2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering
Education for 130 Years
ASEE
Paper ID #39394

New Course Development for Internet of Things

Dr. Yanxiao . Zhao, Virginia Commonwealth University

Dr. Yanxiao Zhao is an Associate Professor in the Electrical and Computer Engineering Department, Virginia Commonwealth University. Dr. Zhao's research interests include, but not limited to: Internet of things (IoT), 5/6G communications, machine learning, cyber security, wireless energy harvesting, power management and communications in smart grid. Dr. Zhao's research has been supported by NSF, NASA, Air Force and Virginia Commonwealth Cyber Initiative (CCI). Dr. Zhao has published over 80 papers in prestigious journals and international conferences. She was the recipient of the Best Paper Award for three international conferences WASA2009, ChinaCom2016 and ICMIC2019. She has been actively organizing international conferences by serving as TPC chairs, publicity chairs and TPC members. She is an IEEE Senior Member.

Dr. Shaobo Huang, University of Saskatchewan

Dr. Shaobo Huang received a Ph.D. degree in Engineering Education from Utah State University. She has over eight years of teaching and/or research experience in engineering education. She is currently an Assistant Professor in the Ron and Jane Graham School of Professional Development with a joint appointment in the Department of Mechanical Engineering at the University of Saskatchewan in Canada. Dr. Huang is one of the key leaders in the RE-ENGINEERED First-Year Program at the University of Saskatchewan.

Dr. Carl Elks, Virginia Commonwealth University

CARL ELKS is an associate professor in the Department of Electrical and Computer Engineering, Virginia Commonwealth University, Richmond, Virginia, USA. His research interests include the design and assessment of safe and secure cyber-physical systems of the type found in critical infrastructure such as nuclear power, aerospace applications, smart cities, and medical devices. HIs recent teaching interests include model based design of embedded systems, and systems thinking for undergraduates. Elks received his Ph.D. in electrical engineering from the University of Virginia in 2005. He is a Member of IEEE. Contact him at crelks@vcu.edu.

Prof. Umit Ozgur, Virginia Commonwealth University

Umit Ozgur received his B.S. degrees in Electrical Engineering and Physics from Bogazici University (Turkey) and PhD degree in Physics from Duke University in 2003. He joined the Department of Electrical and Computer Engineering at VCU in 2008 as an Assis

Dr. Vitaliy Avrutin, Virginia Commonwealth University

Vitaliy Avrutin received his Ph.D. degree in physics and mathematics from Institute of Microelectronics Technology, Russian Academy of Sciences, in 1999. He has held the positions of research scientist at Institute of Microelectronics Technology, Russian Academy of Sciences, Chernogolovka, Russia, and later at the Department of Semiconductor Physics, University of Ulm, Germany. Currently, he is Associate Professor in the Electrical and Computer Engineering Department of Virginia Commonwealth University. His research interests encompass growth, characterization and device applications of semiconductor, nanophotonic, and plasmonic materials. He is a member of the American Physical Society, the Materials Research Society, and IEEE.

New Course Development for Internet of Things

Abstract

IoT is a fast growing technology sector that is estimated to be valued at 100's of billions in US dollars for 2023. The number of IoT connected devices is growing at an annual rate of 20%/year with billions of devices connected. Accordingly, there is much need for foundational IoT engineering courses in our educational institutions to prepare engineers for this technology sector. The challenge for course developers is that IoT technical foundations are exceedingly broad - ranging from smart sensors to low power computing to cloud infrastructure. Most universities focus on one or two aspects of IoT technical foundations, specifically those associated with the computing aspects of IoT. We have developed a novel approach for an IoT course by segmenting the course into three fundamental technology areas. These areas are respectively (1) foundations of sensing, (2) IoT communication and networking, and (3) IoT computing models and architectures. Each module is taught by a different professor with technical expertise in that module. This paper will focus on course objectives, design, and outcomes. We use the Module 2 IoT communication and networking as an example of the type of content. In this module, we cover fundamental knowledge of networking including layered-architecture, protocols, wireless communication and propagation. Then we introduce two common communication protocols, i.e., Hypertext Transfer Protocol (HTTP) and Message Queuing Telemetry Transport (MQTT), which are widely used for IoT systems. Further, Node.js programming is taught and students learned how to use node.js for HTTP and MQTT implementation. Hands-on experiments using Arduino WiFi1010 board and grove sensors are designed to help deepen students' understanding for developing a small-scale IoT system as well. The term project of building an IoT-enabled smart Heating, Ventilation, and Air Conditioning (HVAC) will be introduced. Instructor observations and anecdotal student feedback on the course design and delivery are presented as well. Lessons learned will be discussed and modifications are proposed for future improvement.

