Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #45402

From Circuits to Cloud: An Experience Report of a Full Stack IoT Course Curriculum

Mr. Jean Louis, University of Florida

Jean is currently pursuing PhD in Computer Science from the University of Florida. Jean also holds a master's in computer science and a bachelor's in computer engineering from the University of Florida. His research has involved the Internet of Things, Machine learning and IoT education. His current focus is on the impact of running machine learning algorithms on low-resource devices. His work lies at the intersection of embedded devices and machine learning.

Ms. Nadia Simone Jean Morrow, University of Florida

I'm Nadia Morrow, a PhD student in Human-Centered Computing at the University of Florida. My research focuses on the intersection of race, disability, and computer science education, aiming to reduce barriers in CS education and foster inclusive teaching practices. With a background in computer science and a commitment to social impact, I aim to create more inclusive educational environments and increase diversity in technology development.

Dr. Juan E Gilbert, University of Florida

Juan E. Gilbert is the Banks Family Preeminence Endowed Professor, UF Distinguished Professor & Chairman of the Computer & Information Science & Engineering Department in the Herbert Wertheim College of Engineering at the University of Florida where he leads the Computing for Social Good Lab.

From Circuits to Cloud: An Experience Report of a Full Stack IoT Course Curriculum

Abstract

Intelligent and connected devices are growing at an unprecedented rate. These technologies range from smart assistants to low-resource health or environmental monitoring systems. These devices interact with multiple levels of the technological stack, but often students do not have the breadth of experience these applications require. We present the design and implementation of a full stack Internet of Things (IoT) course for undergraduate computer science students, which addresses this need by integrating multiple technological layers. Unlike some other IoT courses, this course covered several technological layers to build an IoT system, and the students interacted with industry professionals. The students explored trade-offs and challenges at every level of the IoT stack, including devices, communication, web development, security, privacy, and machine learning cloud integration.

The course emphasizes the importance of interlayer interactions and introduces research concepts and connections between AI/ML and IoT. Hands-on experience with simulation and physical devices is also provided, allowing students to combine core competencies from software engineering, computer architecture, networking, and operating systems courses in a meaningful way. Our pedagogical approach and selected assignments are presented, demonstrating tools to scaffold CS students to develop hardware-based skills and create an experiential learning environment. Positive student learning outcomes were observed, and a retrospective survey quantifies knowledge gained in full stack development. Recommendations are provided for future IoT courses, highlighting the value of experiential learning and interdisciplinary integration to develop well-rounded computer science professionals.

1 Introduction

Internet of Things (IoT) has continued to gain importance in many industries. IoT applications are present in many vital sectors such as government, healthcare, energy, transportation, manufacturing, and retail [1]. IoT allows these industries to collect and integrate data for analytics and artificial intelligence (AI) to help increase productivity [2]. As a result of the needs of these industries, it is projected that the market size of IoT will continue to grow in the next several years [3–6]. Fortune Business Insights predicts that the market size of IoT will grow from" $714.48 billion in 2024 to [about $4.062 trillion] by 2032" [3]. With its importance in several sectors and continued growth in market size, many jobs in many fields are available for graduates with IoT skills [2, 7]. While ACM/IEEE has provided guidance on incorporating IoT into information technology and computer engineering courses, ACM/IEEE-CS/AAAI computer science curricula for 2023 do not include IoT as a knowledge unit or any guidance on including IoT in a computer science curriculum [8–11]. In addition to the lack of formal institutional recognition, many degrees do not have a comprehensive IoT course to allow students to gain the necessary skills to be successful in these jobs [12]. A diverse set of skills is needed to build large-scale IoT systems; therefore, a comprehensive IoT course could provide students with the opportunity to reinforce learning from other classes and gain the relevant skills for other potential jobs. For example, one layer of IoT includes building interfaces to communicate and operate what is happening with the device, which helps to build students' web development skills. Because of this diversity, students will also gain experience with important skills for jobs such as user interface, network security, and systems engineering [2]. However, when an IoT course is presented, it often focuses more on one layer of the IoT stack than on the whole picture, despite the benefits of learning the skills from end-to-end system [12]. Understanding the parts and connections of an end-to-end IoT system provides a deeper understanding of the tech stack and exposes students to more marketable job skills. Due to these gaps, we provide an experience report of conducting a comprehensive IoT course at a large R1 institution that contributes the following to the body of research.

- We present a Full-Stack IoT course that covers a breadth of topics to prepare students to build and deploy full end-to-end IoT systems.
- We describe the collaboration with industry guest lectures to both introduce students to career paths and the latest of professional engineering tools that could be used as part of their class project.
- We discuss both the successes and shortcomings of this class style.
- We provide assignment ideas and resources for engaging students with hands-on IoT assignments.

