IMPLEMENTING SELF LEARNING SKILLS WITH MULTIDICIPLINARY ROBOTICS COURSES

Akin Tatoglu and Ingrid Russell
Mechanical Engineering and Computer Science
University of Hartford

Abstract

Current and continuous rapid advancements in technology and science require engineers to quickly adapt to and excel in new concepts and tools. Designing core of engineering education with enhanced self-learning capabilities is immense for the success of future engineers. Engineering faculty's responsibility is to train students for multidisciplinary research and development skills as well as to motivate them to learn implementing new skills to their technical knowledge base. Multi-disciplinary topics such as mechatronics and robotics are suitable for these purposes and common interest in these fields from various departments makes them a promising delivery method of such training. We present experiences and preliminary results of assessment developing and using hands-on engineering curriculum material based on the Arduino platform. The goal is to enhance student learning by engaging them in a contextualized hands-on learning experience and introducing self-exploration methods which incrementally induces programming knowledge in the context of a highly visual and easy-to-use environment.

Keywords

Multidisciplinary, self-learning, mechatronics, robotics, Arduino

Introduction

Current and continuous rapid advancements in technology and science require engineers to quickly adapt to and excel in new environments and tools. Designing the core of engineering education with enhanced self-learning capabilities is essential for the success of future engineers in the field. As engineering faculty, our responsibility is to train our students for multidisciplinary research and development skills as well as to motivate them to learn how to implement new skills to their engineering knowledge base. Multi-disciplinary topics such as mechatronics and robotics are suitable for these purposes and common interest in these fields from various departments makes them a promising delivery method of such type of training.

The Mechatronics Systems course is offered to both senior undergraduate and master mechanical engineering students. The goal of the course it to help students fuse and practice various previous engineering knowledge and topics such as calculus, linear algebra, dynamics, simulation, mechanical design under constraints, sensor fusion, system modeling and especially control systems as well as other multi-disciplinary contents such as scientific programming skills. Mechanical engineering students tend to shy away from computer programming.

The course starts with principles of transducers and sensors and how to interface them with a process in a computer environment. Discussions about types of transducers and different sensors include operating principles, modeling, design considerations, and applications. Computer interfacing work includes signal conversation, interface components, and real time application of microcomputer systems to problems in manufacturing. Component integration and design projects involve problems from industry that require computer interfacing and computer and experimental techniques. Other topics include principles of transducers and sensors, signal processing, and data acquisition, and computer interfacing using case studies.

Software has permeated all aspects of our lives and has significantly transformed the way we live and work. It is therefore critical to prepare a talented workforce capable of ensuring the country's competiveness in this global economy. Mechanical engineering students tend to shy away from learning computer programming. The computational component of the course introduces a visual programming environment, followed by mobile systems development studying specifically autonomous mobile robotics topics. For this goal, low cost Arduino boards are used.

Mechatronics and robotics systems research topics such as manufacturing technologies[1], autonomous mobile robotics[2] as well as bio-inspired mechanical systems[3] are actively studied by academia and industry. While more complicated topics are studied during graduate education, a simplified model of systems could be delivered by utilizing easy-to-use control boards to undergraduate students. While previous control boards had multiple components and were more sophisticated[4], current single board systems make it easier for students to rapidly develop prototypes and study underlying engineering concepts[5-6].

Physical computing is becoming popular in the classroom and in extra-curricular activities. The authors[7-8] examined the potentials of physical computing activities for computer science education. Physical computing has been used to teach computational thinking, programming, and design skills[9]. In this study, the focus was on a graduate course for non-majors. The Arduino platform has been used in the engineering and computing curriculum. The authors[10] presented experiences using the Arduino platform in an introductory course on digital design using the Arduino LilyPad microcontroller and experiences using the Arduino platform in advanced courses in Automated System Design and Implementation using the Arduino Nano microcontroller.

In this paper, we will explore the use of the Arduino platform in the Mechatronics course with the goal of helping students fuse various previously learned engineering concepts and topics, and at the same time improve on their scientific programming skills. The Arduino microcontroller system is a simple microcontroller board with an easy to use programming environment. It is an open source electronics platform based on easy to use hardware and software that can be used for the creation of interactive objects or environments. Different board types come with options including a variable number of analog, digital and Pulse Width Modulation (PWM) pins and with a standard serial communication (USB) port which is used to upload the code and to receive data during run time. Additional properties include a stackable Bluetooth or Wi-Fi communication board as well as higher current motor drivers with a possible external power supply connection.

