Paper ID #25856

Work in Progress: Incorporating Microprocessors across the Mechanical Engineering Curriculum

Dr. Lawrence Funke, Ohio Northern University

Dr. Funke received his PhD in Mechanical Engineering from the University of Notre Dame in 2017. He is currently an assistant professor at Ohio Northern University.

Dr. J. Blake Hylton, Ohio Northern University

Dr. Hylton is an Assistant Professor of Mechanical Engineering and Coordinator of the First-Year Engineering experience for the T.J. Smull College of Engineering at Ohio Northern University. He previously completed his graduate studies in Mechanical Engineering at Purdue University, where he conducted research in both the School of Mechanical Engineering and the School of Engineering Education. Prior to Purdue, he completed his undergraduate work at the University of Tulsa, also in Mechanical Engineering. He currently teaches first-year engineering courses as well as various courses in Mechanical Engineering, primarily in the mechanics area. His pedagogical research areas include standards-based assessment and curriculum design, including the incorporation of entrepreneurial thinking into the engineering curriculum and especially as pertains to First-Year Engineering.

Dr. David Sawyers, Ohio Northern University

David R. Sawyers, Jr. is a professor of mechanical engineering at Ohio Northern University, where he teaches courses in general engineering and in the thermal sciences. He received a B.S.M.E. degree from Rose-Hulman Institute of Technology and M.S. and Ph.D. degrees, both in mechanical engineering, from the University of Notre Dame.

Work in Progress: Incorporating Microprocessors across the Mechanical Engineering Curriculum

Abstract

Engineering has been and continues to be a highly cross-disciplinary field. The growth of mechatronics and robotics has served to increase the demand for students with skillsets from multiple traditional fields (joining electrical, computer, and mechanical engineering and computer science). While some schools are adding programs in mechatronics and robotics, this is not always feasible. An alternative is to increase exposure to programming and electrical content in traditionally mechanical engineering courses, such as through the incorporation of microprocessors across the curriculum.

This work investigates the incorporation of microprocessors (specifically Arduinos) into the Mechanical Engineering curriculum. The goals of this effort are to increase students' familiarity level with microprocessor capabilities, to increase exposure to mechatronic systems, to allow for higher fidelity prototypes in class projects, and to provide an accessible and inexpensive way for students to explore applications of what they learn in class. This work reviews efforts to systematically introduce microprocessors into the curriculum, lays the framework for assessment of these intended benefits, and presents initial findings in the form of professor observations, student feedback, and baseline survey data.

Microprocessor content was added to the first three years of the Mechanical Engineering curriculum. It is first introduced to all engineering majors through the two-semester first-year engineering course sequence. Mechanical engineers then use microprocessors again during their fourth semester in a computer applications class and in their fifth semester during an experimental methods class. All of these classes are required.

Preliminary results indicate that the students enjoy the content because of its hands-on nature and the connection of what is learned in the class to real-world applications. Future work, not addressed in this work-in-progress, will further evaluate the effects of including microprocessors in these classes by examining student survey data as cohorts move through the newly implemented curriculum. Evaluating the use of relevant systems in senior capstone projects both before and after implementation will provide particularly meaningful assessment.

1 Introduction

The advent of inexpensive, easily-accessible microcontrollers, such as the Arduino, has opened new opportunities for integrating hands-on activities related to sensors and mechatronic systems into the mechanical engineering classroom. For example, it is now possible to replace a data-acquisition system or programmable logic controller that might cost several hundred (or even several thousand) dollars with an Arduino-based system that could be one-tenth the cost. The limitations of the lower-cost system, such as slower speed or lower accuracy, are less important in the classroom than in an industrial or research environment. In fact, they are often an advantage since students will be able to more easily observe the effects of these limitations on system performance. Another benefit of Arduino-type systems is the diversity of components (actuators, sensors, I/O devices, etc.) that can be easily integrated into mechanical engineering applications. Finally, incorporating an easy-to-use system like the Arduino platform will allow faculty to develop a wide range of activities that focus on specific learning objectives without requiring the students to spend a lot of time on overhead such as learning a new programming language.