1.1 Related Work

Abichandani et al. surveyed papers that provided an IoT curriculum in one of the following IoT layers: sensing, network, service, interface (four-layer model) [12]. Most courses taught in grades K-12 were based on the sensing layer, allowing students to interact with environmental and proximity sensors [12]. While the undergrad-focused papers reported a greater variety of layers, none had all four layers mentioned above. Abichandani et al. identified the lack of classes covering all four layers and suggested that industry professionals be included in the curriculum development process [12]. The Burd et al. survey reported on proposals to develop IoT into curricula and the ACM and IEEE guidelines on a full recommendation of IoT degree programs [8]. These proposals range from a single course that covers all four layers of IoT to modifying a current degree program to include more IoT competencies [8]. Most papers proposed potential courses or degree changes but did not report implementation results [13–15]. Laird describes how Steven's University incorporated IoT and Cyber-Physical Systems into their Software Engineering and Computer Engineering degrees but not for computer science [13]. Chunzhi et al. report on how they intend to transition from a network engineering major to an IoT major but did not have information on a single comprehensive class [14]. Olagunju and Khan present a curriculum guide for incorporating IoT into existing programs and developing programs but do not provide their experience in implementing these guidelines [15]. One paper reported on their IoT course and the student's learning gains [16]. Dickerson reported providing a comprehensive IoT course that did not include the participation of industry professionals and only used direct assessment to identify success in students [16]. The papers presented either miss the opportunity to develop an IoT course that covers all four layers, include industry professionals in the curriculum, implement their proposals, or include indirect assessments. The Full Stack IoT course we present addresses and builds on these limitations and recommendations.

2 Course Curriculum

The course description provided to students:

> The Internet of Things (IoT) is driving change in our society in many areas, such as healthcare, agriculture, environmental monitoring, and natural resource management. This course will introduce students to the concepts involved in creating an end-to-end IoT system. We will explore trade-offs and challenges at every level of the IoT stack, including devices, communication, web development, and ML Cloud integration. Students will gain hands-on experience with simulation and physical devices.

The hardware requirements included an ESP32 kit with basic circuit components such as resistors, push buttons, sensors, LEDs [light-emitting diodes], and a breadboard. We also describe using different radio communication modules to communicate between devices. We did not have a required textbook, but we recommended a book for IoT methodology[1]. The main software requirements included Tinker-CAD[2], Arduino IoT cloud[3], and Firebase-real-time database[4]. We will describe how these components were integrated into the course. The prerequisite for the course was that students must have taken a Data Structures and Algorithms or equivalent course level course. This was to ensure students had a fundamental knowledge of programming and debugging.

2.0.1 Student Background

In a survey at the beginning of the course, students were asked about their background with certain technologies. At this point in the course, 51 students were enrolled, and all 51 took the survey.

- *Have you worked with Arduino before?*
- *Have you worked with Machine Learning?*
- *Have you worked with cloud computing?*
- *Have you taken a course in networking before?*
- *Are you interested in pursuing a graduate degree?*

[1] Internet of Things (A Hands-on-Approach)- by Arshdeep Bahga, Vijay Madisetti
[2] https://www.tinkercad.com/circuits
[3] https://cloud.arduino.cc/
[4] https://firebase.google.com/

The survey showed, 45% of the students had experience working with Arduino, while 55% have no prior experience. This is followed by relatively low percentages in Machine Learning (37%), cloud computing (27%), and networking (35%). Lastly, 55% of the students were interested in pursuing a graduate degree.

By the end of the term, the class consisted of 49 students. Based on the exit survey, the student demographic consisted of 17 Asian, two Black or African American, and 30 white (Hispanic-16, non-Hispanic 14). It also consisted of 40 males and 9 Females. Additionally, it consisted of three computer engineering majors and 46 computer science majors, of which 34 were seniors.

2.1 Learning Objectives

The course has the following main objectives in relation to Bloom's Taxonomy [17].

- Discuss and engage with research practices-(Understand)
- Employ IoT design methodology concept for a project-(Apply)
- Summarize challenges in developing IoT systems in low resource areas-(Analyze)
- Examine and contrast technology selections at various levels of the IoT stack-(Evaluate level)
- Build an end-to-end IoT solution to a real-world problem-(Create)

The course was offered through the computer science department. The main target audience for this course was senior-level undergraduate students, who were more likely to be enrolled in this special topic course. The student should be familiar with programming and software development. We also anticipate some difficulties students may have. In particular, students may find it challenging to debug both software and hardware if they do not have prior experience. Also, students might not know the best places to get questions related to IoT answered online. To this end, we will describe the scaffolding method used to help students learn to interact with hardware. Scaffolding in CS education involves providing temporary initial support and guidance to students as they learn complex concepts or skills and gradually decreasing support as they mature in the concepts, thus promoting increased autonomy and independence [18].

2.2 Pedagogical Approach

The Maker Movement and Constructionist pedagogy emphasize the importance of play, collaboration, and exploration in learning [19]. The maker-centered learning framework from Clapp et al. [20] provides a valuable structure for incorporating Constructionist pedagogy from Papert [21] and constructivist pedagogy into IoT education. This framework emphasizes the importance of hands-on experiences, experimentation, and collaboration in promoting deep understanding and practical skills [19, 20]. In IoT education, this can manifest as:

Experimentation and tinkering, Allowing students to explore and experiment with IoT devices and technologies in a low-risk environment [19].

Project-based learning and Experiential learning Encouraging students to design, develop, and implement IoT projects that integrate course content in real-world applications. [22]

Collaborative work Fostering a community of learners who can share ideas, expertise, and perspectives on IoT-related projects [20].

