Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #46416

Infusing Internet of Things into Mechatronics to Train Mechanical Engineering Students for Smart Product Design

Prof. Hakan Gurocak, Washington State University-Vancouver

Prof. Gurocak is the founding director of Professional and Corporate Education (PACE) program at Washington State University Vancouver. His research interests include haptics, robotics and automation.

Dr. Xinghui Zhao, Washington State University

Dr. Xinghui Zhao is the Director of the School of Engineering and Computer Science, and Associate Professor of Computer Science at Washington State University Vancouver. She received her Ph.D. from Department of Computer Science at the University of Saskatchewan in 2012. She previously received an M.Sc. from the same university, and a B.Sc. from Department of Computer Science, Nanjing University. Dr. Zhao's research interests lie in the general areas of parallel and distributed systems, big data computing, cloud computing, and machine learning. Dr. Zhao is a member of IEEE, ACM, ASEE, and IEEE Women in Engineering, and has been actively contributing to the professional community. She served as the general chair for the 15th IEEE/ACM International Conference on Utility and Cloud Computing (UCC2022) and the 9th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT2022). She also served as the local arrangement chair for IEEE CLUSTER 2021. She was the guest editor for Special Issue on Integration of Cloud, IoT and Big Data Analytics, Software: Practice and Experience (Wiley Press). In addition, she has served on the technical program committee for a number of conferences, and as reviewer for various journals.

Dr. Kristin Lesseig, Washington State University

Kristin Lesseig is an Associate Professor of Mathematics Education and Academic Director for the College of Education at Washington State University Vancouver. She earned her PhD at Oregon State University and currently teaches elementary and secondary mathematics content and methods courses in addition to doctoral classes in mathematics and science education research.

Infusing Internet of Things into Mechatronics to Train Mechanical Engineering Students for Smart Product Design

Abstract

Smart products that can connect to the Internet to exchange data are becoming ubiquitous. Mechanical engineers play an increasing role in innovating and designing smart products and manufacturing systems that incorporate the Internet of Things (IoT) technologies. However, the current mechanical engineering curriculum has not kept pace. For the last four years, we have been designing, implementing and revising a new IoT course for mechanical engineering students. In this paper, we present the second revision of the new course based on the experience gained and the assessment data collected in the previous offering. Also, several examples of the smart products designed by student teams are discussed. The course contains active learning and project-based learning components. A smart flowerpot device was integrated into the lectures as an active learning platform. For project management, students are introduced to the Agile method, which is widely used in software development companies and is the leading software engineering methodology for IoT development.

1. Introduction

Physical objects (things), such as thermostats and doorbell cameras, connected to the Internet allow remote network access to these devices creating the so called Internet of Things (IoT) [1, 2]. Significant expansion of IoT has rapidly occurred across consumer products, with various devices now featuring internet connectivity to improve convenience, functionality, and automation. Examples include smart thermostats, lights, locks, watches, refrigerators, doorbells, connected vehicles and fitness equipment. These IoT products are only a small part of the growing list and as technology advances IoT is becoming increasingly integrated into daily life.

Driven by the expanding adoption of IoT technologies across industries, the STEM workforce demand for designing or creating Internet of Things (IoT) products is growing rapidly. Mechanical engineers play an essential role in the innovation, design, integration, and manufacturability of IoT devices. Yet, the current mechanical engineering curriculum at Washington State University Vancouver and elsewhere has not kept pace. A report [3] explored what the typical roles of mechanical engineers, manufacturing engineers and machinists are expected to look like by 2030 in the United States as the fourth industrial revolution, commonly known as Industry 4.0, is changing how products are designed and manufactured through digital transformations of cyberphysical systems. Key findings include technologies such as IoT, artificial intelligence (AI), digital twins and collaborative robots will be most relevant. The report recommends that Academia needs to work with industry to revamp current curricula to align what is being taught with anticipated skills for Industry 4.0. This can be accomplished through more experiential learning such as project-based learning.

Literature search shows limited availability of educational materials on IoT in mechanical engineering such as [4, 5]. Much of what is available tends to focus on just one aspect of IoT rather than providing a comprehensive IoT skill set. For example, courses focus on embedded systems [6–10], hardware [11–13], wireless networking [14], and cloud infrastructure [15]. All of these efforts report very positive student feedback and excitement about the topic. But they target students in electrical engineering, computer engineering or computer science programs and tend to focus on *creation* of the underlying IoT technologies. The focus is on low-level hardware/software concepts (e.g. specific software protocols for sensor interfacing) and not on *integration* of these technologies at a higher software abstraction level as needed by mechanical engineers.

To meet this need, we developed a new modernized mechatronics course with IoT focus and offered it for the first time in Spring 2023. Our overarching goals were to integrate skills from computer science and mechanical engineering, and bridge the gap in the mechanical engineering curriculum.

We are building on prior work by others using active learning [16, 17], PjBL [18–21], agile software development methods [22–24], as well as existing IoT course materials such as [6, 7, 13, 25–28]. Mechanical engineers need to develop smart products and systems for Industry 4.0 through *integration* of the IoT technologies. We kept this important distinction front and center in our curriculum design. Another unique feature is the use of a formal software engineering methodology by mechanical engineering students to develop high quality code.