While the use of microcontrollers is becoming more common in mechanical engineering courses, the authors are not aware of an intentional cross-course implementation like that discussed herein. The importance of including microcontrollers and the types of activities they allow is seen in the growth of mechatronics and robotics and the associated increase in demand for students with diversified skillsets [1]. Many schools, including Ohio Northern University (ONU), would like to increase such cross-disciplinary offerings [2-5], but do not have the resources needed to add separate programs in mechatronics and robotics. This paper explores how Arduino-based activities were introduced to the Mechanical Engineering curriculum to increase students' exposure, comfort level, and ability in regards to sensors and mechatronic systems.

2 Setting and Context

Microprocessor content in the form of activities and/or projects has been added to four required courses in the Mechanical Engineering curriculum at ONU: ENGR 1041 (Introduction to Engineering 1), ENGR 1051 (Introduction to Engineering 2), ME 2511 (Computer Applications), and ME 3511 (Experimental Methods). The ENGR classes are required of all first-year engineering students and are cross-disciplinary. ME 2511 and ME 3511 are required courses for mechanical engineering majors in the fourth and fifth semesters, respectively.

The first-year-engineering sequence (ENGR 1041 & 1051) has significant design content. In years past, any prototyping was very low fidelity, generally involving paper, wood, etc. and manual actuation. With the introduction of the Arduino content both hardware and software have been integrated into the prototypes, still allowing for the advantages of low-fidelity prototypes [6,7], while increasing exposure to the electromechanical elements and introducing students to control and actuation. ME 2511 is a software intensive course focused on using SOLIDWORKS to teach CAD and MATLAB to teach computer programing. The Arduino content allows hardware to be added to the loop, increasing students' understanding both of the programming content and application of said programming. Finally, ME 3511 traditionally focused on using the LabView software with a few, very expensive data acquisition devices. The addition of Arduinos has allowed the students to spend less time learning to write programs in a very specialized language (LabView) while also introducing a much wider variety of lower-cost sensors. All of these modifications have allowed the students to better understand the close connection between hardware and software and increase their exposure to hands-on, real-world projects.

To support these activities, students receive an Arduino Kit [8] during their first semester, and use this same kit in all four classes. Beginning in the 2018-19 academic year, the Arduino kits are paid for by a course fee, so that each student will own his or her own kit and will retain possession of the kit for use in later courses.

3 Curriculum Modifications

The four courses that were modified provide a natural sequence within the first three years of the Mechanical Engineering curriculum, allowing microcontrollers to be introduced efficiently and with progressively deeper application. The focus, however, is not meant to be on the Arduino microcontrollers themselves, but rather on their use to accomplish tasks inherent to the profession of engineering, such as acquiring meaningful data or controlling mechatronic systems. The Arduino is simply a cost-effective way of introducing students to the role of such devices in engineering. The flexibility and ease-of-use of the Arduino means that it can be used to demonstrate a wide range of concepts in multiple courses while allowing the students to gain confidence by working with the same hardware in multiple settings.

The course modifications were carried out over 2 semesters with ENGR 1051 (Introduction to Engineering 2) and ME 2511 (Computer Applications) in the Spring semester of 2018 and ENGR 1041 (Introduction to Engineering 1) and ME 3511 (Experimental Methods) in the Fall semester of 2018. The following subsections detail the specific activities and projects implemented in each course.

3.1 First-Year Engineering Sequence

The first-year engineering curriculum at ONU consists of two semester long courses, ENGR 1041 and ENGR 1051, which are taken by all students in the college of engineering apart from computer science. The courses focus on engineering skill building, mindset, and design with coverage of a wide array of engineering software and hardware tools. Prior to this modification, portions of the course had introduced basic programmatic thinking through an introduction to MATLAB. Basic electrical prototyping and measurements were also introduced through a series of labs. No microcontroller content was present in the previous iterations of the course sequence.

