An Arduino-Based Hardware Platform for a Mechanical Engineering Sophomore Design Course

Dr. Mark David Bedillion, Carnegie Mellon University

Dr. Bedillion received the BS degree in 1998, the MS degree in 2001, and the PhD degree in 2005, all from the mechanical engineering department of Carnegie Mellon University. After a seven year career in the hard disk drive industry, Dr. Bedillion was on the faculty of the South Dakota School of Mines and Technology for over 5 years before joining Carnegie Mellon as a Teaching Faculty in 2016. Dr. Bedillion's research interests include control applications in robotics and STEM education.

Dr. Karim Heinz Muci-Kuchler, South Dakota School of Mines and Technology

Karim Muci-Küchler is a Professor of Mechanical Engineering and Director of the Experimental and Computational Mechanics Laboratory at the South Dakota School of Mines and Technology (SDSM&T). Before joining SDSM&T, he was an Associate Professor of Mechanical Engineering at the University of Detroit Mercy. He received his Ph.D. in Engineering Mechanics from Iowa State University in 1992. His main interest areas include Computational Mechanics, Solid Mechanics, Biomechanics, Product Design and Development, and STEM Education. He has taught a variety courses at the undergraduate and graduate level, is author or co-author of over 70 publications, is co-author of one book, and has done consulting for industry in Mexico and the US. He can be reached at Karim.Muci@sdsmt.edu.

Mr. Walelign Messele Nikshi, South Dakota School of Mines and Technology

Mr. Nikshi is a Ph.D. candidate in the department of Mechanical Engineering at South Dakota School of Mines and Technology. He received his B.Sc. degree in Mechanical Engineering from Arba Minch University (Ethiopia), and M.Sc. in Mechanical Automation and Mechatronics from University of Twente (The Netherlands). He has 2+ years experience in high tech industries where he worked as a design engineer. His research interests include control of mobile robots, control of Mechanical systems, and Mechatronics.

An Arduino-Based Hardware Platform for a Mechanical Engineering Sophomore Design Course

Introduction

Many institutions offer a freshman or sophomore level design class in mechanical engineering that focuses on the product development process. While nearly all modern products contain sensors, actuators, and control algorithms, the projects featured in these early design courses typically focus on simple mechanical parts. A substantial limiting factor is the students' lack of exposure to fundamental electronics principles, which makes integrating basic sensors and actuators difficult. This paper describes an approach to integrating mechatronic systems in a sophomore mechanical engineering design experience.

A hardware stack consisting of an Arduino, various shields for sensing and actuation, and a set of plug-and-play sensors and actuators were selected by the authors to minimize students' struggles with implementation. The students were provided with several laboratory activities to familiarize themselves with the hardware, sample code that can be used to drive actuators and read sensor inputs, and a user manual created specifically for the hardware used in the class. Students benefit by being able to tackle more interesting problems for their design projects; instructors benefit by having well-defined subsystems that can be used to spur students' thinking about design for complex systems. After describing the hardware stack and the design decisions that led to its selection, this paper provides results in terms of students' self-efficacy and attitudes towards the use of the hardware platform. The results show that the students have been positive about this new approach to teaching sophomore design, while offering suggestions for improving the experience in the future.

Much work has been done on the use of Arduino hardware to teach mechatronics and controls concepts [1-9]. Among these the most closely related to the current work is [7], which describes the selection of a hardware kit for teaching feedback control that emphasizes usability. Typical uses for Arduino-based educational platforms outside of control systems and mechatronics focus on embedded systems [10] or electrical engineering education [11, 12]. Other efforts have discussed the use of making in design education, in which the Arduino is one enabling component [13, 14]. Finally, some capstone design courses have begun using the Arduino as an enabling component [15, 16]. This work has connections to much of the existing work on Arduino for design, but differs in that it targets students in a sophomore-level class instead of the typical senior-level implementations.

This paper is organized as follows. First, a description of the course is provided along with a glimpse into the curriculum structure and student backgrounds. Second, a description for the hardware platform is given along with the instructional strategy for introducing the platform and its components. Third, two sample semester projects are presented along with descriptions of final student projects. Fourth, the results of a post-course survey are provided to measure students' self-efficacy and performance on basic technical questions related to microcontroller interfacing.

Course description and student backgrounds

The mechanical engineering curriculum at the South Dakota School of Mines and Technology (SDSM&T) includes a four-credit sophomore-level course (ME-265/265L Product Design and Development) that introduces the product design and development process [17-20]. The course focuses on the activities corresponding to the concept development phase and a semester-long capstone project allows student teams to apply what they are learning to the conceptual design of a simple product. Product Design and Development by Ulrich and Eppinger [21] is used as the course textbook and the product development process considered is for “market pull” products of low to moderate complexity. The main topics covered in the course include:

- Introduction to the product development process.
- Product planning.
- Introduction to basic systems thinking and systems engineering concepts.
- Identification of customer needs.
- Setting target specifications.
- Concept generation.
- Product architecture.
- Concept selection.
- Prototyping.
- Concept testing.

The course is structured as a two-credit lecture portion with a corresponding two-credit lab. Lectures are used to provide instruction in the main course topics, whereas labs are used to give students dedicated time to work on their design projects. The lab timeline is aligned with the lectures, so that students work through the product development process as it is presented. Some of these lab sessions early in the class were used to introduce the Arduino hardware kit described in this paper.