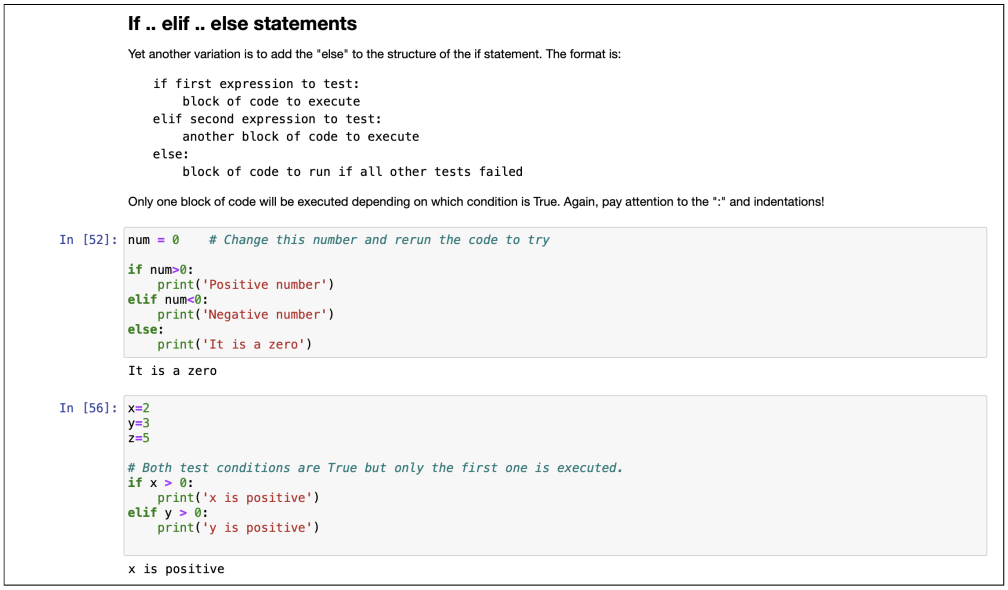
In this paper, we present the design elements of the new course. This is followed by updates made to the curriculum in its second offering in Spring 2024 based on the experience and feedback from the first offering in 2023. Finally, details of the course projects and assessment results are discussed.

2. Design elements of the course

The new course contains several design elements integrated into its modules. The overarching goals in the instructional design were to encourage active learning throughout the lectures and to incorporate a culminating experience in the form of a team project.

Active learning - Active learning increases student success in STEM [16, 17, 29]. In this course, we used *Jupyter notebooks* [30] to implement active learning in Module 1. A Jupyter notebook is a free, open-source, web application that allows students to create and share documents containing live code, equations, visualizations and narrative text. As the instructor explained concepts, code examples were added to the notebook as shown in Figure 1a. Students were typing these examples into their own notebooks along with the instructor and running them. If there were any mistakes, they got immediate feedback from the Jupyter notebook.

In Modules 2-4 (Section 3.2), the active learning approach was continued but this time PowerPoint slides were used for the lectures (Figure 1b). In the slides, there were “Your Turn” sections. Students started from a skeletal Python code file provided to them and completed the code while trying to run it on the flowerpot at their stations. In these modules we used the Thonny Python editor [31] that comes with the Raspberry Pi instead of the Jupyter notebooks. More details about how we integrated active learning can be found in [32].



If .. elif .. else statements

Yet another variation is to add the "else" to the structure of the if statement. The format is:

```
if first expression to test:
    block of code to execute
elif second expression to test:
    another block of code to execute
else:
    block of code to run if all other tests failed
```

Only one block of code will be executed depending on which condition is True. Again, pay attention to the ":" and indentations!

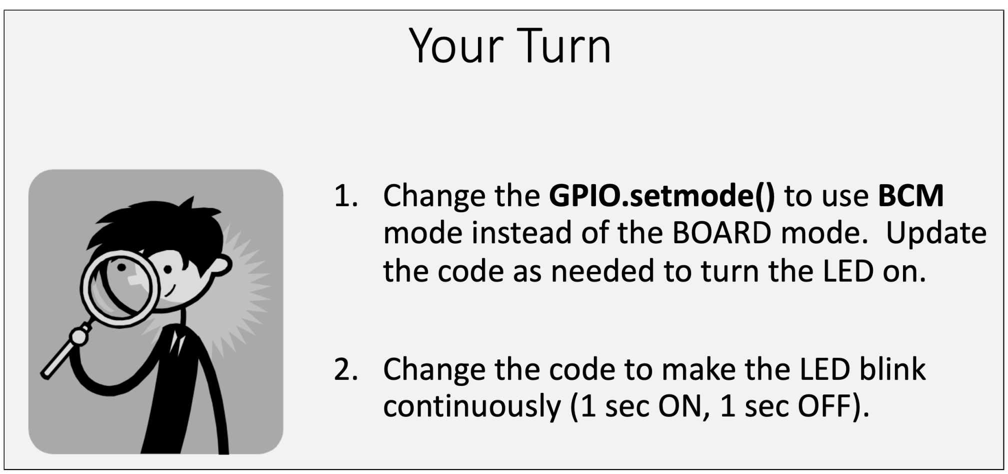
In [52]:
```
num = 0    # Change this number and rerun the code to try

if num>0:
    print('Positive number')
elif num<0:
    print('Negative number')
else:
    print('It is a zero')
```
```
It is a zero
```

In [56]:
```
x=2
y=3
z=5

# Both test conditions are True but only the first one is executed.
if x > 0:
    print('x is positive')
elif y > 0:
    print('y is positive')
```
```
x is positive
```

(a) Python code typed into the Jupyter notebook during the lecture to complete it.