2.3 Course Design Principles

The design of this course took inspiration from traditional CS education literature and a mix of k-12 and post-secondary IoT courses as described in the related works [8]. The main goal of this course was for students to understand each layer in the IoT stack and gain insight into design choices at each layer. **Knowledge of the layers and interactions between layers of a complete IoT system will give students experience with a broad range of technologies.** For example, students gained hands-on experience with sensors, embedded programming, networking, and front-end and back-end web development. The course aimed to build on students' existing engineering knowledge by exposing, strengthening, or applying concepts from other courses, such as Ohm's Law from physics, socket programming from operating systems, or machine learning. By doing so, students developed a greater breadth of skills to leverage in the job market, personal projects, or future academic opportunities.
The course was structured around four main components: mini-projects, guest lectures, a semester project, and knowledge quizzes. Each week, students began a new mini-project in class, which provided hands-on experience with key IoT concepts. These mini-projects were designed to reinforce active learning, enabling students to engage with the material through practical exercises. For instance, during the week on circuits and sensors, students built a light-dependent circuit. This approach not only deepened students' understanding of the current week's IoT concept but also introduced skills that would be valuable in their semester projects and future personal IoT endeavors. A further example of a mini-project involved learning to read and write digital and analog values in a microcontroller, demonstrating the practical application of IoT concepts.
This was a project-based class, but a few knowledge quizzes were given to test certain concepts, such as definitions, formulas, or engineering scenarios. Namely, "Given an application, what might be the benefits of a particular communication protocol?"

This course also incorporated a series of guest lectures. Guest lectures, which will be described in further detail later, gave students access to industry professionals and tools (Section 2.5).

The semester-long group project was a team endeavor to design and deploy a fully functional IoT solution to address a real-world problem. Students were given the freedom to choose a project topic and form groups of four. The final project involved integrating multiple sensors, microcontrollers, user interfaces, and server components as per the IoT level 4 requirements specified by Bahgaiot [23].

Throughout the 15-week course, students engaged with course material in a variety of ways. The class met three days a week, each lasting 50 minutes. The first two days consisted of lectures, while the third day was dedicated to guiding students through the mini-projects in class. Pre-videos were used instead of pre-readings to facilitate deeper understanding. These engaging video resources provided insights into the week's topics from diverse perspectives, allowing students to encounter the material multiple times and gain different insights into its application. This multifaceted approach to learning enabled students to develop their skills and apply them in a meaningful way, culminating in the semester-long group project with teams of four.

Course Topics

	Topics
week 1	Introduction
week 2	Circuit and Sensors
week 3	IoT Methodology
week 4	Project Management Software dev
week 5	Intro to Research Sustainable Development
week 6	Microcontrollers
week 7	Communication Protocols
week 8	IoT Privacy & Security
week 9	Web Development 1
week 10	Web Development 2
week 11	Cloud Computing
week 12	Intro to AI/ML 1
week 13	Intro to AI/ML 2
week 14	Case study
	Student Demos

Figure 1: Week to week course topics for the course

2.4 Modules

This course is called Full-Stack IoT Development because, unlike some other courses that may focus on one layer of the IoT stack, it covers them all while introducing other development concepts. In section 2.6, we will provide selected assignments and describe guest lectures to increase students' understanding of each module. In this section, we will discuss the course content from the perspective of the four-layer IoT architecture previously mentioned and two additional modules.

2.4.1 Sensing Layer

The sensing layer is a critical component of an IoT system. It collects data from the physical world and converts it into a usable format. This involves converting physical phenomena into changes in voltage or current that the system can read. For instance, a flex sensor changes its resistance in response to flexing; reading this change can then trigger a signal to stop or start a motor. In this course, students learned the fundamental principles of circuits and sensors, including Ohm's law and resistor, voltage, and current calculations. We also introduced them to various sensors and actuators, such as potentiometers, temperature sensors, motors, and LEDs. To illustrate the practical application of these concepts, students completed exercises like blinking an LED, which serves as the "Hello World" of IoT. Students can build upon this basic understanding in future projects, where they can use it to trigger web requests or control physical devices like water pumps. Additionally, students explored the pros and cons of popular MCU devices, such as Arduino, Raspberry Pi, and ESP32, and determined their suitability for use with sensors.

2.4.2 Network Layer

The network layer of an IoT system involves moving data; it can also be considered the communication layer. There are many methods with various advantages and disadvantages to move data wired or wirelessly from one location to another. Each method comes with bandwidth, cost, or speed limitations. It is essential to understand these trade-offs when creating an IoT system. In this course, some methods discussed included serial communication, Bluetooth, wifi, and LoRa. In the assignment section, we discuss an assignment to introduce students to the trade-offs of these technologies with hands-on exercises.

2.4.3 Service Layer

The service layer of the IoT stack can be categorized as using a cloud service to store, compute, forward, or visualize data. In this course, students were introduced to essential software engineering concepts related to system design. Although the class projects would not need to scale to service 100,000 users, the knowledge of how engineers plan for and accommodate applications at scale helps promote good engineering practices. Oftentimes, IoT classes focus on the cloud to store, forward, or display data only. In future sections of this paper, we mention assignments leveraging the Arduino cloud for an alarm system (Figure 4), a self-hosted NodeJS server interacting with Firebase for a real-time chat application, and work with Azure IoT cloud.

