Project-based smart systems module for early-stage mechanical engineering students

Jennifer Lynne Tennison, Saint Louis University
Dr. Jenna L Gorlewicz, Saint Louis University, Parks College of Eng.

Jenna L. Gorlewicz received her B.S. in mechanical engineering from Southern Illinois University Edwardsville in 2008, before pursuing her PhD in mechanical engineering at Vanderbilt University, where she worked in the Medical and Electromechanical Design (MED) Laboratory. At Vanderbilt, she was a National Science Foundation Fellow and a Vanderbilt Educational Research fellow. As an Assistant Professor in Aerospace and Mechanical Engineering at Saint Louis Unviersity, Dr. Gorlewicz is currently the director of the Collaborative Haptics, Robotics, and Mechatronics (CHROME) Lab. Her research interests are in medical robotics, haptic devices, human-machine interaction, and in creating and evaluating novel learning technologies.

Dr. Sridhar S. Condoor, Saint Louis University

Dr. Sridhar Condoor is a professor, KEEN fellow, a Coleman Fellow, and the editor of the Journal of Engineering Entrepreneurship. He teaches sustainability, product design, and entrepreneurship. His research interests are in the areas of design theory and methodology, technology entrepreneurship, and sustainability. He is spearheading technology-entrepreneurship education at SLU via the Innovation to Product (I2P), iChallenge, entrepreneurship competitions, and funded research. He is the principal investigator for the KEEN Entrepreneurship Program development grants to foster the spirit of innovation in all engineering students. Condoor has authored several books. Titles include Innovative Conceptual Design, Engineering Statics, and Modeling with ProEngineer. He has published several technical papers on topics focused on conceptual design, design principles, cognitive science as applied to design, and design education. VayuWind, a hubless wind turbine for urban environments, is one of his inventions. VayuWind deploys airfoils parallel to the rotational axis in such a way that, unlike other windmills, it rotates around a ring frame, leaving the central portion open for other uses. This enables VayuWind to extract wind power using existing structures such as commercial buildings and skywalks with minimal noise pollution.

Project-based smart systems module for early-stage mechanical engineering students

Abstract
Systems thinking is a key ingredient for an engineering career. In this paper, we present details of a project-based systems thinking module for an early-stage mechanical engineering course. In this module, students learn systems engineering concepts through a series of activity-based lessons, and then apply design and systems-level thinking skills in an integrative, real-world-based project. While mechatronics-based courses and modules are becoming commonplace in engineering curricula, many of these experiences occur later in the educational experience. The uniqueness of the module presented in this work is that it is designed for a first- or second-year engineering course and could be adapted to complement existing design-based courses and expanded to an entire course. Another unique aspect of the module is the way it integrates creative thinking, systems engineering and microcontrollers to engage students in design-build-test real-world projects. The paper presents the rationale for the pedagogy used, the activity modules, and the project implementation details so that other institutions can either replicate or adapt to their needs.

Introduction

The landscape of engineering is continuously evolving. To tackle the rapidly changing needs of the broad engineering field, industries are looking for graduates with an ability to think at a systems level while at the same time possessing the hands-on hardware and software skills necessary to interface at the sub-system level. Extant literature supports that both engineering faculty and students desire hands-on, system-level projects early on in an engineering curriculum (e.g. [1] [2]). Additional literature supports that training engineers in design-based thinking skills is useful in building and motivating core technical skills [3, 4, 5]. As systems become "smarter" and more interdisciplinary, the skills needed to design, implement, and sustain such systems evolve. Yet, this breadth versus depth balance is often difficult to strike within disciplinary curriculums. The rise of mechatronics and robotics programs illustrates this point and supports the industry need for such interdisciplinary skills [1, 2, 3, 4]. The incorporation of mechatronics-based educational experiences into disciplinary curriculums, such as mechanical engineering, also serves as evidence of this need [1, 5, 6, 7].