In addition to advantages listed above, there are several online open source software and hardware resources for students; time required to setup the system is minimized with easy to connect pins. When compared to building a circuit from scratch, the Arduino platform avoids intense electronical engineering knowledge. In this way, mechanical engineering students could rapidly start focusing on the control structure with an easy to use programming environment, one of the goals of the course. Moreover, based on our experience, user friendly hardware and software environments encourage mechanical engineering students to study robotics and become more comfortable with coding.

Next section explains an overview of the Mechatronics course and its outline, followed by a discussion of the hardware selection methodology used. Then, using hands-on engineering curriculum material based on the Arduino platform is presented with step by step examples. As an outcome of the curriculum, a selected semester project is introduced in detail. Finally, we present experiences and preliminary results of assessment. The goal is to enhance student learning by engaging them in a contextualized hands-on learning experience and introducing them to fundamental engineering concepts in the context of a highly visual and easy to use environment.

Course Overview

Mechatronics is a design process of combining various branches of science and engineering including mechanical, electronics and computer science. It aims to find optimal control structure considering multidisciplinary aspects. The Mechatronics Systems course is mostly offered to mechanical engineering students. Since the course requires a deep control theory background, it is recommended that students take it in their junior or senior year. However, the nature of the course includes various multidisciplinary components –such as sensors and dynamical systems simulations— that its scope could be tailored for most of engineering and science majors from freshman to senior levels.

At our college, Mechatronics Systems Design course is offered to both senior undergraduate and masters of mechanical engineering students. Students are expected to have a good understanding of advanced mathematics, dynamical systems and control theory. This course emphasizes utilizing two or more engineering knowledge together.

The goal of the course it to help students fuse and practice various previous engineering knowledge and topics such as calculus, linear algebra, dynamics, simulation, mechanical design under constraints, sensor fusion, system modeling and especially control systems as well as other multi-disciplinary contents such as scientific programming skills.

Topics include principles of transducers and sensors and how to interface them with a process in a computer environment, discussions about types of transducers and different sensors include operating principles, modeling, design considerations, and applications. Computer interfacing work includes signal conversation, interface components, and real time application of microcomputer systems to problems in manufacturing. Component integration and design projects involve problems from industry that require computer interfacing and computer and experimental techniques. Other topics include principles of transducers and sensors, signal processing, and data acquisition, and computer interfacing using case studies.

Outcomes of the course include modeling and analysis of dynamics of physical systems, selecting appropriate transducers and implementing suitable interfacing methods as well as optimizing control structures. To gain more experience and be prepared for industry or academic life, students are expected to complete a set of real world experiments with actual industrial and scientific systems such as electronic damping for mass-spring-damper system. For modeling and simulation purposes various tools such as VisSim, LabView and Matlab are offered. Outcomes are listed as following:

- Understand the use of modeling, analysis, and control dynamics of physical systems
- Have the ability to select and interface sensors, actuators, and controllers for industrial applications
- Carry out experiments on actual systems involving monitoring and control. (One example involves the application of electronic damping for Mass-spring-damper system)
- Apply simulation and modeling techniques
- Carry out a real life mechatronics project

Finally, students are expected to design and build a mechatronics project of their choice. Students are challenged to implement and present cutting edge solutions. In most cases, students are challenged by being encouraged to leave their comfort zone and to use open source information for rapid prototyping. To support self-learning skills, a set of Arduino-based labs are offered which are discussed in the following sections.

Hardware Selection

Our hardware selection criteria is based on the fact that technology is advancing rapidly and every couple years new hardware is introduced to the robotics and mechatronics community. When this is the case, students – future engineers – are required to adapt new tools on the market rapidly, which requires self-learning skills. Moreover, an easy to use coding interface is a necessity since based on our experience mechanical engineering students tend to shy away from computer programming.

There are various options, including graphical languages on the market that are used for delivering mechatronics and robotics lab sessions. Our focus is given to solutions which students could practice the nuts and bolts while not having an overwhelming programming experience.