2.4.4 Interface Layer

In IoT systems, users interact with the system through various interfaces, including hardware elements like knobs, sounds, and LEDs, as well as online dashboards. A user might activate a component through a physical or virtual push button, as shown in Figure 4. Similarly, a sound or LED can indicate when a component is active. The course covered physical and digital user interface (UI) concepts to equip students with the skills to design intuitive user interactions. During the team project proposal, students had to consider how users would interact with the data they collected and presented. Students learned about CSS and frameworks like Bootstrap and Material UI during the web development modules (see Figure 1). The first week focused on the front-end, while the second focused on back-end technologies such as Node.js, socket programming, and Firebase. To further connect the course to user experience (UX) research, students were introduced to methodologies like A/B testing and card sorting [24, 25]. This exposure helped broaden the students' skill set, as some students were not familiar with user testing methodologies as part of their regular curriculum.

2.4.5 Cyber Security

IoT devices are often low-resource, making them a ripe target for attacks. These devices often have low computational abilities and cannot implement complex encryption schemes or run antimalware software [26]. IoT devices may be more susceptible to both software and hardware vulnerabilities. Any course teaching IoT should review potential attacks and defenses for IoT systems with students. Work in CS education literature has also started to investigate the concept of "creating defensive programmers" through incorporating security best practices in undergrad courses [27]. Security-conscious engineers are vital to the current workforce. To this end, we discussed different aspects of security and privacy in the class. We discussed hardware hacking, social engineering, and other famous Denial of Service (DoS) attacks. As engineers designing systems, it is vital to understand the trade-offs between security, functionality, and usability. We also asked the students to consider encryption and privacy in their final project design. For example, one class activity was for students to decode a *Caesar Cipher*. Knowledge of common vulnerabilities and defensive coding strategies is essential in any computer science career.

2.4.6 AI and TinyML

Given the rapid advancements in AI, the course incorporated fundamental concepts of Machine Learning (ML) and how IoT devices can leverage its benefits. We introduced students to the burgeoning field of TinyML (also known as embedded AI). TinyML offers significant advantages to IoT systems, including enhanced privacy, reduced costs, and increased customization. While ML at the edge, or on-device, is not entirely new (e.g., Siri running locally on smartphones), its potential in IoT is transformative. For voice or health applications, on-device inference means data does not need to be transmitted to a server for processing, thus preserving user privacy. This also saves bandwidth and battery life [28], as transmitting and receiving are among the most energy-intensive tasks for IoT devices. Local ML models alleviate this burden, mitigate the risk of man-in-the-middle attacks, and enable customization, allowing the model to adapt to individual user needs. While highlighting the benefits of ML, we also addressed its challenges and limitations, such as adversarial attacks, fairness concerns, and the need for explainable AI (XAI). Many students, having interacted with AI technologies like ChatGPT, were already familiar with AI's potential for error. However, they were often unaware of vulnerabilities such as vision models being tricked into misinterpreting a stop sign as a speed limit [29]. Introducing these issues prepares students to design future ML or TinyML applications with considerations for fairness, explainability, and a degree of adversarial robustness. These are vital skills and perspectives in the rapidly expanding AI technology market.

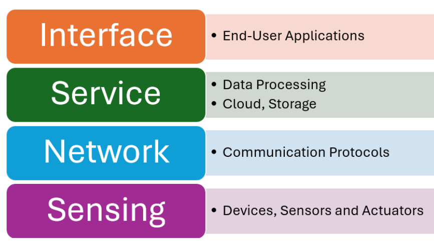


Figure 2: Four-Layer IoT architecture

2.5 Guest Lectures

Several works have highlighted the importance of more industry connections to CS course content [30, 31]. We also see the positive effects of incorporating industry professionals into a CS course [32–35]. In this course, we continue to build on integrating industry professionals as part of the classroom, specifically IoT. To this end, four guest lectures were planned throughout the semester. The four topics covered by the guest professionals included: 1) Project Management, 2) Dev-Opps and Software Engineering Practices, 3) Azure IoT Cloud, and 4) Docker and Containerization.

To introduce students to concepts and tools needed to manage large-scale projects, a guest lecturer who is a project manager (PM) at Microsoft lectured on what it means to be a PM and some tools students could use for their semester projects. Students were able to walk through a case study of running through the steps in a product development life cycle for an IoT project. Student feedback revealed this to be useful and opened their eyes to new career options such as PM and technical project manager (TPM).

The next guest lecture focused on software engineering concepts. Integration and testing are essential to both IoT development and software engineering overall. Continuous integration and continuous delivery (CI/CD) are vital in software development but are not often a standard part of a CS curriculum. Often, class projects are built once and delivered on the due date. To expose the students to CI/CD, a guest lecturer from a startup gave a guest lecture titled "Good software engineering practices and CI/CD." In this talk, students learned about the CI/CD pipeline and were given an opportunity to implement one themselves. Students were able to incorporate some of the practices presented in this lecture into their semester-long group project.

In addition to beginner-targeted tools like Arduino Cloud, we aimed to give students experience with professional IoT cloud environments by having an Azure IoT cloud engineer as a guest lecturer. In this lecture, students learned about IoT cloud integration. In a second lecture, the students then completed a hands-on exercise that connected a simulated Raspberry Pi to the Azure cloud using Message Queue Telemetry Transport (MQTT) protocol and created a dashboard to monitor sensor values. This assignment further contributed to student's experience in *network, service, and interface layer* technology. To further give students experience with modern tools that may not be in the regular curriculum, a Google Cloud engineer delivered a guest lecture on containerization and deployment. In this lecture, students learned how to deploy a full-stack web app using containers.