Like most first semester sophomore students, the students at SDSM&T have some limited preparation in engineering fundamentals. The freshman curriculum at SDSM&T includes the science / mathematics sequence common at most universities along with two courses that introduce mechanical engineering concepts and one that introduces the C programming language. The introductory mechanical engineering courses cover a brief introduction to engineering fundamentals (vectors, static equilibrium, etc.) along with training on manufacturing processes and CAD software. The introduction to C class is unlike most in that it focuses on programming applications in mechatronics. The students purchase a hardware kit based on the PCDuino [22], a microcomputer that supports multiple programming languages and allows easy interfacing with sensors and actuators. The motivation for this sort of programming introduction is twofold: it allows students to grasp the engineering relevance of programming concepts, and it provides a more practical foundation for a required junior-level mechatronics course.

The decision to add a small hardware component to the sophomore design class came from the desire to introduce the basics of systems engineering and the interest in providing more exposure to programming concepts throughout the curriculum. Given students’ limited background in the sophomore year, it is difficult to come up with purely mechanical projects that involve interacting subsystems; the use of a microcontroller-based platform immediately allows such interactions. The use of microcontrollers also acknowledges the reality that many modern products contain some degree of sensing, computation, and actuation and allows the class to take on more relevant projects. From a curriculum standpoint, the change to a course project involving a microcontroller also helped to fill a gap in programming usage in the sophomore year. Complaints were registered from both instructors of the junior-level mechatronics course and its students about the time-delay between students' learning of programming concepts and their practical application. By allowing a more consistent programming experience, the goal was to improve student's retention of programming knowledge and therefore outcomes in the mechatronics course. This paper will present results on student attitudes towards an upcoming mechatronics course after taking the sophomore course, but unfortunately results on improving mechatronics outcomes are not yet available.

Arduino hardware and instructional strategy

The design goals of the hardware stack were to be able to read analog and digital sensors, drive small DC and servo motors, and interface with a commercially available game controller to enable remote control. Given student backgrounds, all components would need to be plug-and-play, i.e. all signal conditioning would need to be contained in the hardware stack or on components themselves. After considering various hardware platforms (including the pcDuino used in the introductory programming class), a simple Arduino-based hardware stack was developed for the course. The large user and support base for the Arduino along with its widely available plug-and-play sensors and actuators were the deciding factors in choosing the Arduino solution.

Taking into consideration the limitations imposed by the level of the course, the time constraints, and the available budget, at the beginning of the semester each team was provided an Arduino Uno based programmable controller that was fully tested and ready to use. The hardware of the control unit (see Figure 1) consisted of an original Arduino Uno board or a compatible SparkFun RedBoard, an Adafruit motor/stepper/servo shield, an Adafruit 16-Channel 12-bit PWM servo shield, a SparkFun or an Arduino USB host shield, a Grove base shield, a Bluetooth 4.0 USB module, a Sony Dual Shock 3 PS3 controller, and 6V and 12V battery packs. The total cost of this hardware stack is approximately $250. Several class sessions were devoted to give a general overview of the hardware mentioned above, provide the knowledge needed to start using the control unit, and present some programming examples demonstrating how to use the controller for different tasks such as driving multiple motors and servos. The only assumption on the capabilities of incoming students is that they have basic familiarity with the C programming language.

In addition to the control unit mentioned above, hardware items such as motors, servos, wheels, gears, and other mechanical and structural components were purchased to facilitate the fabrication of prototypes. The components, which consisted mostly of VEX Robotics EDR parts, could be used by the student teams throughout the semester. Having those parts available removed the need to select motors and machine components, which are skills that are typically not expected of sophomore-level students.

