integrated reverse engineering and design graphics project," Journal of Engineering Education, vol. 89, no. 4, pp. 413–418, 2000.
4. D. Morrell, "AC 2007-341: DESIGN OF AN INTRODUCTORY MATLAB COURSE FOR FRESHMAN ENGINEERING STUDENTS," 2007.
5. M. Feldgen and O. Clua, "Games as a motivation for freshman students learn programming," Frontiers in Education, 2004. FIE 2004. 34th Annual, 2004.
6. A. Behrens, L. Atorf, R. Schwann, B. Neumann, R. Schnitzler, J. Balle, T. Herold, A. Telle, T. G. Noll, and K. Hameyer, "MATLAB meets LEGO Mindstorms—A freshman introduction course into practical engineering," Education, IEEE Transactions on, vol. 53, no. 2, pp. 306–317, 2010.
7. S. Daniels, D. Harding, and M. Collura, "Introducing feedback control to first year engineering students using LabVIEW," pp. 1–12, Apr. 2005.
8. E. Wang, "Teaching freshmen design, creativity and programming with legos and labview," vol. 3, pp. F3G–11–15 vol. 3, 2001.
9. J. M. Gomez-de-Gabriel, A. Mandow, J. Fernandez-Lozano, and A. J. Garcia-Cerezo, "Using LEGO NXT Mobile Robots With LabVIEW for Undergraduate Courses on Mechatronics," IEEE Trans. Educ., vol. 54, no. 1, pp. 41–47.
10. P. Ranganathan, R. Schultz, and M. Mardani, "Use of LEGO NXT Mindstorms brick in engineering education," pp. 17–19, 2008.
11. E. K. Rézaei and S. R. Kolla, "Internet-based ON/OFF controller using labVIEW," Proceedings of ASEE Annual Conference 2003.
12. M. Enelund and S. Larsson, "A computational mathematics education for students of mechanical engineering," World Transactions on Engineering and Technology Education, vol. 5, no. 2, p. 329, 2006.
13. S. Ayasun and C. O. Nwankpa, "Induction Motor Tests Using MATLAB/Simulink and Their Integration Into Undergraduate Electric Machinery Courses," IEEE Trans. Educ., vol. 48, no. 1, pp. 37–46, Feb. 2005.
14. O. I. Okoro, P. Govender, and E. Chikuni, "A new User-Friendly Software for Teaching and Research in Engineering Education," The Pacific Journal of Science and Technology, vol. 7, no. 2, pp. 130–136, 2006.
15. "Introduction to MATLAB for Engineers", William Palm III, 2011, ISBN: 9780073534879.
16. "Matlab: A Practical Introduction to Programming and Problem Solving, 3rd Ed.", Stormy Attaway, 2013, ISBN: 978-0124058767
17. "Engineering Computation with MATLAB, 3rd Ed.", David M. Smith, 2013, ISBN: 9780132568708

18. M. A. Rubio, C. M. Hierro, and Á. Pablo, "USING ARDUINO TO ENHANCE COMPUTER PROGRAMMING COURSES IN SCIENCE AND ENGINEERING," presented at the Proceedings of the EDULEARN13, 2013.
19. J. Sarik and I. Kymissis, "Lab kits using the Arduino prototyping platform," presented at the Frontiers in Education Conference (FIE), 2010 IEEE, 2010.
20. P. Jamieson, "Arduino for teaching embedded systems. are computer scientists and engineering educators missing the boat?," Proc. FECS, pp. 289–294, 2010.