Industry professional guest lectures added a rich, insightful layer to the course. Our guest lecturers were four alumni currently working at leading companies such as Microsoft and Google, working in areas such as IoT, Cloud, and Project Management. The first two lectures helped the students set up their project with PM and good software engineering practices in mind. These lectures focused on establishing best practices for project management and software engineering, equipping students with a solid foundation for their project work. Subsequent lectures introduced students to more advanced technologies, allowing them to gain hands-on experience and spark interest in emerging trends. However, coordinating these lectures presented several challenges. To ensure a seamless integration of guest lectures into the course, we met with each lecturer in advance to review the content and incorporate relevant skills needed for the class. One key challenge we encountered was designing engaging hands-on activities for students. For instance, a seemingly straightforward assignment on continuous integration and continuous deployment (CI/CD) proved to be a stumbling block for many students, while others found it too simplistic. We also encountered issues with guest lecturers who were unfamiliar with university teaching environments. To address this, we provided additional support, including practice sessions and presentation reviews, to help them adapt to the academic setting. This included reminders to pause and check for understanding and rephrasing technical concepts for the specific context of the IoT course.

2.6 Selected Assignments

In this section, we describe some of the assignments given in the course. We present the objective of the assignment, the task, and relevant outcomes. We employed both direct formative and direct summative approaches to assessing. This included both a survey at the start of the class to gauge students' starting points and an exit survey to understand their self-reported growth in the course. Although there were some written quizzes, this class was a project-based course. One of the early assignments in the course was to introduce students to breadboards and basic circuits; this was under the *sensing layer* of the IoT stack. Students could engage in active and experiential learning in a safe simulation environment as they built their skills. In this course, we leveraged a simulation environment for students to practice and explore. As shown in Figure 3, students learn the dangers of incorrectly wiring an LED. Students were able to see that an LED would burn out if not wired correctly without sacrificing an actual LED from their kit. They also explored the effects of changing the brightness of an LED as the light-dependent resistor changes resistance. Environments like this are important to help scaffold students up to building physical circuits in larger IoT applications.

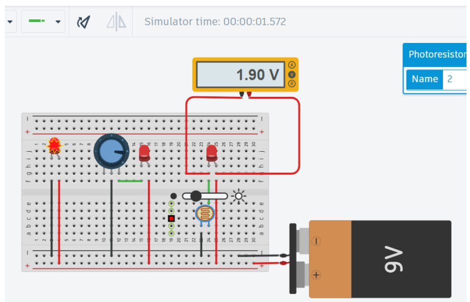
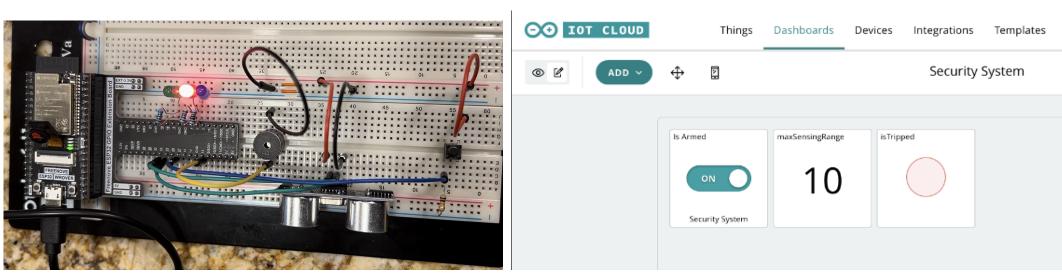


Figure 3: TinkerCad screenshot of early circuits assignments



(a) Proximity sensors based security alarm circuit

(b) Cloud dashboard design for security alarm assignment

Figure 4: Hardware and software interface design for security alarm project. (4a) Contains: LEDs, push-button, Ultrasonic proximity sensor, and buzzer. (4b) Contains: ON/OFF slider, Input box, Boolean colored interface

Another assignment that we would like to highlight also falls under experiential learning and targets the *network/communication layer* of the IoT stack. The objective of this project was to understand the benefits and drawbacks of certain radio communication technologies. This project compared three radio modules across multiple factors: cost, frequency band used, data rate, power usage, and range. These are essential factors when selecting technology for an IoT application. In particular, as the range increases, so does the cost and power usage; similarly, as the data rate increases, so does power usage. Instead of having students only look up these metrics, students were given physical modules, modified code, and went outside to measure transmission distance for these radio modules. The three radio modules tested included: a generic 434MHz RF transmitter and RF receiver, an NRF24L01 Wireless Transceiver Module, and an Adafruit RFM95W LoRa Radio Transceiver.