Addressing this skills gap is challenging traditional, disciplinary courses and curriculums to diversify and include training of these systems-level skills. There are several innovative, mechatronics-style courses and design experiences that have been developed to address this gap [1, 8, 9, 10] (for a thorough sampling of mechatronics education resources, please see [11]). Inevitably, resource and time restrictions, coupled with needing extensive training through pre-requisite courses, limits early exposure to mechatronics-style design projects. Unfortunately, this often delays this important introduction to mechatronics and system design to late in the engineering curriculum, likely coinciding with other courses which would benefit from students having had prior experience of such skills (such as capstone design projects). As a result, there is a growing interest in providing systems-level, mechatronics-like training early on in the engineering educational experience [6, 7], followed with additional opportunities to practice and expand such skills throughout the curriculum.

In this work, we present a five-week project-based smart systems module designed specifically for entry-level (freshman or sophomore) mechanical engineering students with little or no exposure to mechatronics concepts. The experiential module is focused on building system level-design thinking skills with four supporting themes: system architecture, hardware architecture, software architecture, and integration and validation. The purpose of this work is two-fold: 1) to provide an introductory module rooted in design- and systems-level thinking for other researchers and practitioners to expand upon and 2) to lay the foundation within our curriculum of providing systems-level training early on and often through our students' educational experiences.

Module Overview

The module was incorporated into a current second-year Mechanical Engineering course with an emphasis on design principles. Twenty-five mechanical engineering students were enrolled in the course. These students were concurrently taking an introduction to electrical engineering course and had completed an introduction to computer programming course. The class met twice a week for a total of two-and-half hours. Students work in self-selected three-member teams throughout the course and the module was incorporated into the second half of the course. Students learn necessary design thinking skills in the first half and are therefore aware of user research and need identification process by the time this smart systems module is introduced.

The module strategically promotes systems thinking by involving students in

1. Discussions on the basic concepts of systems thinking
2. Connection of core concepts with a diverse set of real-world applications
3. Participation in small-scale, hands-on team-based activities ranging from 20-45 minutes, which occurred in class
4. Completion of a major Arduino-based project that provides an integrative experience, which occurred mostly out of class

The module is rooted in the system design approach, outlined in Fig. 1. The model is a simplified version of the V-model of systems engineering, which is a well-recognized model [12]. Throughout the module, students learn about the four stages in the system design process – system architecture, hardware architecture, software architecture, and integration validation. While the small-scale activities vary in complexity and scope, all activities emphasize one or more of these four steps. Table 1 illustrates various skills students learn during the execution of each activity. Namely, in the system architecture stage, students learn about identifying needs and translating such needs into design requirements and concepts. In the hardware architecture stage, students learn how to select appropriate sub-system components and assemble them in early system prototypes. In the software architecture phase, students learn how to implement system logic and debug supporting software. In the integration and validation stage, students learn how to create test protocols and models of their system to anticipate points of iteration and improvements.

This module was designed for students with no formal system design experience and limited hardware/software skills. To realize the designs in a timely manner, 3 easy to use, free software tools were incorporated into the course: TinkerCAD [13], Arduino [14], and draw.io [15]. TinkerCAD (by Autodesk) was one of the primary software tools used in the module. It enabled teams to simulate their design and debug without wiring any hardware. Students could move through the design, electronics, and coding phases seamlessly. TinkerCAD is easy to use and has readily available tutorials which enable students to create and simulate. Each student team was also given an Arduino Starter Kit (#K000007) for the activities and to design and realize the hardware of their project. Lastly, draw.io was used as a design tool to illustrate the logic behind both hardware and software concepts discussed in the module. In draw.io, students could create schematics, flowcharts, wire frames, and the like to illustrate logic and process in their designs