(b) Sample “Your Turn” slide from lecture slides.

Figure 1: Examples for active learning during lectures.

Smart flowerpot - Two students shared a workstation with a smart flowerpot during the lectures as their IoT device. The system consists of a plastic flowerpot on a motorized rotating base platform (Figure 2). The clear plastic bottom section of the pot is a water reservoir with a submersed pump. The white plastic top part is where a plant can be placed. The smart flowerpot contains a light sensor to measure the amount of light the plant receives. It also has sensors to measure the soil moisture, water level in the reservoir and temperature, and humidity sensors for ambient air. All of the electronic components, wiring, and a Raspberry Pi are housed inside the metal pan at the base of the flower pot. Each flowerpot is connected to a monitor, keyboard, and mouse to construct a workstation in the computer lab. The smart flowerpot was custom designed and built. Excluding the Raspberry Pi , the rest of the hardware cost about $200 per pot.

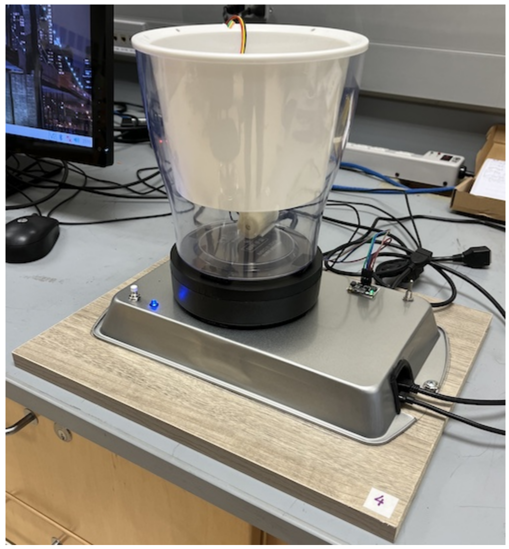


Figure 2: Smart flowerpot. It can connect to a cloud service to retrieve 7-day weather forecast for the location of the pot, take measurement using its sensors and adjust its actions based on the forecast to periodically rotate the plant and deliver just the right amount of water to keep it alive. The flowerpot functions can be monitored over the Internet using a remote dashboard with gauges, digital displays and trend charts.

Agile software engineering methodology - Developing high quality code is challenging, especially for non-Computer Science majors [23]. The mechanical engineering students tend to use an ad hoc approach in code development. The Agile method is systematic and used often by the rapidly growing and volatile Internet software industry for project management [24].

The Agile method was introduced in Module 5. We used Trello [33] as the software platform to implement the Agile method. Fundamental concepts of the method were explained and hands-on demonstrations of how to use Trello were provided to the student teams. Each team is required to use Trello to manage their project, track progress, and collaborate with teammates. Instructor has access to each team's Trello site to monitor their progress [34].

Practice problems - These assignments are in place of the traditional homework assignments in a typical course. Each week practice problems were assigned, but unlike homework, they were not graded. Instead, students were provided with solution files as well as recommendations for how to use the practice problems to enhance their learning and confidence in the material covered. At the end of each course module, a module quiz was administered, which was graded.

Project-based learning to frame the curriculum and instruction - Project-based learning (PjBL) has been shown to be significantly more effective for student learning in engineering education and in mechatronics courses [18–21, 35].

At the end of the semester, student teams are assigned a class project. Each student team can either choose to build a new smart product or use a flowerpot and develop complete control software and remote dashboard for it. Each team submits a proposal to the instructor for feedback. Once a project is approved, parts are ordered by the department staff. At the end, each team submits a report, gives project presentation and demo and returns the prototype device to the department.

3. Second offering of the course in Spring 2024

3.1. Course profile

The mechanical engineering program at WSU Vancouver has a senior-level elective course on microcontrollers. This course is part of a 3-course sequence in the mechatronics option track. It is a 3-credit semester course with two 75-minute lectures per week. The new curriculum was piloted in this course to introduce the IoT curriculum into the mechatronics track. The curriculum contains 10 weeks of instructional material organized into five modules. The last 5 weeks constitute the class project phase where student teams develop smart products they propose.

Course learning outcomes listed below are closely tied to the assessment process used by the department for ABET accreditation. In each course, the learning outcomes are scored on a scale of 1-5 (highest) for each student at the end of the semester. Data collected from each course of the program are then submitted to the department to compile indicators for the attainment of the Mechanical Engineering program level student outcomes.

1. Develop software to meet design requirements
2. Write project reports following format requirements
3. Deliver well-prepared presentations
4. Develop Agile project management board jointly with team members
5. Share responsibilities on project tasks with other members of the team

3.2. Updates in Spring 2024

The new course was offered for the first time in Spring 2023 [32]. Based on the experience gained and student feedback, we updated the module contents and changed the sequence of some of the topics for Spring 2024. There were 24 students (5 from electrical and 19 from mechanical engineering). They were a mix of juniors and seniors.