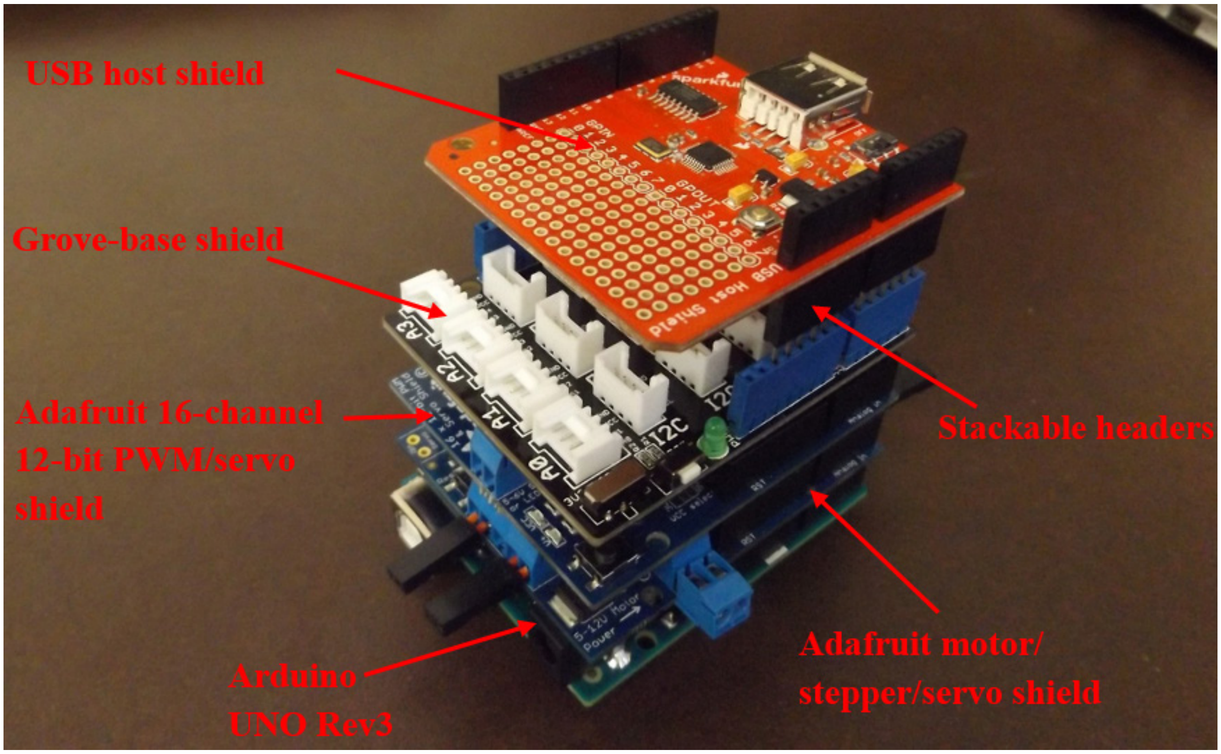

Figure 1. Programmable controller for sophomore capstone design projects.

While various resources are available for learning about the Arduino and the shields used in the kit in isolation, students beginning in their study of microcontrollers might have difficulty in synthesizing that information. To speed their development, a custom user manual was developed for the specific set of shields provided for the capstone project. The manual begins by discussing installation of the Arduino development environment along with options for powering the Arduino board. Next the Grove shield is introduced along with code to read the various sensors in the kit and light LEDs using the common Arduino commands. The other shields in the kit are then discussed in order, with sample code for driving DC motors, driving servo motors, and using a SONY PlayStation 3 controller as an input device. The use of a gaming controller is a convenient way to enable human-in-the-loop control; while some degree of autonomy may be required in the final project, full autonomy is an unreasonable goal for sophomore-level students.

The philosophy of the user manual and associated course materials is not to replicate a mechatronic design course, but rather to provide students with the minimum building blocks necessary to read analog and digital sensors and drive low-power motors. With the sample code provided and the simple exercises that follow, students will have access to snippets of code that can be combined to generate complex behavior with very little prior programming knowledge. The details of analog-to-digital conversion, PWM, H-bridge drivers, and I^2C communication are left for proper courses on mechatronics. By controlling the hardware used, students can gain confidence while avoiding complexities due to motor current limitations, I^2C address conflicts, and circuit layouts. The full user manual along with a lecture presentation introducing the hardware kit may be accessed at http://seed.sdsmt.edu/.

In total, five 2-hour lab sessions were dedicated to introducing the hardware kit and providing basic instruction in its usage. The first lab session was used to provide a brief introduction to the Arduino platform and the Arduino Uno, installing and becoming acquainted with the Arduino IDE, and learning how to use the serial monitor. The second lab session was used to introduce the Grove base shield, learn about writing and reading digital signals, and presenting the concept of PWM. The third lab session dealt with reading analog signals, using the Grove LCD screen to familiarize students with I^2C, and exploring different Grove sensors and actuators. The fourth lab session focused on learning about the motor and servo shields. Finally, the fifth lab session focused on learning how to use the USB host shield together with the SONY PS3 controller.

Course projects

The course centers around the development of a product concept and building a physical prototype of the concept selected; this section provides details on two projects that were run in the fall 2016 and fall 2017 semesters. Course projects were designed with two somewhat conflicting goals: projects should represent real-world products with stakeholders that students can interact with, and projects should be amenable to a small-scale demonstration that students can construct using the components in the hardware kit. In the 2016 implementation, a two-step process was used with conceptual design being at a different scale than the final demonstration. In the 2017 implementation the product was chosen to be at the scale of the kit.

For the first implementation, the instructor decided to use small, remote-controlled ground robots for urban search and rescue (USAR) missions to provide a context for the semester's product development activities. USAR involves the location, rescue (extrication), and initial medical stabilization of individuals trapped in confined spaces. Structural collapse is most often the cause for people being trapped, but individuals may also be trapped in transportation accidents, mines, or collapsed trenches. Ground robots are often used to perform tasks that humans cannot.

Taking into consideration the limitations of the Arduino kit and the need to keep the project scope reasonable, the project was divided into two phases. For the purposes of identifying customer needs, performing a competitive benchmarking, and setting target specifications, the student teams were asked to proceed as if they were part of a company that is planning to introduce a small, remote-controlled ground robot for USAR into the market. Once the teams completed those activities of the concept development phase, the focus of the project was shifted to a small-scale design competition involving tasks that had some resemblance to an urban search and rescue mission. Table 1 presents the project mission statement that was provided to the students at the beginning of the semester.

To help students learn about the capabilities and limitations of the Arduino kit, the teams were asked to use VEX components to build a simple robot (see Figure 2) that they could drive using the Arduino kit. This activity was completed before the information about the small-scale design competition was provided to the students so that they could understand the need to make the transition to the second phase mentioned above.

Table 1. Mission statement for the first phase of the USAR related capstone project involving the Arduino kit.