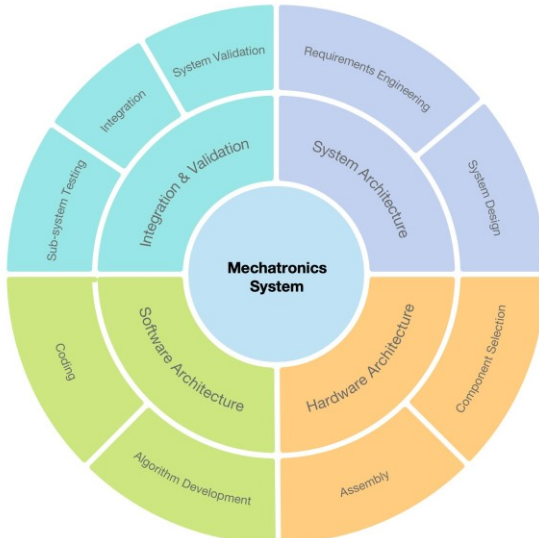


Fig. 1. System design process used in the course

Table 1. Skills learned by students in each stage

Stage	Sub-stage	Skills
System Architecture	Requirements Engineering	1. Identify needs or opportunities 2. Translate needs into requirements
	System Design	3. Propose system solutions
Hardware Architecture	Component Selection	4. Select components
	Assembly	5. Assemble hardware on breadboard
Software Architecture	Algorithm Development	6. Develop flow-charts
	Coding	7. Code and debug
Integration & Validation	Sub-system Testing	8. Create test protocol 9. Create simulation models in TinkerCAD
	Integration	
	System Validation	

Activity Lessons

Short activity lessons were delivered to students for five weeks on topics including algorithms, basic circuits, Arduino basics, communication, digital logic, functions and interrupts, system design, sensors, servos, DC motors, and other topics. Each lesson had a tangible outcome and a team-based activity associated with it. These activities were delivered in-class and each activity roughly spanned nine, one-hour class period (Table 2). Outcomes support the execution of the culminating team-based project.

Table 2. Sub-modules and expected outcomes

	Activity Lesson	Outcomes	Activity
1	System design	Understand the system design process Develop a simple smart system using the system design process	Design an alarm system design with different sensors and modes of operation
2	Algorithms	Understand program control structures Construct algorithms for simple tasks	Develop algorithms for vending machine and coffee maker
3	Basic circuits	Identify basic elements of a circuit Select an appropriate resistor for a given application Design, build and test basic circuits with pushbuttons and LEDs	Create a circuit to control three LEDs using three switches – parallel, series and independent modes
4	Arduino basics	Identify the elements of an Arduino and a sketch Code and execute a basic sketch	Blink internal and external LEDs at desired frequencies
5	Arduino communica-tion	Use serial communication to monitor Arduino and provide command signals	Control LEDs with input from serial port
6	Digital logic	Understand binary system – Representation, Logic, Truth Table Apply Boolean logic to a smart system design	Develop the Boolean logic for a smart agriculture watering system
7	Sensors	Apply different sensors to quantify real-world scenarios	Apply potentiometer to control the blink rate
8	Servo motors	Apply Arduino Servo Library to control a servo motor	Speed control of the servomotor with input from potentiometer or serial monitor
9	Test protocol	Create a test protocol	Create a test protocol for home security system

The first few activity lessons were designed to facilitate planning for and designing a system. This sub-module incorporated lessons on system architecture, how to solicit requirements, and then how to design a platform that addresses those requirements. The later lessons are focused on creating system prototypes for simple tasks and the basics of programming and hardware. Lessons were technical, but approachable for new students. The activities in each lesson allowed teams to create and make mistakes without having to fear broken components or harming themselves. A few activities are illustrated in Table 2.

Projects

The projects were designed to provide students with real-world system design experience that encompassed design principles, prototyping, hardware skills, and presentation experience. Students were placed in groups of three students and were given two weeks to identify a need within a design space, design a solution and prototype it.