Finally, we have chosen the Arduino Uno board which is an open-source single-board microcontroller with an easy to use hardware and software components. The cost of such a system is relatively inexpensive and the software is open source. The board comes with multiple I/O options including analog, digital and PWM pins. 16 Mhz ATmega328P processor offers sufficient power for real time basic processing and smooth serial communication via USB or Bluetooth connections. Moreover, low power consumption of the processor makes it possible to drive the system with low cost, low weight batteries which are ideal for mobile robot applications.

Lab Sessions

Students are assigned four Arduino-based lab assignments and a term project. Lab assignments study the working logic of the board, connections, assigning soft variables representing physical sensors, analog and digital signals, motor control and individual component working principles. Also, students are given a list of typical term projects and asked to pick one or, with the approval of the instructor, propose a new project. Below, we will give a brief overview of the lab assignments. In the following section, we discuss the term project.

The Arduino Lab sessions start with introducing the board structure. The board has various high speed digital input and output and 10 bit analog input pins. Header connectors are standard and add-on modules/shields could be added to the stack for various purposes including Bluetooth, Ethernet connection or additional motor controllers. In all labs, the Arduino Uno is used whose interface is shown in Fig. 1 (left image).

The hardware and the integrated development environment (IDE) are then introduced. The compiler is Java based and is cross platform. The IDE has typical properties such as brace matching, automatic indentation and syntax highlighting. Each block of code is called "sketch". An illustration of the IDE is shown in Fig. 1(right image).

The programming structure has components of “setup” and “loop”. The first block is structured for pin assignments and parameter initializations. While this block is executed one time only, the second block –loop— is run continuously. The second block includes reading and writing from/to the I/O port before or after processing parameters, based on the program being implemented. The Arduino’s IDE offers a math library with generic mathematical functions which can process float and double data types.

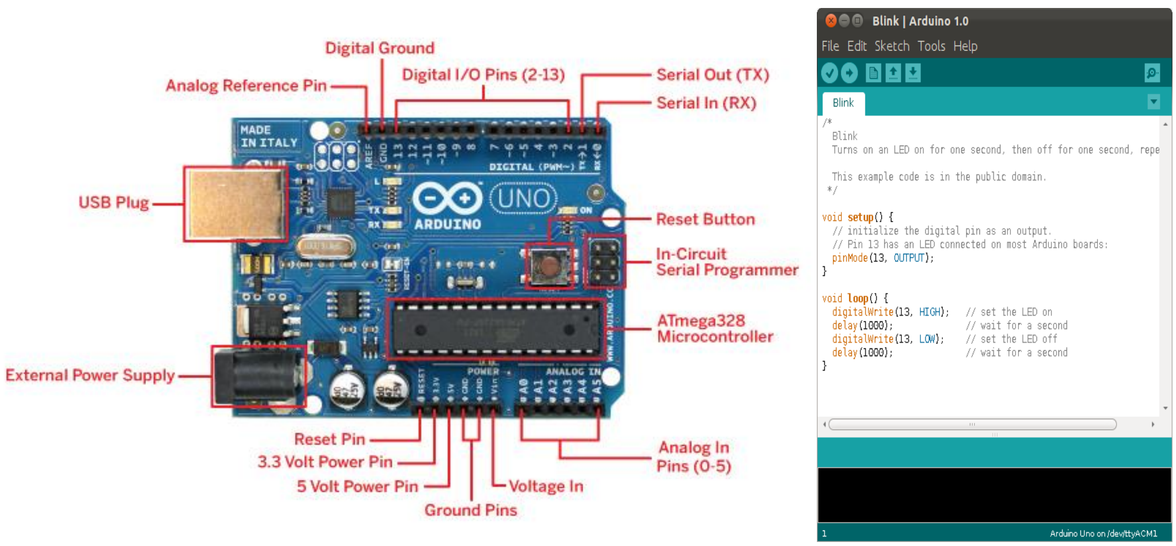

Figure 1. Arduino Uno board (left) and IDE(right) (Courtesy of Arduino)

Followed by the introduction, the logic of how to assign the value of a physical input/sensor to a variable is shown. For this purpose, initially an example of accessing the LED onboard is delivered step by step. After then, reacting a state change problem statement is explained and students are expected to complete the project themselves with minimum help. Diagram shown in Fig.2 is given to students and they are asked to build the circuit and write a sketch to control the behavior of the circuit with respect to button’s state change. If they need help, an example of similar project is shown which is included in the help files of Arduino IDE to simulate the getting help from open source community.