Product Name	SARBOT-G.
Product Description	Small ground robot used primarily for search and rescue (SAR) operations.
Key Business Goals	• Proof-of-concept prototype ready by the end of “Month & Year” (Note: replace “Month & Year” by the due date). • Selling price like those of the most commonly used small SAR ground robots. • Has distinctive features that make it an attractive option over alternatives.
Primary Market	• Search and rescue teams.
Secondary Markets	• Military / Department of Defense (DoD). • Law enforcement agencies.
Assumptions	• Easy to carry, deploy, and operate. • Works in different types of terrain and weather conditions. • Easy to reconfigure for the needs of the mission. • Can operate continuously for several hours. • Highly reliable and durable. • Requires minimal maintenance. • Easy to repair. • Wireless remote control. • Complies with applicable standards and regulations.
Constraints	• Uses a control platform that the “company” already developed and tested. • Uses existing technologies.
Stakeholders	• End users. • Purchasers. • Manufacturing department. • Service centers. • Sales department. • Resellers.

Unlike some of the traditional design competitions that provide a relatively large number of constraints and/or target specifications, the competition document that the students received only provided a general description of a “scaled down” SAR scenario and the tasks that their robot was expected to perform. The teams were asked to use the customer needs and the target specifications that they identified during the first phase of the project as a guide for all the other aspects that needed to be considered.

To understand the implications of the approach, let us consider the weight of the robot as an example. The information about the competition didn’t specify a requirement for that metric. However, if a team did a good job during the process of identifying customer needs, the students should have identified that the end users wanted a robot that was lightweight and that this was a very important product characteristic for them. Consequently, the team should have considered that requirement during concept generation and selection even though it was not mentioned in the document about the competition that the team received.

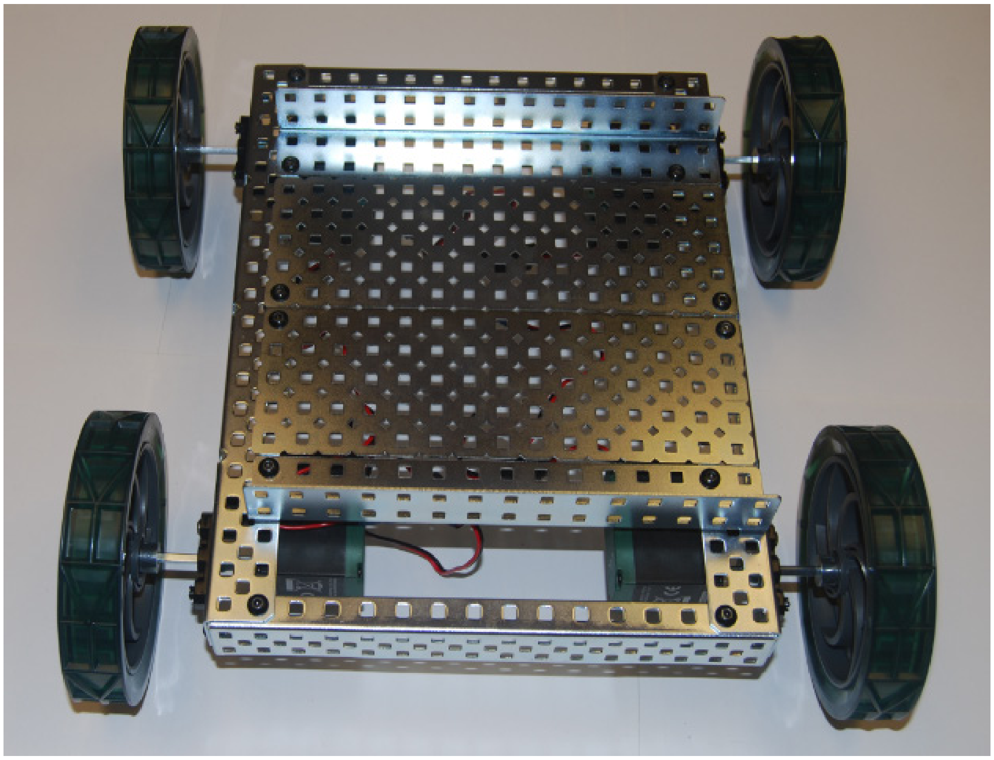

Figure 2. Chassis and drivetrain corresponding to a simple robot that the students built as part of an in-class activity.

The scaled-down competition version of the project had students design robots that could be driven over an obstacle course to reach a “victim”. Once the “victim” was located, the robot had to complete a series of required and optional tasks and return to the location where it started the mission. The detailed information about the scaled-down competition that was provided to students can be found in Table 6 and Table 7 in the Appendix. Figure 3 shows the different ground robot concepts that were proposed by the teams. Note the similarities between most of the robots presented in Figure 3 and the sample robot in Figure 2; it is clear that the process of building the sample robot greatly influenced the teams’ final designs. In fact, some teams simply built on top of their example robot frames. For the fall 2017 implementation students also used a robot to practice programming, but unlike the fall 2016 experience they did not have access to the robot when constructing their prototypes.

The competition was held the day of the final exam for the course. The competition took place indoors since the outside temperature was so low that the robot batteries performed at only a small fraction of their specification. Figure 4 shows a partial view of the obstacle course that was used for the competition. Five out of the seven teams in the class were able to complete the mission.

Figure 3. Ground robot concepts proposed by the student teams.

Figure 4. Partial view of the obstacle course that was used for the competition.

This first implementation in fall 2016 provided much valuable feedback that was used to improve the experience for students in the fall 2017 class. Specifically, the following changes were made between the two implementations:

- Additional content was added to the Arduino kit manual based on student feedback, and the manual was reformatted to improve its visual appeal.
- The kit's introductory presentation was improved based on student feedback.
- Unlike the concentrated introduction provided in fall 2016, the kit was introduced in a more gradual fashion in fall 2017.
- In fall 2017 each team was given a simple pre-built robot to practice their programming skills (Figure 5). These robots <u>were not</u> available for students to use when building their final prototypes.