Students were provided with design spaces to innovate. The spaces were:

- Care for single/home alone pets
- STEM education for students with disabilities
- Professionals who work remotely
- Smart home improvements for busy people
- Frequent flyers
- Parents who transport multiple kids
- Smart clothing/wearables
- Keeping workers safe in construction zones
- Smart pharmacy solutions
- Transportation of perishable goods

The design spaces were purposefully open-ended to encourage students to think of the "big picture" while creating a mechatronics solution. To help in the successful execution, the project was divided into three phases with explicit deliverables.

After choosing a design space and creating a solution, students were given another two weeks to demonstrate a working, physical copy of their system. Students were graded on 1) project functionality, 2) presentation of the project, and 3) evaluation by 4-5 professors. The project presentations were laid out as 10-minute demos wherein visitors were invited to evaluate the products and the presentation by each group of students. Each group's presentation consisted of the working microcontroller system neatly laid out on poster board with important aspects labelled and a short PowerPoint presentation to show to visitors.

The project as presented in this paper addresses several ABET criteria [16] with an emphasis on Student Outcomes 2 and 3.

Student Outcome 2 is "an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors." In the project, students identify a solution that meets the needs of specific user groups. The student teams identify user needs and create solutions to address these requirements.

Student Outcome 3 is "an ability to communicate effectively with a range of audiences." The evaluation is carried out by a range of audiences from peers, to graduate students, to faculty with varying degrees of familiarity with the project, design, and mechatronics.

Phase I - Need Identification

Students are encouraged to explore a design space or a combination of spaces based on their interests. Then, they conduct research on user habits, competing products, and societal and technological trends. Based on their research, they would have identified unmet user needs or potential opportunities. Then, they select an opportunity and develop design requirements. Rather than starting off with hardware or code, teams first determine a need, then identify the requirements necessary to address the need, and finally devise a plan using flowcharts and system architecture. By concentrating on project development at this stage, teams are not focused on the vehicles in which to obtain the solution, but the practical requirements and implications of the solutions themselves.

Deliverable – A five-minute presentation to the class summarizing the user research, insights, and specific unmet needs and potential opportunities for new products. The system requirements should be specified for product development.

Phase II - System Design Phase
Students propose a system design. The system design includes both hardware and software. They should specify the components (sensors, actuators, peripherals) so that they can be acquired in a timely manner if not available in the Arduino Starter Kit. Also, as the goal is to design a scaled model the system to demonstrate the functioning, the size of actuators is not of major concern. They develop a detailed flowchart using a computer tool, such as draw.io.

Deliverable – A five-minute presentation showing a system layout, flowchart, and the bill of material in

Phase III - Prototyping, Testing and Validation
Students are required to model the system using TinkerCAD and then create functional prototypes that demonstrate the concept. In the process, they translate the system sketch into hardware and flowchart into a workable code. Students are encouraged to work on one sub-system at a time. They are required to develop a test protocol and validate the system. The overarching goal is for students to address significant problems with the design of their system in the simulation model.

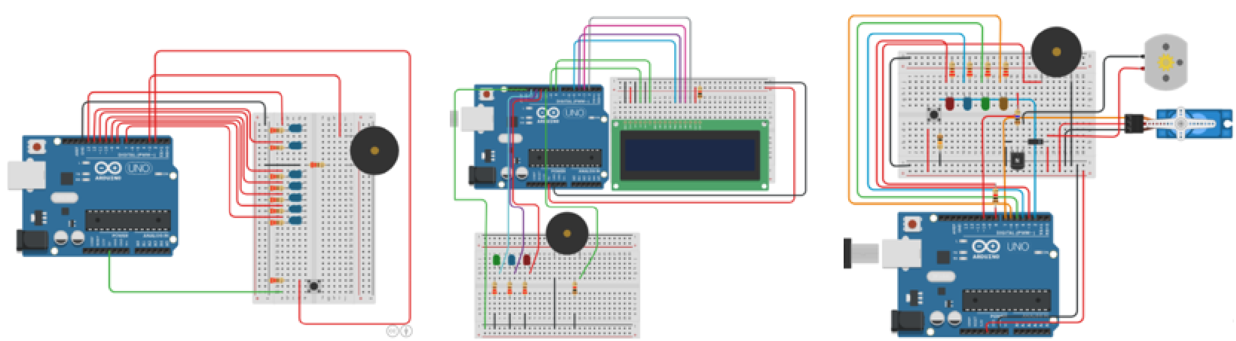
Deliverable - Demonstration of the projects to diverse audiences with a poster presentation describing the needs for their system and how they addressed those needs. Teams prepare short demos wherein audience and evaluators rotate every 10 minutes to review the products and provide feedback.