Introduction

The Internet of things (IoT) technology connects millions and billions of physical objects with sensors and actuators, processing ability, software and other technologies, to exchange data with each other over the Internet, resulting in economic benefits, efficiency and performance improvement, as well as reduced human efforts (Hassan, 2019). IoT is a rapidly rising area and has been penetrating into virtually every sector of our daily lives including home automation, wearable technology, connected health, vehicle-to-everything communication (V2X), real-time surveillance, smart city and intelligent transportation. The number of IoT devices worldwide has exponentially increased over years. It is forecast to triple from 9.7 billion IoT devices in 2020 to more than 29 billion in 2030 (Elnashar, 2019). According to a report from Mordor Intelligence, the IoT market grew to $761.4 billion by 2020 and is expected to reach $1.39 trillion by 2026 (Job, 2023). Correspondingly, the prosperous market of IoT brings new job opportunities such as IoT engineers, IoT architects, IoT researchers and consultants. Therefore, it is important to train students to gain essential skills and technology relevant to the IoT field and hence to prepare them to enter the IoT-related workforce after their studies.

To meet the needs of delivering IoT related technologies to students, the department of Electrical and Computer Engineering (ECE) in Virginia Commonwealth University (VCU) decided to open a brand new course Introduction to IoT. This course is at 500 level, which targets both senior students and early-stage graduate students. Developing such a new course in IoT is very challenging, because IoT covers a broad spectrum of technologies including sensing, computing, communication, security, etc. Our course attempts to introduce and cover a broad range of fundamental concepts in IoT including a systems approach to realizing IoT, sensing methods and materials, sensor design, communications, wireless networking technologies, edge and cloud computing, and hardware constraints. Students also have opportunity to work on small projects individually or in teams to design and implement small-scale IoT systems and components. However, a single textbook that is tailored for the broad range of topics covered in the course does not exist; thus several books and research papers were adopted as reference to design the lecture material. Since the teaching materials cover a broad range of areas, we divide course content into three modules: Module 1 Devices and Materials, Module 2 Communication and Networking, and Module 3 Computing in IoT. Each module is taught by a different professor specializing in that field. Study shows co-teaching helps highlight the strengths and compensate weaknesses of individual instructors and enables instructors to integrate diverse subject matter knowledge in one course (Vesikivi, 2019).

This paper will use Module 2 as an example to explain how we design each module. Module 2 mainly covers fundamental knowledge for communication and networking such as wireless propagation and traffic model in IoT, some common protocols for different network layers, application protocol design using Node.js. In the fall 2022, we added a new topic, security concerns in IoT systems, considering sensor attacks and cryptography. Hands-on experiment using Arduino WiFi1010, sensors and ThingSpeak platform are as well. We have taught this course for three years at VCU, and have improved the course development based on students' comments and feedback received.

The remainder of the paper is organized as follows. Related work for new course design and the benefits of co-teaching is summarized in Section 2. Section 3 briefly introduces the module-based course architecture design. Section 4 focuses on Module 2 and presents how Module 2 is designed. Section 5 introduces the hands-on experiment design. Section 6 lists some student comments and conclusions are drawn in Section 7.