For the fall 2017 implementation, the class of 25 students was divided into five teams of five students each. Each team came up with multiple project ideas that were presented to the entire class. After all the teams provided their suggestions, the group selected one for the capstone project. General aspects about the mission statement for the product were defined with input from all the teams. However, to provide some flexibility, each team ultimately wrote its own mission statement with the overall goal to design a toy RC car that could carry at least two action figures that were about 4" tall, was all terrain, and could interact with similar toy RC cars. A sample mission statement is provided in Table 2.

.

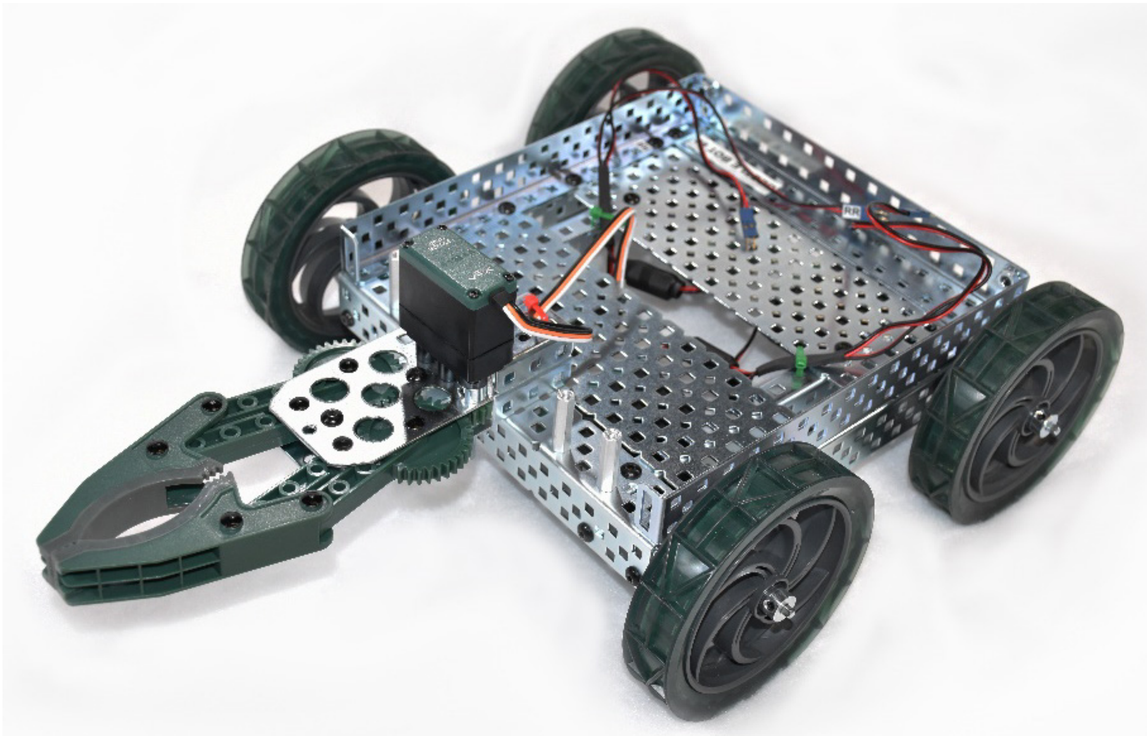

Figure 5. Pre-built robot for students to test programming skills.

Given the variations in the mission statements from the teams, there was a larger variety of final prototypes than were seen in the fall 2016 implementation (Figure 3). Figure 6 shows two student designs from fall 2017. While looking at the designs in Figure 6, it is important to keep in mind that these are only proof-of-concept prototypes that don’t reflect all the details of the final product envisioned by the teams. The left-hand image in Figure 6 shows a concept with a small camera that works with a receiver for Android phones. The students turned the camera on/off using the PS3 controller in the hardware kit. The design also had a projectile launcher (adapted from a toy) that fired a single projectile in the hood. The projectile launcher was controlled using a servo.

The right-hand image in Figure 6 shows a concept that used a track-type design for locomotion. This team ran into some issues with power consumption (they chose motors that required more current than the hardware stack could supply). Like the first team, this design had a projectile launcher (adapted from a toy) that could throw several small rubber balls in rapid succession.

Overall the students seemed to respond better to the fall 2017 project, and they clearly showed greater creativity in their designs than the fall 2016 class. Note that while both projects that have been implemented revolved around ground vehicles, there is no reason that other types of projects cannot be attempted using the hardware kit. In principle any project that uses sensors, actuators, and computation can be attempted provided that the power requirements and pin counts on the hardware stack are respected.

Table 2. Sample mission statement for the RC toy car project.

Product Name	RC Toy Car.
Product Description	Safe and attractive all-terrain remote-controlled toy vehicle that can transport at least two action figures that are approximately 4 inches tall and is capable to interact with the environment and similar toys.
Key Business Goals	• Proof of concept prototype ready by the end of fall 2017 semester. • Keep selling price of toy in the $50 to $100 range. • The cost additional parts needed for the proof of concept prototype should be under $300.
Primary Market	• Male children ages 8 to 14 years old.
Secondary Markets	• Female children ages 8 to 14 years old. • Action figure enthusiasts. • RC car enthusiasts.
Assumptions	• All terrain. • Safe. • Can interact with similar vehicles. • Easy to use.
Constraints	• Remote controlled. • Able to transport at least two action figures that are approximately 4 inches tall. • The resources available on campus to build the proof-of-concept prototype are the machine shop, the wood shop, and 3D printers. • Satisfies applicable government regulations.
Stakeholders	• Children that will play with the toy. • Parents and family members. • Toy stores. • Retail stores.