Students were evaluated continuously throughout the project. Evaluating at each of the three project phases helped to identify pain points in the project early. As students came into the module with diverse sets of skills, evaluating student performance at this level helped to account for individual deficits in knowledge or experience in mechatronics. After each phase, student projects and performance were evaluated to be at, above, or below expectations depending on the key content of each phase (see Table 3).

Results and Discussion
The module was implemented in the Spring 2019 semester for the first time. A total of 8 student teams participated in the smart systems module. Because this was the first implementation of this module, formal data has not yet been collected, however we use this paper to share samples of student work generated through the module as well as lessons learned and future improvements that will be made in subsequent semesters.

Table 3. Project Evaluation

		Evaluation Rubric		
		Above Expectation	Meets Expectation	Below Expectation
Phase	I	Expanded upon user needs during design. Exceptionally expressed needs as system requirements.	Identified appropriate user needs during design. Satisfactorily expressed needs as system requirements.	Did not appropriate identify user during design. Did not express needs adequately as system requirements.
	II	Detailed system design and algorithm flowchart for product. Working simulation.	Satisfactory system design and algorithm flowchart for project. Working simulation.	Did not provide a system design or algorithm flowchart OR documentation was not adequate. Incomplete simulation.
	III	Fully functional physical prototype. Code logic with no errors. Exceptionally detailed and engaging presentation.	Fully functional physical prototype. Code logic with minimal errors. Detailed and engaging presentation.	Physical prototype does not function as intended OR at all. Code logic has many errors. Presentation was not appropriately detailed.



(a) (b) (c)

Fig. 2. Examples of student projects designed in TinkerCAD. (a) Medmory; (b) Smart Alarm Clock; and (c) Pet Provider.

Three student projects we highlight in this paper are as follows (see Fig. 2):

- Medmory (a) – An assistive pill dispenser for individuals with memory loss.
- Smart Alarm Clock (b) – A smart alarm clock for busy people has 3 different alarm settings, varying in pitch.

- Pet provider (c) – A device to feed, exercise and entertain pets while owners are away.

The students found the projects to be both exciting and intimidating. They were eager to design their own solutions to problems that were of greatest interest to them. One group of students mentioned that it was the first time that they had been given the freedom to come up with something on their own in their engineering career. A quote from another student participating in this module had said at the conclusion of the project, “[It] really gave us room to stretch our creativity to the limits. Honestly, I loved this course. Everything was so interesting, and I feel like I learned a lot. These types of courses should be emphasized in every engineering program. Really reminded me why I decided to major in engineering in the first place.” Some groups also had multiple project ideas wherein they were tasked with the extra step of reconciling their ideas and compromising on different aspects of their ideas, an important aspect of collaboration and problem solving.

Most of the apprehension and intimidation arose from the coding aspect of the project. All groups except for one had little to no experience with coding and had difficulties imagining how they would realize their projects in Arduino. However, this apprehension dissipated after the first week for most students. In participating in this module’s activities, students quickly picked up on the skills they needed to program the Arduino board. Many remarked after the course concluded that while difficult at first, programming their project was easier than expected. A final area of contention was learning to wire the hardware components to the Arduino board correctly. Thankfully, students were able to simulate their project and test their hardware hook-ups in the safety of the TinkerCAD environment. The simulator helped alleviate student concerns when they went to wire the hardware of their physical prototype because they had made all of their mistakes in the simulator. Students were relieved to have this functionality available to them and it allowed them to create more complicated projects than they would have been comfortable creating otherwise. One group opted to use an ultrasonic sensor in the TinkerCAD simulator and learned how it worked, despite not having one in their Arduino Starter Kit. This group followed through with this component and requested one from us to build into their physical prototype.