In Spring 2023, Module 1 "Python programming" was the most difficult for the students. We had anticipated this issue given this was the first exposure to Python for many Mechanical Engineering students. In Spring 2024, we removed some of the advanced topics from this first module and shortened it. Those topics were distributed into the rest of the modules and were presented just-in-time as they were needed. This approach led to much better learning outcomes. As the students went further into the semester, they became more competent with Python programming. When they encountered the more advanced Python topics, they were able to easily understand them since now these programming topics were presented just as they were needed. For example, they learned about Python dictionaries just as they needed to analyze data retrieved from the NOAA servers. Having this additional context made it easier for them to make sense out of the Python tools as they immediately saw how they were used in a real application.

Another change was the sequence of the modules. In Spring 2023, we introduced the hardware/software interfaces module followed by the remote data transmission and processing module. This made it difficult for the students since some of the topics in the hardware/software interfacing also required remote data retrieval. In Spring 2024, we swapped the sequence of modules 2 and 3. This allowed more practice with Python in Module 2 before getting into details regarding hardware in Module 3. Removing the added complexity of hardware/software interfaces made it easier for the students to transition into the entirely new concept of retrieving data from a remote server with Python programming in module 2. When students encounter this skill again in Module 3, they are already comfortable with it and can concentrate on learning the hardware/software interfacing details.

Module 1: Overview of Python - (2 weeks) This is an introductory review of Python programming language. Data types, strings, operators, print statements, if statements, Loops, functions, and modules are reviewed.

Module 2: Data Transmission and Processing - (3 weeks) This module starts with an overview of of cloud computing. Then, programming details on how to retrieve weather forecast data from the National Oceanic and Atmospheric Administration (NOAA) servers are presented. Python Lists and Dictionaries are reviewed just-in-time to help students understand the data format received from the NOAA servers.

Module 3: Data Collection - (2 weeks) This module examines interfacing sensors and actuators to the microcontroller (Raspberry Pi) to explain how a typical mechatronic system is designed. Circuit diagrams are presented for each type of device and code segments are given for hands-on demonstrations.

Module 4: Data Transmission and User Interfaces - (2 weeks) This module starts with an overview of the MQTT protocol for network communications. Then, programming details of how to build a remote user interface with gauges, digital displays, and buttons are presented for real-time display of data transmitted over the Internet from a smart device.

Module 5: Software Engineering - (1 week) This module starts with an overview of the software process models. The Agile software development method is introduced and its pros and cons are analyzed. To help students easily manage their projects, Trello software [33] is introduced as a management tool for the class projects.

Class project - (5 weeks) Students work in small teams and propose a smart product to build as their class project. The project requires using the agile software development method and building a prototype device. At the end, student teams present their project to the class.

3.3. Course project and prototypes delivered

In the last 5 weeks of the semester, student teams met in the classroom during the regular lecture hours and worked on their projects. In Spring 2024, seven teams were formed by the students. Six teams chose to build new IoT devices and one team chose to use a flowerpot as their smart device and developed fully-automated control software to keep the plant alive for a week. The project required each team to develop a functioning IoT device with a remote dashboard for control and monitoring over the Internet. They also had to use agile method in project management and give a presentation at the end.

The team projects were quite diverse and creative. We had a miniature parking garage that could let an approaching driver book a parking spot and showed available or taken spots on a remote phone app and with LED lights in the garage (Figure 3a, 3b). A second project involved a magic mirror that not only functioned as a normal mirror but also showed daily stock quotes and weather forecast. One of the teams built a smart bird feeder which closed the feeder if a squirrel or something of a similar weight landed on it. Users could receive status updates on their phones and alerts if the feeder became empty. Another team built a sun tracker that moved a solar panel to follow the sun throughout the day to maximize the light exposure and energy conversion. One of the teams built a tidal clock. The user could select one of the four locations available on the west coast. The clock retrieved real-time tide information from the NOAA servers and displayed it on a phone (Figure 3c, 3d). It also had a moving tide display and digital display for weather forecast at the selected location. Finally, the most complex project was built to track airplanes in the sky to take photos of them (Figure 3e, 3f). This device retrieved real-time flight information from the ADS-B Exchange servers (just like the popular app FlightAware) and moved a camera to point to a plane passing by to take its photo.

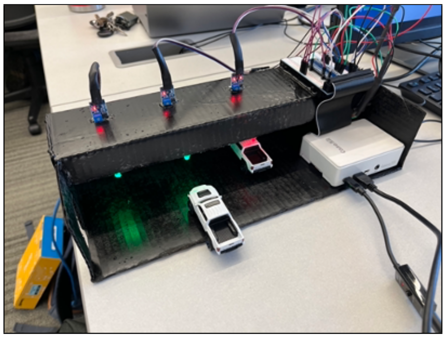
4. Assessment results

After completing each module, students were given a module quiz and a survey to assess the concepts/skills addressed in that module. Each quiz was 1-hour long and required using a computer to demonstrate programming skills. Module 3 required students use the flowerpots to demonstrate their skills with the hardware and software. For module 5, we observed and assessed student accomplishments in the project assignment instead of using a quiz. We analyzed all module survey responses and carefully considered student comments. In the following paragraphs, we present a summary of the evaluations for the Spring 2024 offering of the course.