3.1.1 ENGR 1041 – Introduction to Engineering 1

In ENGR 1041, modifications focused on the addition of a week-long module on microcontrollers and basic programmatic structures. Students were given their kits at the beginning of the module and instructed to create an account on TinkerCAD (TinkerCad.com). TinkerCAD is a free, browser-based package from AutoDesk that has two core functions – basic 3D modeling and electrical simulations. The electrical side of the program allows users to click-and-drag various electrical components into a workspace, wire them together, and simulate the behavior of the resulting circuit. Most importantly for this effort, the system also includes a simulated Arduino Uno, which can be wired into the circuit along with any of the assortment of other electrical components. The simulated board can also be programmed, either through the in-program graphical interface or by uploading an Arduino script. In this way, the program itself may also be simulated – creating a kind of virtual test-bench environment.

The graphical interface for programming the board is similar to the click-and-drag introductory programming games that have grown in popularity over recent years. Users can select from any of several programming structures, such as a loop or if structure, and drag in variables, nested structures, or input-output blocks to interface with the connected circuit components. In this way, users build a program flowchart, that the interface then converts to an Arduino script. The program may be simulated in-browser or downloaded and loaded onto a physical board through the Arduino IDE. If desired, the script may also be opened in the IDE and modified directly, allowing even deeper exploration of the program.

The module as introduced in this modification walked students through a series of activities, mostly self-paced and self-guided. Students built, programmed, and simulated various circuits within TinkerCAD, then replicated those circuits with physical hardware, downloaded their programs to the physical Arduino board, and then demonstrated that the circuit was working as expected. Complexity of both circuit arrangement and program elements increased as students moved through the module, eventually moving to downloading TinkerCAD scripts and then modifying those scripts in the Arduino IDE before loading them onto the board. Topics covered included basic input/output using the various types of Arduino pins, programming logic (AND/OR) and IF structures. The module culminated in a mini-design activity, in which students were asked to use a set of components out of their kits to build and program a circuit of their own design. The circuit was required to be a simplified version of something that would solve some need to be identified by the student, such as a light-sensing switch to automatically turn on a lamp. Full details of the logistics and details of the activities are published in a parallel work. [9]

3.1.2 ENGR 1051 – Introduction to Engineering 2

Modifications in ENGR 1051 were focused on integrating a two-day activity using MATLAB to program Arduino-based tanks to move on command. It is important to note that the modification as described here was deployed in the academic year prior to the introduction of the above EGNR 1041 modifications. As such, the activity as described will be adjusted in future iterations to better leverage the prior exposure provided by that module.

The first day included two activities. The first focused on introducing the Arduino itself, including the board layout, how to connect the robot components, and an introduction to basic Arduino-interaction commands in MATLAB. These included creating an Arduino object, reading analog pins, and writing both digital and PWM (Pulse Width Modulation) pins. The second activity walked students through connecting to the board using the Arduino IDE and uploading blink then connecting to the board using MATLAB and running a provided script. Students were guided through the various script components and instructed to explore the impact of altering various settings within the script. Students were also guided in altering the code by copying provided additional commands to enable additional control of the robot motion. The second day included a third activity, focused on altering the provided code through a series of steps. The first step added a counter to register how many times a button was pressed, intentionally designed such that a press-and-hold action registers as continuously counted presses with each execution of readVoltage. The second step added functionality to read a button in a useful way, registering discrete button presses rather than continuously counting. The third step asked students to use the programming content that they had previously learned to alter the script to perform different actions depending on how many times the button has been pressed. This involved a more complex code structure and a good understanding of what the various Arduino inputs control.

3.3 ME 2511 – Computer Applications

The next opportunity for Mechanical Engineering students to interact with microcontrollers comes in their fourth semester, when they take ME 2511. This course traditionally focused on 3D modeling and MATLAB programming with one or two activities using Arduino-based tanks. Though these activities were limited to programming a given hardware setup. Two Arduino-based activities (LED control and motor control) and a semester-long Arduino-based NABC project [10] were added to ME 2511, allowing a better connection between modeling, software (MATLAB), and hardware (the Arduinos, sensors, and motors used in the project). Including these activities allows the students to better see how the topics are connected and make their own connections between topics that often seem disparate at first. Making such connections has been shown to improve students' understanding and retention of material [11].