The students were asked to select a partner; one would hold the transmitter and the other the receiver. They programmed the ESP32 with a radio module. The transmitter sends a signal to a receiver device to blink an LED (Using the RadioHead Library[5]). The receiver student would then start to walk away from the transmitter until the LED was no longer blinking. Then, they noted the distance for this particular radio. They reran the experiment but with a solid object in between them (e.g., a wall). Students were asked to write a write-up with the pros and cons of each, as well as the advantages and disadvantages of the modules, to each other. They were also asked to compare their theoretical distance for line of sight and obstruction to their actual real-world findings.
This class then departs from traditional "full-stack courses" by including an "introduction to the research section." In this section, students learned how to read and categorize research papers. This skill was used as part of a mini literature review assignment that students were required to complete related to their semester project. Students were free to choose their semester project; as a result, the professor would not be an expert in all the areas chosen. The goal was to have students learn more about their chosen field. For instance, one group working on "Multi-point water quality monitoring system" discovered which sensors were most useful in the current literature and became more confident in their design.

Also, 55% of students in the course were interested in graduate school, but many were unfamiliar with types of research publications, using Google Scholar features, or doing a literature search. In this section of the course, we discussed methods for reading research papers, types of experiments, and writing an abstract. All skills that would help improve their final project, **researching new technology while on the job, or becoming a graduate student**.

Lastly, this course featured a semester-long project. The goal was to create a project that solved a real-world issue and used the skills learned from the class at each layer of the IoT stack, for example, circuit schematics, testing plans, Gantt charts, communication protocols, relevant research papers, etc. The project guidelines were presented so that topics from each week would be represented in the final project. Deployed demos were presented at the end of the course. Before the final class presentation, we had a mini-demo day, which allowed us to see students' progress and have them give each other feedback and share ideas.

3 Retrospective Analysis and Student Feedback

3.1 Student Feedback

For indirect assessment, the class completed an end-of-class survey to understand the students' experiences. In this retrospective survey (retrospective post-then-pre questionnaire design), the students were asked about their perceived difference in knowledge from before to after the class on various topics, what improvements they believed the class would benefit from, and what they believed was good about the class. This survey design is meant to allow students to have a clearer reflection of what they gained from an educational experience by having them comment on their previous knowledge after ensuring that they are answering the questions from the same frame of understanding [36]. In the following section, the results of this survey will be discussed and presented. Figure 5 shows the difference between the student's experience with relevant class topics before the class and what they believe their experiences were after the class. Figure 5a

[5] `http://www.airspayce.com/mikem/arduino/RadioHead/`

(a) Before the Class

(b) After the Class

Figure 5: Student's Perceived Understanding of Topics

shows that most students did not have much experience in the related IoT topics and various other important topics in computer science before taking the class. Notably, most students reported no experience with web development, communication protocols, project management, and software development. These topics are important not only for IoT but also for other career options, such as user experience development, cybersecurity, and project management. Additionally, current hot topics such as AI/ML, IoT privacy and security, and IoT methodology were also areas that most students did not understand. Figure 5b shows that after the class, all of the students said they had at least a little experience in all the topics covered. The notable improvement is that in the topics where most students said they had no experience, most of the students indicated having either somewhat or a lot of experience in these transferable skills. The student feedback alludes that this class helps students learn about all dimensions of IoT and also important skills for other careers or further education. Additionally, it alludes that the topics were covered fairly comprehensively due to all the students walking a way with some understanding of all of the topics covered.

The feedback collected from students in the IoT course reveals several key areas that require improvement to enhance the overall learning experience. One of the most notable themes from the data is the need for more structured assignments and explicit guidance on certain topics. Many students expressed frustration with vague assignment descriptions, although some found them straightforward. This is understandable, as over half of the class did not have prior experience with certain technologies, such as the Arduino platform. Some assignments were intentionally left open-ended to encourage students to develop skills like debugging and information finding, which are essential for maker-centered learning and Inquiry-Based Learning [37], promoting critical thinking and problem-solving skills. Future iterations of this course should work on fine-tuning the balance between the open-endedness of problems and the rate at which "scaffolding" support is removed while still promoting exploration and critical thinking. This will help ensure that students maintain confidence and receive appropriate guidance as the topics become more complex. The student feedback also underscores the value of hands-on learning experiences in IoT education. Many students praised the course's emphasis on practical application and experimentation, which helped them develop a deeper understanding of complex concepts and technologies. However, given the breadth of topics covered in the course, managing the workload of hands-on activities is important to avoid overwhelming students. Overall, when students were asked to reflect on what they hoped to learn from the course, 47 students (96%) indicated that the course either met or surpassed their expectations (23 students indicated it surpassed their expectations, and 24 students indicated it met their expectations). Only two students (0.4%) felt the course fell below their expectations. Additionally, when asked about the relevance of the course material to their future work, 23 students (46.94%) found it "Very Relevant," 21 students (42.86%) found it "Slightly Relevant," three students (6.12%) were neutral, two students (4.08%) found it "Slightly Irrelevant," and none found it "Irrelevant." The student feedback shows that careful attention should be given to writing assignments. However, overall, the majority of students found this to be a valuable educational experience and enjoyed the real-world, hands-on activity.

3.2 Challenges

Although students were able to improve or gain skills as part of taking this course, administering this course did come with some challenges. For one, the class had a wide range of hardware skill levels as both computer science and computer engineering students participated. The range in the background sometimes made it difficult to calibrate assignment deadlines. To combat this, an undergraduate peer mentor advised assignment instructions and deadlines to help bridge the expert-novice gap. Although this helped, the assignment calibration might come with time and several sessions of teaching the course. Another challenge that arose for students was co-debugging hardware and software. In the future, we would demonstrate more methods beyond probing and display more in-class hardware/software error debugging. Lastly, some other issues included students missing or having faulty electronic components or WiFi issues during class activity. For this, it is recommended that the instructor has additional components and maintain contact with school IT if WiFi issues prevent activity. To circumvent WiFi issues, some students used hot spots on their phones.