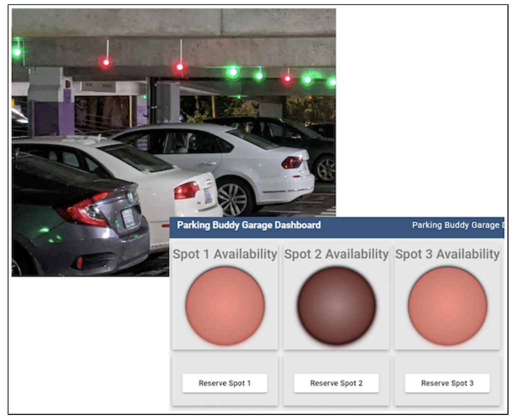
Module 1 is on Python programming. Its quiz contained 9 programming questions that spanned skills from all major topics in the module. Students with the lowest two scores indicated they had issues with debugging their code in two questions. As shown in Figure 4a, overall, the students did very well (median = 27, st. dev. 0.93).

Survey responses indicated that the in-class examples, active learning components and Jupyter notebooks were very successful. Students noted enjoying the ability to learn at their own pace.

Module 2 is on remote retrieval of weather data from National Weather Service (NOAA) servers over the Internet. Its quiz contained 6 programming questions requiring demonstration of skills such as constructing URL queries for the NOAA servers to retrieve data, understanding JSON data files, retrieving individual pieces of information from data files, etc. Students used web browsers and wrote Python programs to accomplish the tasks. As it can be seen in Figure 4b, students did very well (median = 18, st. dev. = 1.2).

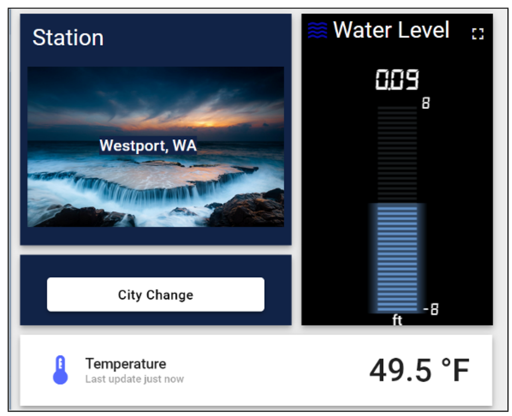


(a) Miniaturized parking garage.

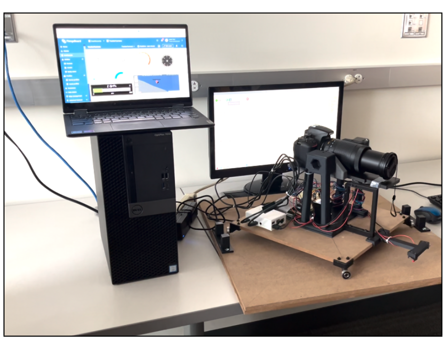


(b) Real-world concept and remote user interface.

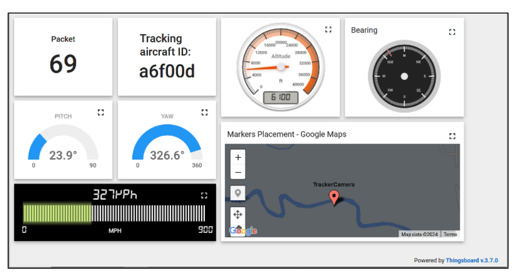
(c) Tidal clock.



(d) Remote dashboard for tidal clock.

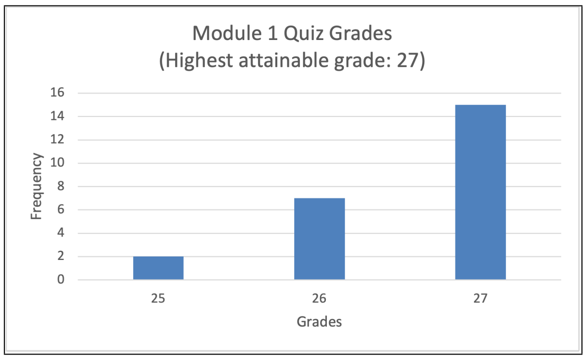


(e) Plane tracker to take a photo.

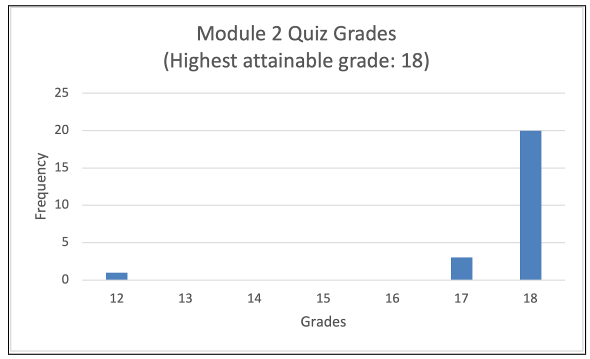


(f) Remote dashboard for the plane tracker.