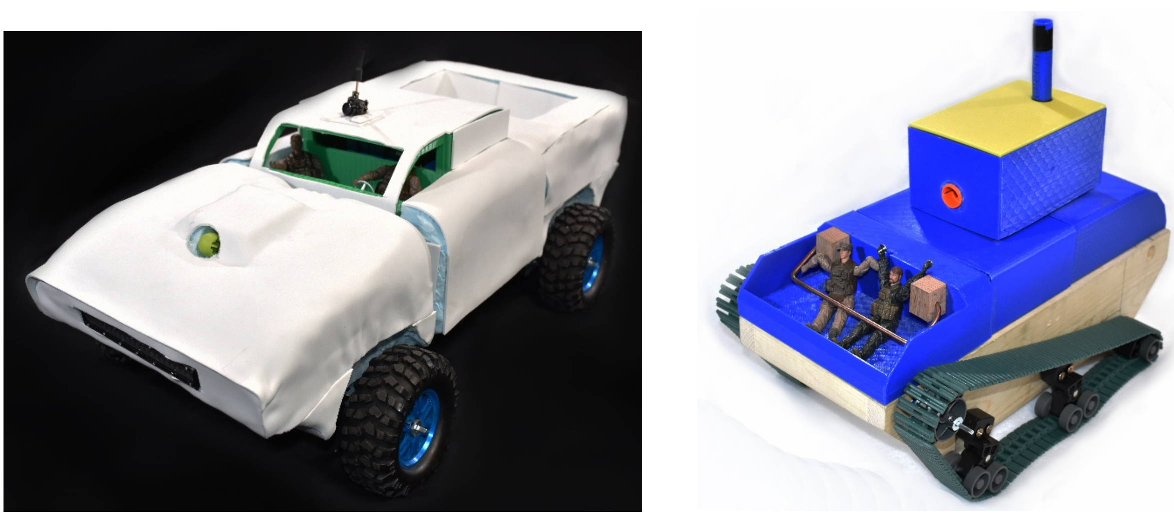

Figure 6. Example student design concepts for fall 2017.

Results and analysis

Students were given a survey at the end of the fall 2017 semester to gauge their attitudes towards the Arduino kit, their self-efficacy in basic mechatronics concepts, and their competency in mechatronics fundamentals. There were 25 students in the class and all of them completed the survey, which was not used as part of their course grade. Overall, the responses to the use of the Arduino kit were very positive, but the conclusions are limited due to the small sample size.

Table 3 gives the results for student satisfaction questions that used a 5-point Likert scale. To summarize the table results, 60% of students reported that using the kit made the class more enjoyable, 92% of students reported that the kit allowed them to consider more complex systems, 84% of students found the provided manual useful for learning to use the kit, 85% of students felt that they received enough instruction to complete the project, 64% of students thought that the kit should be used in future course offerings, and 76% felt that the course prepared them well for a follow-on mechatronics course. The authors were surprised by the difference in results between the questions about instruction and the questions about satisfaction. Students seemed to feel that they had the resources they needed to complete the project, but were less enthusiastic about the use of the kit. This could simply reflect the fact that some students are not interested in the mechatronics side of mechanical engineering; mechatronics instructors at SDSM&T have long seen a similar trend in that class.

Table 3. Student satisfaction results. Overall, the students viewed the Arduino kit favorably.

Question	Strongly Disagree	Somewhat Disagree	Neither Agree Nor Disagree	Somewhat Agree	Strongly Agree
Using the Arduino kit made the class more enjoyable.	1	1	8	9	6
The Arduino kit enabled us to develop more complex systems.	0	0	2	6	17
The Arduino kit manual was useful for learning the kit basics.	0	1	3	13	8
We received sufficient instruction on using the Arduino kit to complete the final project.	0	1	4	9	11
In the future, the class should continue using the Arduino kit.	0	2	7	7	9
Completing this course has made me well-prepared going into the junior-level Mechatronics and Measurement Systems course …	0	0	6	12	7

Self-efficacy results show some similar trends. Table 4 shows the students' confidence in tasks such as reading sensors, driving motors, and writing code on a scale from 1 (Low Confidence) to 5 (High Confidence). For every task, more than 50% of the class rated their confidence as 4 or 5. The lowest confidence scores came from the tasks of using servo motors (52%) and writing code (56%). Given the limited amount of instruction on the tasks rated in Table 4, the authors were pleased with students' self-efficacy. However, the population size is small, and currently no control group was used to isolate the effects of the course from other experiences. Future work will address these concerns.

Table 4. Student confidence results. The 5-point Likert scale range from Low Confidence (1) to High Confidence (5).