3.3 Recommendations

We recommend using low-cost Arduino-compatible kits like an ESP32 kit. This ESP32 is about \$10 and features WiFi, Bluetooth, low-power operation, and multiple serial communication modes. A camera and SD card could also be added to the device. On the other hand, there is a limited amount of I/O, which falls in line with IoT devices. An ESP32 provides a range of features with responsible constraints. As it relates to TinyML with ESP32, *Edge Impusle*[6], it has many examples and tutorials. Regarding assignments, we recommend "show of worthy" assignments. We advise assignments that introduce students to new technical skills and encourage them to share their creations with others. To illustrate, we could simply have students learn about socket programming and send test messages, or instead we could wrap the work into a chat application, deploy it, and have students demo the app to a friend as part of the submission. Students must connect what they are learning to the real world, both as skills for industry and as tools to create. Students should be able to play, learn skills, and make connections across multiple levels of technological abstraction. Also, as previously mentioned, teachers should *scaffold the playground* as students build these skills. This can be done with simulations, demos, starter codes, extra video resources, and lecture content. This means creating environments where students are empowered to explore more complex ideas beyond the guided activity. We also recommend guest lectures from industry professionals. Students showed positive sentiment, and the lecturers enjoyed giving back to their

[6]https://www.edgeimpulse.com/blog/tag/tiny-ml/

alma mater. This is one method to help further attenuate the industry college gap. Likewise, some students are interested in graduate school but have little knowledge of research. If feasible, students should participate in both internship and undergraduate research as their computer science training. Certain skills, such as locating and synthesizing new work, are important for developers and researchers. We recommend that students be taught skills related to creating a literature review. We believe these recommendations will help foster a great IoT learning environment for students.

1. Leverage low-cost feature-rich IoT kits like the ESP32 kit
2. Scaffold assignments at each level of IoT stack
3. Incorporate "Show of worthy" assignments
4. Leverage guest lectures from industry professionals
5. Incorporate other components such as AI ethics and Research
6. Be flexible with scheduling and build in-class working days

4 Conclusion

IoT is a growing field with many opportunities. In this work, we presented the design and implementation of a full-stack IoT course. We introduced our pedagogical approaches. Unlike some other IoT courses, this course covered all layers of the IoT stack. Students are introduced to a broad range of topics, from circuits to network technology and the cloud. We also featured guest lectures from industry professionals to introduce real-world technology and skills to students. This course emphasizes the importance of the interaction of all layers of the IoT stack. In addition to the traditional 4-layer IoT architecture, we introduce students to research concepts and the interactions between AI/ML and IoT. The students showed positive learning outcomes for the core modules of the course. The recommendations and design choices could be used to develop future Full Stack IoT classes at the collegiate level.

References

[1] [Online]. Available: https://www.oracle.com/internet-of-things/what-is-iot/#why-is-iot-important

[2] G. L. Team, "Iot career opportunities: Ultimate guide 2024," May 2024. [Online]. Available: https://www.mygreatlearning.com/blog/iot-career-opportunities/

[3] [Online]. Available: https://www.fortunebusinessinsights.com/industry-reports/internet-of-things-iot-market-100307

[4] Sep 2022. [Online]. Available: https://beyondmarketinsights.com/report/internet-of-things-market/

[5] [Online]. Available: https://www.statista.com/outlook/tmo/internet-of-things/worldwide

[6] [Online]. Available: https://www.mordorintelligence.com/industry-reports/internet-of-things-iot-market

[7] [Online]. Available: https://www.indeed.com/career-advice/finding-a-job/cloud-computing-careers

[8] B. Burd, L. Barker, F. A. F. Pérez, I. Russell, B. Siever, L. Tudor, M. McCarthy, and I. Pollock, "The internet of things in undergraduate computer and information science education: exploring curricula and pedagogy," in *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 2018, pp. 200–216.

[9] C. Servin, S. Aly, Y. Cheon, E. Eaton, C. Guevara, A. Kumar, T. Pirtle, and M. Scott, "Cs2023: Acm/ieee-cs/aaai computer science curricula-specialized platform development," 2024.

[10] A. N. Kumar, R. K. Raj, S. G. Aly, M. D. Anderson, B. A. Becker, R. L. Blumenthal, E. Eaton, S. L. Epstein, M. Goldweber, P. Jalote, D. Lea, M. Oudshoorn, M. Pias, S. Reiser, C. Servin, R. Simha, T. Winters, and Q. Xiang, *Computer Science Curricula 2023*. New York, NY, USA: Association for Computing Machinery, 2024.

[11] C. T. Force, *Computing Curricula 2020: Paradigms for Global Computing Education*. New York, NY, USA: Association for Computing Machinery, 2020.

[12] P. Abichandani, V. Sivakumar, D. Lobo, C. Iaboni, and P. Shekhar, "Internet-of-things curriculum, pedagogy, and assessment for stem education: A review of literature," *IEEE Access*, vol. 10, pp. 38 351–38 369, 2022.