Conclusion

This paper presents a design and systems-level thinking, project-based learning module for introducing mechatronic systems concepts to early engineering students. We present the contents of the modules, its expected student outcomes, as well as details on the activities and project requirements that support the module. Samples of student work along with observations garnered through implementation are shared. Future work will focus on iterative improvements to the module, a formal assessment of how such a module enhances students’ design thinking and systems engineering skills, and follow-on modules for subsequent courses that expand on these critical skills. Further investigations should also determine what effects of this type of module has on senior-level design and capstone projects.

References

[1] J. Wikander, M. Torngren, and M. Hanson, “The science and education of mechatronicsengineering,” IEEE Robotics & Automation Magazine, vol. 8, no. 2, pp. 20–26, 2001.

[2] M. K. Habib, “Mechatronics engineering the evolution, the needs and the challenges,” inIECON 2006-32nd Annual Conference on IEEE Industrial Electronics. IEEE, 2006, pp. 4510–4515.

[3] M. K. Habib, “Interdisciplinary mechatronics engineering and science: problem-solving, creative-thinking and concurrent design synergy,” International Journal of Mechatronics and Manufacturing Systems, vol. 1, no. 1, pp. 4–22, 2008.

[4] M. Acar and R. M. Parkin, “Engineering education for mechatronics,” IEEE Transactions on Industrial Electronics, vol. 43, no. 1, pp. 106–112, 1996.

[5] M. Grimheden and M. Hanson, “Mechatronics—the evolution of an academic discipline in engineering education,” Mechatronics, vol. 15, no. 2, pp. 179–192, 2005.

[6] J. Vaughan, J. Fortgang, W. Singhose, J. Donnell, and T. Kurfess, “Using mechatronics to teach mechanical design and technical communication,” Mechatronics, vol. 18, no. 4, pp. 179–186, 2008.

[7] S. B. Nokleby and R. Pop-Iliev, “A design challenge-incorporating design into the first year engineering curriculum,” Proceedings of the Canadian Engineering Education Association (CEEA), 2005.

[8] R. G. Allen, “Mechatronics engineering: A critical need for this interdisciplinary approach to engineering education,” in Proceedings of the IJME-INTERTECH Conference on Synergistic Integration of Mechanical Engineering with Electronics (2006). Citeseer, 2006, pp. 1–21.

[9] A. Sripakagorn, R. Chancharoen, K. Maneeratana, and K. Panyajirakul, “An implementation of cdio/design thinking in mechatronics projects,” in 2014 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE). IEEE, 2014, pp. 516–521.

[10] Y. Wang, Y. Yu, N. Xie, C. Xie, and X. Feng, “Exploration of practical-oriented teaching in mechatronics engineering education,” in Frontiers in Computer Education. Springer, 2012, pp. 105–112.

[11] “Mechatronics education.” [Online]. Available: https://www.mechatronicseducation.org/

[12] B. S. Blanchard, W. J. Fabrycky, and W. J. Fabrycky, Systems engineering and analysis. Prentice Hall Englewood Cliffs, NJ, 1990, vol. 4.

[13] Autodesk, “Tinkercad.” [Online]. Available: https://www.tinkercad.com/

[14] “Arduino.” [Online]. Available: https://www.arduino.cc/

[15] “draw.io.” [Online]. Available: https://draw.io/

[16] “Criteria for Accrediting Engineering Programs, 2020-2021.” [Online]. ABET. Available: https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2020-2021/