The LED control activity involved the students using wires and a bread board to connect a potentiometer (POT) and an LED to the Arduino board. The circuit layout and commands were largely prescribed. The goal of the activity was to ensure that the students could "talk" to the board, read and write voltages, and use MATLAB to analyze data received from the board. The activity was split into two tasks the students had to accomplish. The first task was simply to read the voltage from the POT and write this voltage to the LED using a PWM pin. This task could be completed without the use of the Arduino board, so the second task focused on taking advantage of having the Arduino in the loop. The students were instructed to write code that allowed the LED to go from off to fully bright to off again as the POT was turned from one extreme to the other. This required basic programming skills covered previously in the class. This simple activity is meant as an introduction more than anything else. The students who took the class in the Spring of 2018 had not seen microcontrollers previously in the curriculum, so it was necessary. In subsequent offerings this activity will likely be included as an optional tutorial rather than a required activity since it will be largely redundant to the activities incorporated into the first-year engineering courses.

The motor control activity taught the students how to wire and control a brushed DC motor using the Arduino kits and also consisted of two tasks. The first task was a step-by-step guide for wiring the motor and commands to make the motor spin in either direction at different speeds. The second task required the students to take an input voltage from a POT and make the motor move in a certain direction at varying speeds based on this information. This part is nearly identical to the second task of the LED control activity, only multiple output voltages are needed to control the motor, instead of just one for the LED. The students were also tasked with creating a sub-function inside of MATLAB that would take the input voltage and return the voltages to be sent to the pins that controlled the motor. This programming aspect required knowledge covered previously in the class.

The Arduino content in computer applications culminated in the NABC semester-long project that allowed students to apply everything they had learned about Arduinos thus far. NABC, which stands for Need, Approach, Benefit, Competition, is a framework for developing and expressing a value proposition [10]. The students were tasked with completing a semester-long project that had them identify all of these things. The project consisted of a request for proposal and four deliverables. The first deliverable involved identifying a need in the community and how solving this need would benefit the stakeholders. The second deliverable was a design proposal where the students presented their approach. Deliverable three was a demonstration of the bench-top prototype they made to solve the need. The prototype had to use the board, at least two sensors, and a motor. The final deliverable was a full report, including a value proposition statement based on the NABC frame work, such as "Our (product) helps (specific users with a need **N**) who want to (achieve benefits **B**) by (approach **A**) better than (competition **C**)." This deliverable also included a presentation with a video demonstrating their prototype. This project allowed the students to combine the programing and 3-D modeling they learned in the course, with the Arduino knowledge they gained to make a physical prototype to solve a real need.

3.4 ME 3511 – Experimental Methods

Taken in the fifth semester, ME 3511 has traditionally focused on using the LabView software as an interface for National Instruments USB-based data acquisition (DAQ) systems, in conjunction with industrial quality sensors. While LabView is commonly used in industrial DAQ applications, and is especially capable of interacting with National Instruments hardware, it is a complex, expensive programming environment that has a fairly steep learning curve. In addition, the number and variety of sensors that could be used in class was limited by the high cost of industrial-grade equipment.

There are a number of advantages of using the Arduino platform for teaching experimental methods to mechanical engineering students. First, the low cost of the hardware allows each student access to their own equipment. As a result, it is possible to develop in-class exercises where the students develop data acquisition systems under the direct supervision of the instructor, rather than scheduling a time outside of class to use one of a limited number of experimental setups. In addition, the low cost allows students to use a much wider range of sensors than was previously possible. Even the limitations of the Arduino can be used as advantages since students will have to address issues such as limited memory or low-resolution analog-to-digital conversion that might not even be noticed when using the more expensive systems.

Another advantage of the Arduino platform is the amount of source code and information that is readily available online. In ME 3511 students use the Arduino IDE programming environment to write programs to acquire data from a variety of sensors and interact with other hardware such as LCD output screens. Due to the open-source nature of the Arduino, and the enthusiasm of the Arduino community, it is usually possible to find existing code online from which to begin almost any task. As a result, there is a relatively shallow learning curve that allows students to accomplish a great deal without spending a significant amount of class time teaching a new programming language.