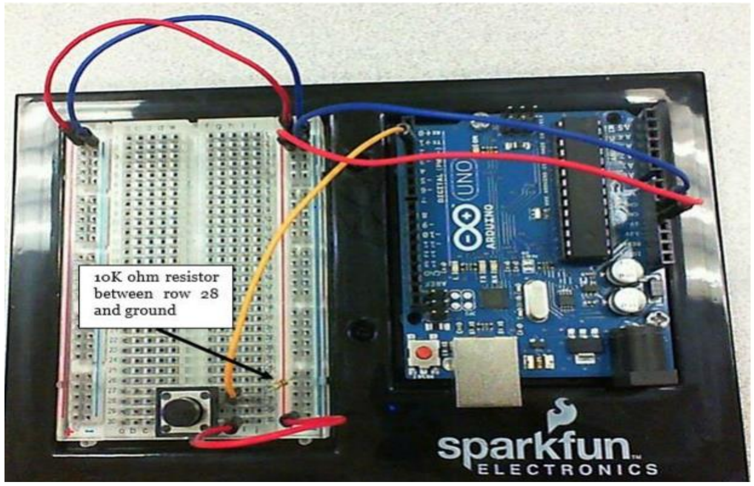

Figure 2. Example Experiment: Setting physical output with respect to state change of a button (Courtesy of Arduino)

Followed by these basic problems, students work on more advanced variable input problems which include steps for getting a variable input from a potentiometer, controlling blinking frequency of a component and monitoring it. After students feel more comfortable with controlling the frequency of a component, the pulse width modulation (PWM) is introduced. It is simply applied to an LED to monitor the PWM input changes. Then it is applied for the real purpose of controlling a motor and external circuit/shield, and power supply connection is practiced. The lab sessions includes a variety of open ended questions for practice.

Term Project

After students complete the lab assignments, they are expected to work in teams to complete a term project which includes multiple input and output as well as control methodology such as a PID controller. Typical projects implemented by students include line following, maze solving and sumo robots as well as firefighting robot and robotic arm. We present below one of the student projects, the Firefighting Robot project. The Firefighting Robot project design and development steps are explained below. Fig. 3 presents the initial mechanical CAD design and the developed final product.

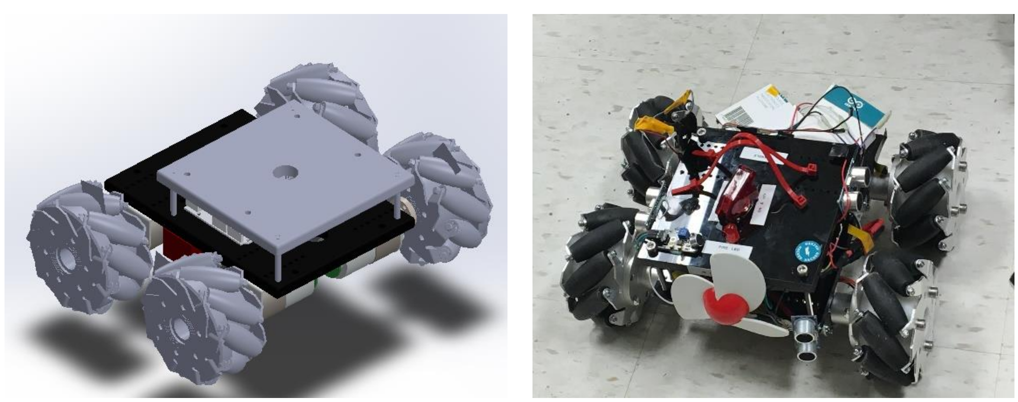

Figure 3. Firefighting Robot: Mechanical design to realization of the system.

Components used include the Arduino board, ultrasonic sensors, motor driver, geared DC motors and mecanum wheels for navigation and locomotion as well as photo-sensors and infrared sensor for fire detection. The system is powered with external lithium polymer batteries. The components are illustrated in Fig.4.