Skill	**1**	**2**	**3**	**4**	**5**
Reading analog and digital sensors.	0	1	6	11	7
Driving DC motors.	0	3	3	12	7
Using servo motors.	0	3	9	8	5
Using Arduino shields.	0	2	5	10	8
Writing Arduino code.	0	2	9	8	6

The final portion of the survey asked students a set of brief technical questions to get an objective understanding of students' knowledge. The questions and associated percentages of correct answers are shown in Table 5. All questions allowed for open-ended answers; for questions with a code-based answer, slight variations of correct syntax were accepted as correct answers (e.g. *digitalread(pin4)* was given credit along with the correct *digitalRead(4)*). Most students did well on the first two questions. For the third question, the most popular answer by far was 255 – apparently students confused the analog input range with the analog (PWM) output range. For the last question there were a range of answers. Several students had the correct code but the wrong frequency. If those answers are considered correct the percentage jumps to 36%. The remainder of the answers to this question showed varying degrees of confusion, from giving improper arguments to the digitalWrite function to not changing the state of the pins at all. Anecdotally, one of the authors has asked a similar question on a final exam in the junior mechatronics course and not had substantially better performance.

Table 5. Student skill results.

Question	**% Correct**
What command would you use in an Arduino sketch to read the state of digital pin 4?	68
A PWM value (value sent to a PWM digital pin) of 127 corresponds to what percentage of the maximum output?	80
What is the maximum reading that an analog sensor can return?	8
What commands are needed within the loop() function of a sketch to blink an LED on Pin 4 at 2 cycles per second (i.e., two times per second)?	16

In summary, most students seem to be satisfied with the addition of an Arduino hardware component to the course and report high self-efficacy in mechatronics concepts. While the sample size is small, the results thus far are promising enough to continue the use of the kit. Data will continue to be collected to solidify the current course of action.

Conclusions and future work

This paper has described an Arduino-based hardware platform and instructional strategy for increasing the complexity of sophomore-level design projects. The integration of Arduino into a conventional sophomore design course enables the study of interacting subsystems, and has the additional benefit of bringing continuity in programming education to the curriculum at SDSM&T. Students have generally responded positively to this change, with high levels of satisfaction with the course materials and self-efficacy in basic mechatronics projects. These results are based on an initial survey population of 25 students; the results are suggestive, but certainly not conclusive with such a small sample size.

Future work will improve the confidence in the results through additional polling. It will also collect control data from similar design courses without the programming component to understand the extent to which student self-efficacy results from specific course components versus other experiences. Lastly, outcomes in follow-on classes including a required junior-level mechatronic design class and a senior design class will be used to determine whether the changes in the sophomore classes propagate throughout the curriculum.

Acknowledgements

This work was supported by the Office of Naval Research under Award No. N00014-15-1-2419. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation hereon.

The authors would also like to acknowledge Mr. Aaron Bost, who is an undergraduate mechanical engineering student, for his work on the Arduino kit user manual and the pre-built sample robot shown in Figure 5.

Finally, the authors would like to thank the students that took ME-265/265L Product Design and Development at SDM&T in the fall 2016 and fall 2017 semesters for their useful feedback. The design concepts presented in Figure 3 and Figure 6 were conceived and built by student teams as part of the course project.

Bibliography

[1] Hopkins, M. A., and Kibbe, A. M., 2014, "Open-source hardware in controls education," ASEE Annual Conference, Indianapolis, IN.

[2] Parker, J. M., and Canfield, S. L., 2013, "Work-in-progress: using hardware-based programming experiences to enhance student learning in a senior feedback controls lecture course," ASEE Annual Conference, Atlanta, GA.
[3] Parker, J. M., Canfield, S. L., and Ghafoor, S. K., 2014, "Using hardware-based programming experiences to enhance student learning in a junior-level systems modeling course," ASEE Annual Conference, Indianapolis, IN.
[4] Candelas, F., Garcia, G. J., Puente, S., Pomares, J., Jara, C. A., Pérez, J., Mira, D., and Torres, F., 2015, "Experiences on using Arduino for laboratory experiments of automatic control and robotics," IFAC-PapersOnLine, 48(29), pp. 105-110.
[5] Grover, R., Krishnan, S., Shoup, T., and Khanbaghi, M., 2014, "A competition-based approach for undergraduate mechatronics education using the Arduino platform," Interdisciplinary Engineering Design Education Conference (IEDEC), IEEE, pp. 78-83.
[6] Reguera, P., García, D., Domínguez, M., Prada, M., and Alonso, S., 2015, "A low-cost open source hardware in control education. Case study: Arduino-feedback MS-150," IFAC-PapersOnLine, 48(29), pp. 117-122.
[7] Sarik, J., and Kymissis, I., 2010, "Lab kits using the Arduino prototyping platform," Frontiers in Education Conference (FIE), IEEE.
[8] Sobota, J., Pisl, R., Balda, P., and Schlegel, M., 2013, "Raspberry Pi and Arduino boards in control education," IFAC Proceedings Volumes, 46(17), pp. 7-12.
[9] Ishikawa, M., and Maruta, I., 2010, "Rapid prototyping for control education using Arduino and open-source technologies," IFAC Proceedings Volumes, 42(24), pp. 317-321.
[10] Jamieson, P., 2010, "Arduino for teaching embedded systems. are computer scientists and engineering educators missing the boat?," Proc. FECS, 2010, pp. 289-294.
[11] Hertzog, P. E., and Swart, A. J., 2016, "Arduino—Enabling engineering students to obtain academic success in a design-based module," Global Engineering Education Conference (EDUCON), IEEE, pp. 66-73.
[12] Jamieson, P., and Herdtner, J., 2015, "More missing the boat—Arduino, Raspberry Pi, and small prototyping boards and engineering education needs them," Frontiers in Education Conference (FIE), IEEE, pp. 1-6.
[13] Blikstein, P., 2013, "Digital fabrication and ‘making’ in education: The democratization of invention," FabLabs: Of machines, makers and inventors, 4, pp. 1-21.
[14] Aprile, W. A., and van der Helm, A., 2011, "Interactive technology design at the Delft University of Technology-a course about how to design interactive products," 13th International Conference on Engineering and Product Design Education, London, UK.
[15] Yoon, S.-B., and Jang, E.-Y., 2014, "A development of creative capstone design education," Journal of the Korea Convergence Society, 5(4), pp. 15-20.
[16] Yoon, S.-B., and Jang, E.-Y., 2014, "The application of micro controller board to engineering education for multidisciplinary capstone design," Journal of Digital Convergence, 12(2), pp. 531-537.
[17] Degen, C. M., Huang, S., Ellingsen, M. D., Muci-Kuchler, K. H., Bedillion, M. D., and Ziadat, J., 2017, "Leveraging a newly developed sophomore design course to increase students’ career awareness," ASEE Annual Conference, Columbus, Ohio.
[18] Muci-Kuchler, K. H., Bedillion, M. D., Degen, C., Ellingsen, M., and Huang, S., 2016, "Incorporating basic systems thinking and systems engineering concepts in a sophomore-level product design and development course," ASME International Mechanical Engineering Congress and Exposition, Phoenix, AZ.