Since the department already possesses the LabView license and National Instruments hardware, the Arduinos were used to supplement and expand the course rather than replace previous content completely. The LabView component of the course was reduced to a level that still allows students to write programs to acquire and process data from a variety of sensors, but without achieving the same degree of advanced proficiency as in previous course offerings. This freed up time that was used to obtain a basic competency with the Arduino IDE, and to introduce several new in-class activities, such as:

- Acquiring analog data from a three-axis accelerometer. This activity requires students to understand how the Arduino 10-bit Analog-to-Digital Converter converts an analog input into a digital value, to perform simple calculations, and to use serial communications to observe the results.
- Measuring temperature using a thermistor. This activity requires students to build a bridge circuit to convert resistance into voltage, and then use the Arduino to acquire and convert the results into temperature.
- Measuring angular speed using an optical rotary encoder. This activity requires students to use the interrupt input feature of the Arduino to count edges of a digital signal, and perform the calculations necessary to determine angular speed.
- Measuring pressure using a pressure transducer. This activity requires students to acquire an analog voltage from an inexpensive pressure transducer (designed for automotive applications). The pressure is varied using a large syringe connected to the transducer by a flexible tube. The results are displayed using a two-line LCD.

These exercises introduce the students to a wide range of data acquisition issues, many of which are not visible when using high-end systems (such as analog-digital conversion).

One aspect of ME 3511 that has traditionally been difficult to achieve is the inclusion of experimental design. Selection of equipment is a critical aspect in designing an experiment; however, in the past, hardware choices have been limited by the cost of the high-end equipment used. Because of the wide range of inexpensive sensors that can be used with the Arduino platform, it is now possible to include a much more open-ended experimental design assignment in the course. During the Fall semester of 2018, students were required to identify an experimental application for one sensor (from a selection provided by the instructor). Each team developed a test plan that included:

- an objective statement
- definition of variables, including expected range and desired accuracy of measurements
- an equipment list, including expected cost
- an experimental procedure
- description of calculations to be performed
- the expected presentation format for results

In addition to the written test plan, each team was also required to construct a simple “proof-of-concept” device, in which they used their Arduino kits with their chosen sensor to demonstrate that they could acquire and display measurement data – although not necessarily accomplishing their specific test objective. This allowed the project to include a hands-on component, without limiting the scope of the proposed tests to what could be achieved with the available hardware.

4 Preliminary Findings

Work to date has focused on implementing the new content without much regard to quantifying its effect. A pre-post survey was implemented in ENGR 1041 in Fall of 2018 where the class was given a survey before and after the new Arduino content was delivered. This survey was conducted as part of a study by Sierzputowski et. al [9] exploring the impact of microcontroller exposure on first semester students. The surveys asked students to rate their understanding and confidence in performing programming, circuit, and Arduino activities on a five-point scale.

Mechanical engineering majors had an average increase of 1.20 points in these areas from before the new content to after.

Qualitative data was collected in the form of student feedback and instructor observations. In ENGR 1051 students informally expressed positive attitudes toward the activity, enjoying the opportunity to have a tangible hardware aspect to the coding content they had been learning. They were almost unanimous in suggesting that the activity be retained for future years. Notably, electrical and computer engineering students were especially positive. This is especially important for this course given that much of the other course content is more approachable for the mechanical and civil students.

Similar positive attitudes were expressed by students in ME 2511. The instructors observed that students were always excited to work on the project. Some students went as far as to teach themselves about sensors not covered in class and at least one group decided to learn the Arduino IDE so that they could use more advanced sensors. This sort of voluntary investigation is an exciting and encouraging demonstration of intellectual curiosity and self-guided learning among students.

Enjoyment and motivation to learn more about Arduinos was also expressed in the following student comments from the course evaluations for ME 3511:

- "I also liked the Arduino portion of the course. I feel more emphasis could be put on this in the future, such as more labs involving it. The Arduino language was pretty easy to understand and is easily applied to real world scenarios."
- "Wish we would have gotten into the Arduino kits a little more, possibly doing more projects with them."
- "Learning about Arduino was fun."