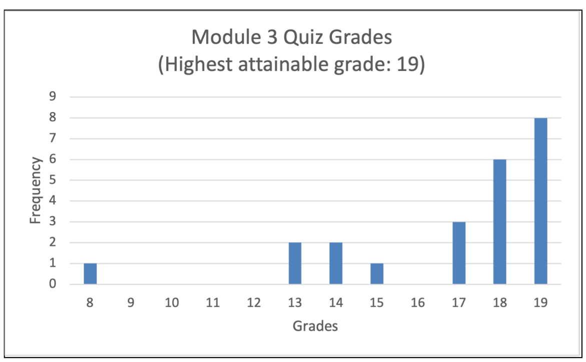
Figure 3: Sample team projects (Spring 2024).



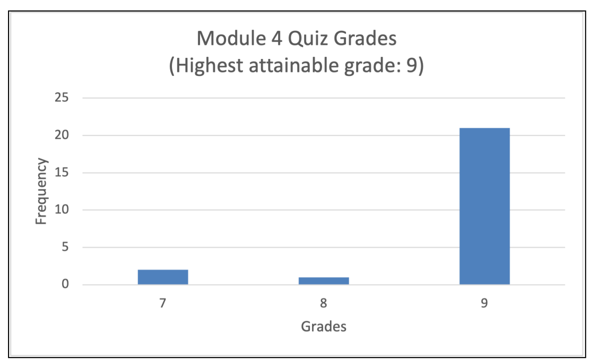
(a) Distribution of Quiz 1 grades. Highest attainable grade was 27.



(b) Distribution of Quiz 2 grades. Highest attainable grade was 18.



(c) Distribution of Quiz 3 grades. Highest attainable grade was 19.



(d) Distribution of Quiz 4 grades. Highest attainable grade was 9.

Figure 4: Quiz grades after completing each module (Spring 2024).

Survey responses indicated similar results with great success in using the active learning exercises, Jupyter notebooks and practice assignments. Several students indicated that just-in-time introduction of the Python dictionaries made the topic less intimidating. Learning the concept and immediately practicing it in the real context of NOAA data was helpful for them.

Module 3 is about hardware and software interfaces for mechatronic devices. It targeted data collection from the sensors of the flowerpot using various programming approaches such as looping and event-driven. It also involved interfacing sensors, motors and controlling them with specific software routines. As shown in Figure 4c, overall students did well in this quiz but there was a bit more spread in the grades (median = 18, st. dev. = 4.4). The programming skills involved in this module were more complex than the previous modules and required application of completely new knowledge to control the physical device.

Survey comments indicated that students really appreciated the slow-paced introduction of each specific topic followed by hands-on practices as the content of this module was more confusing for them, especially the mechanical engineering students. Some of them wanted to have more opportunities to practice with the hardware outside the lecture hours. Due to university safety regulations, giving unsupervised access to the classroom with the hardware has proven to be difficult. Extended availability of the teaching assistant was another challenge.

Module 4 was about remote user interface design and control of the physical device over the internet. Its quiz contained three questions targeting skills such as building a remote dashboard with real-time updates from the flowerpot over the Internet, programming the dashboard and microcontroller for remote procedure calls, etc.

As seen in Figure 4d, majority of the students demonstrated a strong grasp of the concepts through developing working programs and demonstrating them to the instructor during the quiz (median = 9, st. dev. = 0.6).

Once again, students found the practice assignments and in-class activities very helpful. They also thought seeing things happen in the flowerpot over the Internet made the lectures more enjoyable. The technical content of the module involves many steps of setting up things in software to be able to operate the real devices over the Internet. The instructor developed a 14-page handout with all the details, which was much appreciated by the students.

Module 5 was about software engineering, specifically learning/applying the agile method for project management. An updated and more detailed rubric was developed in this offering of the course. Using the new rubric, we assessed the skills/outcomes expected from the project. The rubric completed with scores from the student team projects is shown in Figure 5.

All teams completed the projects successfully and demonstrated working IoT devices they built. Most teams came up with innovative ideas and implemented them except for one team with a more basic design. Some grade points were lost due to missing axis units or incomplete figure captions in reports, and small things missing in the developed software. Overall, there was great excitement in the classroom throughout the 5 weeks of project work and the final presentations were very enjoyable.

		Team 1	2	3	4	5	6	7	total points
Working prototype	**Dashboard**								
	connectivity	5	4	5	5	5	5	5	5
	presentation	5	3	5	5	5	5	5	5
	creativity	5	3	5	3	5	5	5	5
	Device								
	implementation	4	3	3	4	4	4	4	4
	smart behavior	4	3	3	4	4	4	4	4
	demonstration	4	2	4	4	4	4	4	4
	Code								
	basics	4	4	4	4	4	4	4	4
	querying	4	3	4	4	4	4	4	4
	hardware control	5	5	5	5	5	5	5	5
	total:	40	30	38	38	40	40	40	40
Teamwork	consistency	10	10	10	10	10	10	10	10
	sprint details	8	8	10	8	10	8	10	10
	readability	3	3	3	3	5	5	3	5
	total:	21	21	23	21	25	23	23	25
Presentation	***Explain ideas***								
	clarity	2	2	2	2	2	2	2	2
	scope	1	1	1	1	1	1	1	1
	style	1	1	1	1	1	1	1	1
	response	1	1	1	1	1	1	1	1
	Organization								
	logical flow	1	1	1	1	1	1	1	1
	time	1	1	1	1	1	1	1	1
	structure	1	1	1	1	1	1	1	1
	teamwork	1	1	1	1	1	1	1	1
	demonstration	2	1	3	3	3	3	3	3
	Presentation								
	eyes	1	1	1	1	1	1	1	1
	body	1	1	1	1	1	1	1	1
	appearance	0	1	1	1	1	1	1	1
	total:	13	13	15	15	15	15	15	15
Report	***Device design***	2	2	2	2	2	2	2	2
	summary								
	inspiration								
	presentation								
	Smart behavior	2	1	2	1	2	2	2	2
	summary								
	details								
	Remote dashboard	2	0	2	2	2	2	2	2
	summary								
	components								
	presentation								
	Results	2	2	2	2	2	2	2	2
	summary								
	setbacks								
	future improvements								
	Organization/format	1	1	2	2	2	2	2	2
	references								
	professionalism								
	readability								
	total:	9	6	10	9	10	10	10	10
	Team proj. total:	***83***	***70***	***86***	***83***	***90***	***88***	***88***	90