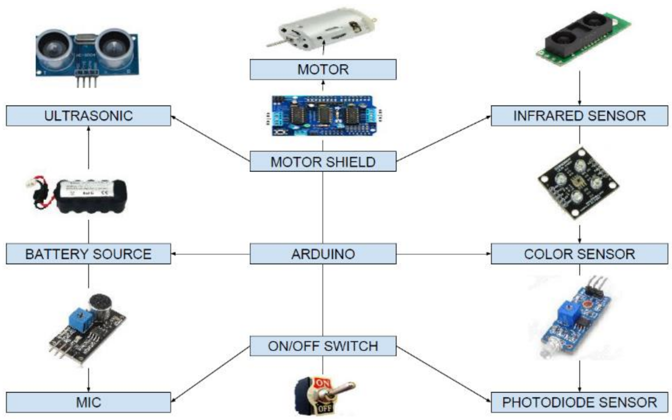

Figure 4. Firefighting Robot: Controllers and sensor suite components

Students successfully searched for various open source examples and learned how to implement the control code themselves by reading blogs and watching videos from reliable sources, with minimal instructor support. During the earlier stages, when students raised a question, they were asked to research it and come up with a solution that they could find including reliable online sources. After then, based on a solution presented by the student, a positive or negative feedback is given to support the student's evaluation and self-learning process. Following these steps, students are encouraged to apply rapid evaluation techniques themselves. For the rest of the project, students are left alone to practice self-learning skills: Only an experimental setup is described for each project so that they could evaluate the quality and accuracy of the final product themselves as well. Given time restrictions, students could successfully implement the control logic of a firefighting robot. Fig. 5 illustrates the flow diagram of the developed system.

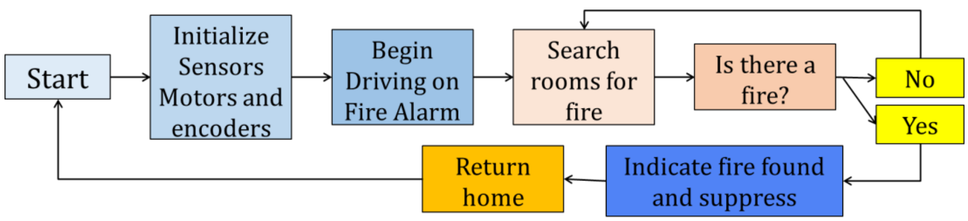

Figure 5. Firefighting Robot: Control system logic

Experiences and Preliminary Results

Preliminary results are positive and showed that students had good experiences in the classes. We found the Arduino microcontroller platform to be effective in teaching students about engineering systems and was easy to use as a component within these systems. Based on informal interactions in class and survey responses, the students also found the Arduino easy to use. The level of students ranged from seniors to graduate with varying microcontroller experiences. The Arduino was appropriate for different backgrounds including students with little microcontroller experience to graduate students who have used other microcontrollers extensively.

Students in the spring 2016 offering of the course were asked to fill out a questionnaire with the following eight questions, rating their responses to them based on their experiences in the course:

- Using the Arduino microcontroller enhanced my understanding of how to design engineering systems.
- Using the Arduino microcontroller enhanced my experience in this course.
- I would choose to use the Arduino microcontroller in the future.
- I had a positive learning experience in the course.
- Describe what you liked most about the Arduino microcontroller.
- Describe what you liked least about the Arduino microcontroller.

Table 1 presents results of the questionnaire. Student response rate was 25%. As can be seen from Table 1, student feedback was positive. We also compared the quality of the final project between two types of offerings, with and without the Arduino. Offerings where the Arduino platform was used showed around 20% increase in the number of students receiving grades of 'A' or 'B'. A similar result was reported in the grades of the final project, with the quality of student final projects improving significantly when the Arduino platform was used.