Perhaps even more significantly, the only negative student comments related to using the Arduinos were those that called for including more Arduino content in the course.

5 Framework for assessment

Though the preliminary findings have been overwhelmingly positive much remains to be done in assessing the main objective of this project - to better prepare the students to use mechatronic systems in their careers. In order to ensure that the students are achieving this outcome, in addition to enjoying the hands-on content, a more formal assessment needs to be developed and implemented. The authors are proposing a two-fold approach. First, to administer surveys before and after the Arduino content in all of the modified courses. The surveys will cover prior experience, understanding of content, motivation, and confidence in success for several relevant topics. Appendix A contains the proposed survey. Analyzing pre- and post-surveys for a given course will indicate the success of a particular course in achieving the desired outcomes. Looking at pre-surveys course to course will show how much information students are retaining and allow the effects of multiple exposures to be investigated.

Analyzing capstone projects will provide a secondary assessment point. If the course modifications are meeting the desired outcomes, it is expected that mechanical engineering senior capstone projects will contain higher quality prototypes when it comes to related content such as sensors, motors, and mechatronic systems. Assessing the projects over the next several years will give insight into the relationship between the amount of microprocessor exposure and quality of capstone prototypes. Specifically, the number of groups using sensors or mechatronic systems in their prototypes will be recorded. The grade given for the quality of these prototypes and anecdotal observations will be recorded for these groups as well. The class of 2019 (current seniors) did not see any of the modified content, 2020 will see it in ME 2511 and ME 3511, 2021 will see these two plus ENGR 1051, and 2022 and on will see all modifications. Collecting data over the next four years will provide insight into how increased exposure affects prototype quality. An additional comparison point for the effects of repeated exposure could be gained by comparing mechanical engineering projects to projects from other majors, since all majors will see the content in ENGR 1041&1051, but not in the other courses.

6 Conclusions

Overall the project has been a significant success. At the program level, introducing Arduino activities at multiple stages of the curriculum has been done in a way that provides consistency while also addressing a wide variety of software and hardware applications. The flexibility and ease-of-use of the Arduino platform has allowed it to be used in different courses to achieve different learning outcomes, while also providing a familiar environment that allows students to learn new concepts while gaining confidence by building on past experience.

This paper presented the authors' work in using the Arduino platform to systematically introduce activities across multiple levels of the Mechanical Engineering curriculum. The preliminary efforts have been very successful in both increasing student engagement and in achieving specific learning objectives of the involved courses. The efforts described herein will be continued in the future by refining the activities described, adding new activities, and continuing to coordinate across the different courses to ensure that the activities are leading to the goal of continuing to introduce real-world problems and applications of sensor and mechatronic systems throughout the mechanical engineering curriculum. The methods proposed in Sec. 5 will be used to assess the effectiveness of this endeavor as students proceed through the modified courses.

Acknowledgements

The authors would like to gratefully acknowledge the Ohio Space Grant Consortium for partially funding this work through the Curriculum Innovation grant program.

References

[1] Fromm, E. (2003), The Changing Engineering Educational Paradigm. Journal of Engineering Education, 92: 113-121. doi:10.1002/j.2168-9830.2003.tb00749.x

[2] Sala, A. L. (2013, June), *Infusing Mechatronics and Robotics Concepts in Engineering Curriculum* Paper presented at 2013 ASEE Annual Conference & Exposition, Atlanta, Georgia. https://peer.asee.org/19765

[3] Cherng, J. G., & Li, B. Q., & Natarajan, N. (2013, June), *Development of a Senior Mechatronics Course for Mechanical Engineering Student* Paper presented at 2013 ASEE Annual Conference & Exposition, Atlanta, Georgia. https://peer.asee.org/19438

[4] Tomovic, M., & Tomovic, C., & Jovanovic, V. M., & Lin, C. Y., & Yao, N., & Katsioloudis, P. J. (2017, June), *Integrative Experiences through Modeling and Simulation of Mechatronic Systems* Paper presented at 2017 ASEE Annual Conference & Exposition, Columbus, Ohio. https://peer.asee.org/28570