Figure 5: Project rubric completed with scores from the student team projects.

4.1. Course outcomes

Each student's grades from the module quizzes, exams and team projects have been mapped to the five course outcomes given in Section 3.1. The grades are then converted into the scale of 1-5 (highest) to be submitted to the department. Overall, students did very well in achieving the course outcomes (Outcome 1: 4.8/5.0, Outcome 2: 4.6/5.0, Outcome 3: 4.8/ 5.0, Outcome 4: 4.5/5.0, and Outcome 5: 4.5/5.0).

5. Conclusions

In this paper, the second iteration of the curriculum used in Spring 2024 for a new mechanical engineering elective course on Internet of Things (IoT) was presented. The course was offered for the first time in Spring 2023 [32]. It was designed to bridge a gap in STEM education, specifically in mechanical engineering, to better prepare future students for the Industry 4.0 revolution and for smart product design. The new course focuses on the IoT technologies and brings software engineering methods from computer science into mechanical engineering.

A custom designed smart flowerpot was used as a platform throughout the course so students could gain hands-on experience with the IoT technologies. Active learning components were integrated into the lectures. In each lecture, there were multiple "Your Turn" sections where students could try the materials just shown in the lecture using Jupyter notebooks or a compiler on Raspberry Pi and a flowerpot to test their understanding and to try "what-if" scenarios. The course also contained a project where student teams worked on building smart products. The popular agile method used by the tech companies in software development was introduced to the mechanical engineering students to manage their project over the 5-week timeline.

Throughout the course, many students explicitly mentioned how they appreciated the hands-on practice they received throughout the course. The use of Jupyter notebooks, relevant instructor examples, in-class active coding experiences, "Your Turn" practice problems, as well as the class project itself allowed students to immediately apply their learning.

Assessment results are very encouraging. Comments from the four quizzes/surveys revealed that students had a good grasp of the content presented within each of the modules. The students' final projects provided further evidence of the course success. All projects met the requirements and demonstrated an understanding of IoT concepts. Moreover, student projects were much more creative and diverse in 2024 with only 1 out of 7 groups choosing to augment the flowerpot compared to 3 of 5 groups in 2023. In the second offering, we also updated the project rubric making it much more detailed to assess various expectations of the project assignment. In addition, we provided the rubric to the student teams at the beginning of the project assignment to help them better meet the expectations.

Many universities have mechatronics courses, which can be replaced by this new course. Coupled with the inexpensive hardware ($\approx$\$200 per station) and the free open-source software, the transition can be relatively easy. At institutions with large class sizes, scalability can be achieved by holding the lectures in a large computer lab but usually these labs are set up for open access. As a result, the flowerpots may need to be set up before each lecture and taken away after. Another possibility is to hold multiple sections of the course with smaller section sizes.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. DUE-IUSE-2116226. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Autodesk Inc., "The Essentials of IoT for Modern Engineers," https://www.autodesk.com/industry/manufacturing/resources/mechanical-engineer/iot-internet-of-things-essentials-for-engineers, 2016.

[2] W. Mahmoud and N. Zhang, "Disrputive technologies: An educational prespective." 2018 ASEE Mid-Atlantic Section Spring Conference, Washington DC.

[3] A. Huderson, E. Peiffer, S. Shamsi, F. Plazaand, and E. Collins, "Future of mechanical engineering, manufacturing engineering, and machinist roles for industry 4.0." ASEE Annual Conference & Exposition (CIEC), 2022.

[4] Z. Zhang, Y. Chang, S. K. Esche, and A. S. Zhang, "IoT to enable remote collaboration in robotics class of mechanical engineering technology." ASEE Virtual Annual Conference Content Access, 2021.

[5] J. Shen, D. Jones, K. Imran, X. Wang, W. Chen, and L. Mee, "Integrating IoT in mechatronics lab for mechanical engineering technology curriculum: Embracing industry 4.0." ASEE Annual Conference & Exposition (CIEC), 2024.

[6] J. He, D. C.-T. Lo, Y. Xie, and J. Lartigue, "Integrating Internet of Things (IoT) into STEM undergraduate education: Case study of a modern technology infused courseware for embedded system course." IEEE Frontiers in Education Conference (FIE), 2016.