Table 1. Survey Analysis: Learning with Arduino

Survey Questions	**1**	**2**	**3**	**4**	**5**
1. Using the Arduino microcontroller enhanced my understanding of how to design engineering systems.	0	0	10%	20%	70%
2. Using the Arduino microcontroller enhanced my experience in this course.	0	0	10%	0	90%
3. I would choose to use the Arduino microcontroller in the future.	0	10%	0	10%	80%
4. I had a positive learning experience in the course.	0	0	0	20%	80%

(1) Strongly Disagree, (2) Somewhat Disagree, (3) Neither Agree nor Disagree, (4) Somewhat Agree, (5) Strongly Agree

While this is a small sample to make definitive conclusions, these preliminary results have motivated further study. Feedback from our first offerings are being used to improve the course. We are using the Arduino platform in the same course again during Fall 2016 and will collect additional assessment of its impact on enhancing student learning experiences.

Conclusion

We explored the use of the Arduino platform in the Mechatronics course with the goal of helping students fuse various previous engineering concepts and topics and at the same time improve on their scientific programming skills as well as self-learning skills. Results of our experiences were positive and showed that students had good experiences in the class. We found the Arduino microcontroller platform to be effective in teaching students about engineering systems and was easy to use as a component within these systems while promoting implementation skills. The course will be offered again during spring 2017. Feedback from our first offerings are being used to improve the course. We plan to collect and report on additional assessment of its impact on enhancing student learning experiences.

References

1 Guerin, K. R., Lea, C., Paxton, C., and Hager, G. D., 2015, "A framework for end-user instruction of a robot assistant for manufacturing," IEEE International Conference on Robotics and Automation, 2015, pp. 6167–6174.

2 Bonin-Font, F., Ortiz, A., and Oliver, G., 2008, "Visual navigation for mobile robots: A survey," Journal of Intelligent and Robotic Systems, 2013, 53(3), pp. 263–296.

3 Li, Y., Wang, Y., Geoffrey Chase, J., Mattila, J., Myung, H., and Sawodny, O., "Survey and introduction to the focused section on mechatronics for sustainable and resilient civil infrastructure," IEEE/ASME Transactions On Mechatronics, 2013, **18**(6), pp. 1637–1646.

4 Giurgiutiu, V., Lyons, J., Rocheleau, D., and Liu, W., "Mechatronics/microcontroller education for mechanical engineering students at the University of South Carolina," Mechatronics, 2005, **15**(9), pp. 1025–1036.

5 Stark, B., Li, Z., Smith, B., and Chen, Y., "Take-Home Mechatronics Control Labs: A Low-Cost Personal Solution and Educational Assessment," Proc. ASME 2013 International Design Engineering Technical Conference and the Computer and Information in Engineering Conference (August), 2013,

6 Mynderse, J. A., Shelton J. N., "Implementing Problem-Based Learning in a Senior / Graduate Mechatronics Course," Proceedings of the 121st ASEE Annual Conference and Exposition, 2014.

7 Przybylla, M., "Physical Computing in Computer Science Education," In Proceedings of the 10th Annual Conference on International Computing Education Research (ICER'14*)*, 2014.

8 Russell, I., Jin, K., Sabin, M., "Make and Learn: A CS Principles Course Based on the Arduino Platform," Proceedings of the 2016 Conference on Innovation & Technology in Computer Science Education (ITiCSE'16),2016.

9 Barba, E., Chancellor S., "Tangible Media Approaches to Introductory Computer Science," Proceedings of the 2015 conference on Innovation & technology in computer science education (ITiCSE'15), 2015.

10 Mellodge, P. and Russell, I., "Using the Arduino Platform to Enhance Student Learning Experiences," Proceedings of the 2013 Conference on Innovation & Technology in Computer Science Education (ITiCSE'13), 2013.

Akin Tatoglu is an Assistant Professor of Mechanical Engineering at University of Hartford, CT. He received his Ph.D. degree in mechanical engineering from Stevens Institute of Technology, NJ, in 2015. His research focuses on robotics, collaborative navigation and mapping, stochastic localization, as well as robotics design and control.

Ingrid Russell is Professor of Computer Science at the University of Hartford, CT. Her research interests are in the areas of machine learning, data mining, and computer science education. Her research has been funded by the National Science Foundation, National Aeronautics and Space Administration, and the Connecticut Space Grant Consortium. Russell has served in editorial capacities for numerous Computer science conference proceedings and journal special issues and has chaired several conferences in her areas of research. In addition, she has served in several board leadership positions of national and regional computing organizations.