[5] Djuric, A., & Jovanovic, V. M., & Yaprak, E., & Chen, W. (2017, June), *Learning Module on Electric Motors Modeling, Control, and Testing (LM-EMMCT)* Paper presented at 2017 ASEE Annual Conference & Exposition, Columbus, Ohio. https://peer.asee.org/28613

[6] Christie, E. J., & Jensen, D. D., & Buckley, R. T., & Menefee, D. A., & Ziegler, K. K., & Wood, K. L., & Crawford, R. H. (2012, June), *Prototyping Strategies: Literature Review and Identification of Critical Variables* Paper presented at 2012 ASEE Annual Conference & Exposition, San Antonio, Texas. https://peer.asee.org/21848

[7] Wettergreen, M., & Hinds, T. J. (2016, June), *Rapidly Deployable Prototyping Activities to Teach Engineering Design.* Paper presented at 2016 ASEE Annual Conference & Exposition, New Orleans, Louisiana. 10.18260/p.26036

[8] Elegoo EL-KIT-003 UNO Project Super Starter Kit with Tutorial for Arduino available from Amazon. https://www.amazon.com/Elegoo-Project-Starter-Tutorial-Arduino/dp/B01D8KOZF4/ref=sr_1_1_sspa?s=electronics&ie=UTF8&qid=1511792853&sr=1-1-spons&keywords=elegoo+arduino+uno&psc=1 [Accessed: 24- Jan 2019]

[9] Sierzputowski, W., France, T., Katterle, K., Dazey, T., & Palmer, J. (2019, March 22-23). *Avoiding the Syntax: An Accessible Approach For Introducing First-Year Engineering Students To Microcontrollers.* Paper presented at the American Society for Engineering Education North-Central Section Conference, Grand Rapids, MI.

[10] Carlson, C., and Wilmot, W. "Innovation: The Five Disciplines for Creating What Customers Want," Crown Business, ISBN 13:978-0-307-33669-9, 2006.

[11] Haughery, J. R., & Raman, D. R., & Kaleita, A. L., & Freeman, S. A. (2017, June), *Toward Understanding the Impacts, Whys, and Whats Behind Mechatronic-based Projects and Student Motivation* Paper presented at 2017 ASEE Annual Conference & Exposition, Columbus, Ohio. https://peer.asee.org/29037

Appendix A: Proposed survey to be given before and after Arduino content in each modified course.

Name: ______________________________________ Section: 9AM 10AM

1. How often have you programed Arduino (or other) microcontrollers before this class. (Circle one)	Never	Once or Twice	Several Times	Regularly

For the statements below, please circle one number.

1 = “strongly disagree,” 2 = “disagree”, 3 = “agree,” 4 = “strongly agree”

	Strongly Disagree			Strongly Agree
2. I have a basic understanding of computer programming structure and logic (e.g., if & else statements).	1	2	3	4
3. I have a basic understanding of how to use a breadboard to prototype electrical circuits.	1	2	3	4
4. I understand the basics of how an Arduino board works.	1	2	3	4
5. I feel confident that I could program a microcontroller and construct a circuit to complete a useful task.	1	2	3	4
6. I enjoy programming microprocessors.	1	2	3	4

Please answer all of the following questions by selecting the answer that best represents your beliefs and judgment of your current abilities. Answer each question in terms of who you are and what you know today about the given tasks.

Rate how motivated you would be to perform the following tasks

	Not At All Motivated			Highly Motivated
7. Use computer programming in an engineering context.	1	2	3	4
8. Prototype a design involving electrical components (e.g. sensors, switches).	1	2	3	4

Rate how successful you would be in performing the following tasks

	Not At All Successful			Very Successful
9. Use computer programming in an engineering context.	1	2	3	4
10. Prototype a design involving electrical components (e.g. sensors, switches).	1	2	3	4

Rate how confident you are about performing the following tasks

	Not At All Confident			Very Confident
11. Use computer programming in an engineering context.	1	2	3	4
12. Prototype a design involving electrical components (e.g. sensors, switches).	1	2	3	4