[7] D. V. Gadre, R. S. Gaonkar, S. N. Ved, and N. Prasannakumar, "Embedded systems and Internet of Things (IoTs) - challenges in teaching the ARM controller in the classroom." ASEE Annual Conference and Exposition, 2017.

[8] J. O. Hamblen and G. M. E. van Bekkum, "An embedded systems laboratory to support rapid prototyping of robotics and the Internet of Things," *IEEE Transactions on Education*, vol. 56, no. 1, 2013.

[9] X. Zhong and Y. Liang, "Raspberry Pi: An effective vehicle in teaching the Internet of Things in computer science and engineering," *Electronics*, vol. 5, no. 56, 2016.

[10] V. Galluzzi, C. A. Berry, and Y. Shibberu, "A multidisciplinary pilot course on the Internet of Things: Curriculum development using lean startup principles." ASEE Annual Conference and Exposition, 2017.

[11] L. O. Kehinde, O. T. Ayodele, O. O. Akintade, and K. O. Olawale, “Development of a module to teach basic concepts of interfacing and connectivity in Internet of Things.” ASEE Annual Conference and Exposition, 2016.

[12] S. Abraham and A. Miguel, “Creation of an Internet of Things (IoT)-based innovation lab.” ASEE Annual Conference and Exposition, 2017.

[13] P. Bisták, R. Moezzi, F. Semeraro, and G. Nicassio, “Teaching IoT Using Raspberry Pi Based RC-Car,” 2020, pp. 1–6.

[14] S. J. Dickerson, “Preparing undergraduate engineering students for the Internet of Things.” ASEE Annual Conference and Exposition, 2016.

[15] D. Guo and A. Koufakou, “How cloud computing is addressed for software development in computer science education.” Human-Computer Interaction. User Interface Design, Development and Multimodality, 2017.

[16] P. Seeling, “Active learning moves programming students from novice to skilled,” https://www.pearsoned.com/active-learning-programming-skilled/, 2016.

[17] O. Mironova, I. Amitan, J. Vilipõld, and M. Saar, “Active learning methods in programming for non-IT students.” International Conference e-Learning, 2016.

[18] N. L. Toner and G. B. King, “Restructuring an undergraduate mechatronic systems curriculum around the flipped classroom, projects, LabVIEW, and the myRIO.” Boston: American Control Conference (ACC), July 6-8, 2016.

[19] S. Chandrasekaran and J. M. Long and M. A. Joordens, “Evaluation of student learning outcomes in fourth year engineering mechatronics through design based learning curriculum.” IEEE Frontiers in Education Conference (FIE), 2015.

[20] J. Mynderse and J. Shelton, “Assessment of an improved problem-based learning implementation in a senior/graduate mechatronic design course.” ASEE Annual Conference and Exposition, 2015.

[21] K. Meah, “First-time experience of teaching a project-based mechatronics course.” ASEE Annual Conference and Exposition, 2016.

[22] R. C. Martin, *Agile Software Development: Principles, Patterns, and Practices*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2003.

[23] M. Guzdial and A. Forte, “Design Process for a Non-majors Computing Course,” in *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE ’05, 2005, pp. 361–365.

[24] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, “Agile Software Development Methods: Review and Analysis,” *CoRR*, vol. abs/1709.08439, 2017.

[25] C. Rennick, C. Hulls, D. Wright, A. J. B. Milne, E. Li, and S. Bedi, “Engineering design days: Engaging students with authentic problem-solving in an academic hackathon.” ASEE Annual Conference and Exposition, 2018.

[26] Y. Zhao, S. Huang, C. Elks, U. Ozgur, and V. Avrutin, "New course development for internet of things," in *ASEE Annual Conference & Exposition*, 2023.

[27] L. McLauchlan, D. Hicks, and M. Mehrubeoglu, "Board 372: Remote engaged student learning through hands-on internet of things," in *ASEE Annual Conference & Exposition*, 2023.

[28] J. H. Kleinschmidt, "Teaching internet of things for engineering courses: a project-based cooperative approach," *The International Journal of Electrical Engineering & Education*, vol. 58, no. 4, pp. 858–873, 2021.

[29] C. Martínez and M. Muñoz, "ADPT: An active learning method for a programming lab course." Proceedings of the 10th International CDIO Conference, Universitat Politècnica de Catalunya, Barcelona, Spain,, 2014.

[30] Jupyter.org, "Jupyter Notebook," https://jupyter.org/index.html, June 2019.

[31] Thonny.org, "Thonny Python IDE," https://thonny.org, Jan 2024.

[32] H. Gurocak, X. Zhao, and K. Lesseig, "Preparing mechanical engineering students for industry 4.0: an internet of things course." ASEE Annual Conference and Exposition, 2024.

[33] Trello.org, "Trello Software for Agile Method," https://trello.com, Jan 2024.

[34] X. Zhao, H. Gurocak, and K. Lesseig, "IUSE: Agile methods coupled with project based learning to train mechanical engineers in the era of industry 4.0." ASEE Annual Conference and Exposition, 2025.

[35] J. Larmer and J. Mergendoller, "The main course, not dessert," Buck Institute (www.bie.org), 2011.