[19] Muci-Kuchler, K. H., Bedillion, M. D., Huang, S., Degen, C. M., Ellingsen, M. D., Nikshi, W. M., and Ziadat, J., 2017, "Incorporating basic systems thinking and systems engineering concepts in a mechanical engineering sophomore design course," ASEE Annual Conference, Columbus, Ohio.
[20] Ziadat, J., Ellingsen, M., Muci-Kuchler, K. H., Huang, S., and Degen, C., 2016, "Using practical examples to motivate the study of product development and systems engineering topics," ASME International Mechanical Engineering Congress and Exposition, Phoenix, AZ.
[21] Ulrich, K. T., and Eppinger, S. D., 2016, Product design and development, McGraw-Hill Education, New York, NY.
[22] LinkSprite, 2018, "LinkSprite pcDuino3 Nano," http://www.linksprite.com/linksprite-pcduino3-nano/.

Appendix: Small-scale design competition involving tasks resembling an urban search and rescue (USAR) mission

Table 6 and Table 7 provide the details of the small-scale design competition involving tasks resembling an urban search and rescue (USAR) mission that was used during the fall 2016 implementation.

Table 6. Information about the design competition involving the Arduino kit – Part 1.

Mission Overview	• A team member is selected as the "rescuer". • The "rescuer" will: 1. Take the robot from a table and carry it to the starting point. 2. Place the robot on the ground at the starting point. 3. Perform any assembly or adjustments required by the mission. 4. Remotely operate the robot to perform a series of required tasks and optional tasks in a single round trip to and from the "victim".
Basic (Required) Tasks	• Take a "monitoring device" to the location of the "victim". o Deliver it as close as possible to the "victim" and leave it there. • Record and display the temperature at the location of the "victim". o Measurement must start once the "victim" is reached. o Measurement must be shown on an LCD display. o The LCD display and temperature sensor must be located on the robot. • Provide a clear audio and/or visual signal that the "victim" has been located.
Additional (Optional) Tasks	• Try to wake up a "victim" who is unconscious. • Deliver a "water bottle" to the "victim". • Retrieve a "sample" of the environment near the "victim". • Lay down a rope/line between the "rescuer" and "victim". o The team can choose the type of rope/line. • Illuminate the area in which the "victim" is located. o The illumination device must be located on the robot.

Table 7. Information about the design competition involving the Arduino kit – Part 2.

<table>
<tr>
<td>Competition Details</td>
<td>

The competition will take place on or near the SDSM&T campus.
The competition could be held indoors or outdoors.

If it is held indoors:

The robot will not need to open doors or climb stairs to access the “victim”.

If it is held outdoors:

It won’t involve sand. However, it may involve other challenging terrains, such as rocks.
It won’t take place during rain, hail, or heavy snowfall. However, it may take place in other harsh weather conditions, such as strong wind or cold temperatures.

No hazardous substances will be involved.
The height of any rigid obstacle that the robot must climb will be no more than 1.5in.
Since this is only a small-scale simulation, the following surrogates will be used:

Victim: A 5in.× 5in. × 5in. solid wood cube.
Monitoring device: A 1in. × 1in.× 1in. solid wood cube.
Water bottle: A 1-inch diameter, 3in. long solid wood cylinder.
Environment sample: A 0.5in. × 0.5in. × 0.5in. solid wood cube.

The mission must be completed in less than 20 minutes, or it will be considered unsuccessful.

The mission time will be recorded from the moment the robot leaves the starting point to when it returns, having completed all basic tasks.

</td>
</tr>
<tr>
<td>Materials Allowed (Prototype)</td>
<td>

The Arduino kit provided to each team will serve as the main control unit for the robot.
Students can use any of the hardware provided by the instructor, including the items in the Arduino and VEX kits, to build the robot.
Teams may also fabricate wood or 3D printed parts for their robots.
Each team will have a budget of $100 to purchase additional items for their robot.
Use of any materials not listed above, purchased out-of-pocket or otherwise, is strictly prohibited.

</td>
</tr>
</table>