[13] L. Laird and N. Bowen, "A new software engineering undergraduate program supporting the internet of things (iot) and cyber-physical systems (cps)," in *Proc. ASEE*, 2016, pp. 26–29.

[14] W. Chunzhi, X. Hui, and M. Xia, "Construction of hardware curriculum group for transition from network to internet of things engineering major," in *2012 7th International Conference on Computer Science & Education (ICCSE)*. IEEE, 2012, pp. 1575–1579.

[15] A. O. Olagunju and F. Khan, "Challenges of interdisciplinary iot curriculum," in *Proceedings of the 17th Annual Conference on Information Technology Education*, 2016, pp. 110–110.

[16] S. J. Dickerson, "A comprehensive approach to educating students about the internet-of-things," in *2017 IEEE frontiers in education conference (FIE)*. IEEE, 2017, pp. 1–7.

[17] D. Krathwohl, "A revision bloom's taxonomy: An overview," *Theory into Practice*, 2002.

[18] D. Wood, J. S. Bruner, and G. Ross, "The role of tutoring in problem solving," *Journal of child psychology and psychiatry*, vol. 17, no. 2, pp. 89–100, 1976.

[19] M. Schad and W. M. Jones, "The maker movement and education: A systematic review of the literature," *Journal of Research on Technology in Education*, vol. 52, no. 1, pp. 65–78, 2020.

[20] E. P. Clapp, J. Ross, J. O. Ryan, and S. Tishman, *Maker-centered learning: Empowering young people to shape their worlds*. John Wiley & Sons, 2016.

[21] S. Papert, "Situating constructionism," *Constructionism/Ablex*, 1991.

[22] D. Kokotsaki, V. Menzies, and A. Wiggins, "Project-based learning: A review of the literature," *Improving schools*, vol. 19, no. 3, pp. 267–277, 2016.

[23] V. M. Arshdeep Bahga, *Internet of Things: A Hands-on-Approach*. VPT and Universities Press, 2014. [Online]. Available: https://hands-on-books-series.com/iot.html

[24] M. Speicher, A. Both, and M. Gaedke, "Ensuring web interface quality through usability-based split testing," in *Web Engineering: 14th International Conference, ICWE 2014, Toulouse, France, July 1-4, 2014. Proceedings 14*. Springer, 2014, pp. 93–110.

[25] J. R. Wood and L. E. Wood, "Card sorting: current practices and beyond," *Journal of Usability Studies*, vol. 4, no. 1, pp. 1–6, 2008.

[26] I. Yaqoob, E. Ahmed, M. H. ur Rehman, A. I. A. Ahmed, M. A. Al-garadi, M. Imran, and M. Guizani, "The rise of ransomware and emerging security challenges in the internet of things," *Computer Networks*, vol. 129, pp. 444–458, 2017.

[27] C. Resch, "Creating defensive programmers: Evaluating the impact of adding cybesecurity topics to core computer science courses," in *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 2*, 2023, pp. 87–91.

[28] M. M. H. Shuvo, S. K. Islam, J. Cheng, and B. I. Morshed, "Efficient acceleration of deep learning inference on resource-constrained edge devices: A review," *Proceedings of the IEEE*, vol. 111, no. 1, pp. 42–91, 2022.

[29] F. Khalid, M. A. Hanif, S. Rehman, and M. Shafique, "Security for machine learning-based systems: Attacks and challenges during training and inference," in *2018 International Conference on Frontiers of Information Technology (FIT)*. IEEE, 2018, pp. 327–332.

[30] S. Valstar, S. Krause-Levy, A. Macedo, W. G. Griswold, and L. Porter, "Faculty views on the goals of an undergraduate cs education and the academia-industry gap," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 2020, pp. 577–583.

[31] J. Schilling and R. Klamma, "The difficult bridge between university and industry: a case study in computer science teaching," *Assessment & Evaluation in Higher Education*, vol. 35, no. 4, pp. 367–380, 2010.

[32] L. Zhang, M. Craig, M. Kazakevich, and J. J. Williams, "Experience report: Mini guest lectures in a cs1 course via video conferencing," in *Proceedings of the ACM Conference on Global Computing Education*, 2019, pp. 222–228.

[33] J. Tenenberg, "Industry fellows: bringing professional practice into the classroom," in *Proceedings of the 41st ACM technical symposium on Computer science education*, 2010, pp. 72–76.

[34] L. Gentelli, "Using industry professionals in undergraduate teaching: Effects on student learning," *Journal of University Teaching & Learning Practice*, vol. 12, no. 4, p. 4, 2015.

[35] P. K. Linos, R. Rybarczyk, and N. Partenheimer, "Involving it professionals in scrum student teams: An empirical study on the impact of students' learning," in *2020 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2020, pp. 1–9.

[36] G. A. Davis, "Using a retrospective pre-post questionnaire to determine program impact." 2002.

[37] M. Pedaste, M. Mäeots, L. A. Siiman, T. De Jong, S. A. Van Riesen, E. T. Kamp, C. C. Manoli, Z. C. Zacharia, and E. Tsourlidaki, "Phases of inquiry-based learning: Definitions and the inquiry cycle," *Educational research review*, vol. 14, pp. 47–61, 2015.