Related Work

Co-teaching, also called collaborative teaching, is a teaching model that has been widely used in various professional training such as teacher preparation and medical programs (Steele, Cook, & Ok, 2021; Dehnad, Jalali, & Shahabi, 2021). It allows instructors to collaborate in integrating diverse course content, delivering a wide spectrum of content, creating an inclusive learning environment for diverse student groups, providing comprehensive feedback, and supporting each other in pedagogical practices (Vesikivi, 2019; Bacharach, & Heck, 2007; Angela, 2006; Walther-Thomas, 1996).

However, the effectiveness of co-teaching is determined by the levels of coordination and collaboration between instructors. It is not a surprise to notice minor or even major discrepancies in expectations held by different instructors in a co-taught course. To optimize teaching and learning experience in a co-taught course, Felder (1996) suggested constantly adjusting schedules, content, pedagogy, and assessments.

Module-based course design smooths the coordination between instructors and promotes course breadth (Ferguson etc. 2013; French, 2015). In modular curriculum, a subject is deconstructed into learning objectives/topics and then they are clustered by modules. Logically ordered modules scaffold knowledge and skills development in a course. The module-based course design also forms transparent milestones of each module and helps diminish redundant teaching activities and learning gaps. Assessments are not arranged as midterm/final exams; instead, they are placed by modules. This feature enables timely feedback for both students and instructors (Hodgson, 1997).

Course Design

In terms of course structure design, we adopt a module-based course design, i.e., dividing all teaching content into three modules: Module 1 Devices and Materials, Module 2 Communication and Networking, and Module 3 Computing in IoT. Table 1 summaries the teaching materials for each module, which is taught by a different professor with research and teaching experiences in that field. These three modules are interconnected through course practices, both in theory and experiments. These modules will equip students with fundamental and broad knowledge to eventually develop an IoT-enabled systems. In Module 1, we cover sensors and the importance of understanding the basic principles of sensors for enabling the IoT. In Module 2, we teach how IoT architectures provide the framework to transform raw data into insights and actionable knowledge, as well as how to send sensor data via networks (mainly wireless) and the protocols of networks facilitates turning data into information. In Module 3, we cover computing, various wireless technologies, etc.

Table 1. Module-based Course Design

Module Name	Covered Content
Module 1: Devices and Materials	1. The IoT model – focus on sensing the real world 2. A primer to electromagnetic fields and waves 3. A brief review of electronic materials 4. Sensing fundamentals and devices 5. Energy harvesting / power management and storage
Module 2: Communication and Networking	1. The IoT Model – focus on the communication and data characteristics 2. Communication and networking 3. Wireless propagation and traffic model in IoT 4. Application protocols using Node.js (e.g., HTTP, MQTT) 5. IoT Cyber security
Module 3: Computing in IoT	1. The IoT model – focus on the computational aspects 2. Cloud, edge computing and fog computing 3. Sensor characteristics, power concerns, etc. 4. Wireless networks and technology (e.g., IEEE 802.11, Zigbee, Cellular networks)

Content Design for Module 2

This section will mainly focus on the design of Module 2: Communication and Networking. Students enrolled in this course are from different majors such as Electrical Engineering (EE), Computer Engineering (CPE) or Computer Science (CS), at both undergraduate and graduate levels. Therefore, it is necessary to cover some fundamental knowledge of networking including layered-architecture, protocols, wireless communication and propagation. When we review these fundamentals, IoT-related applications are introduced such as the typical traffic model for IoT systems.

Then we introduce two common communication protocols, i.e., Hypertext Transfer Protocol (HTTP) and Message Queuing Telemetry Transport (MQTT), which are widely used for IoT systems. Further, Node.js programming is taught and students learn how to use node.js for HTTP and MQTT implementation. The IoT cyber security is also covered. The following subsections will give examples for MQTT protocol implementation and IoT cyber security.

MQTT Implementation

MQTT is an open standard, simple, lightweight network protocol that transports messages between devices (Dinculeană, 2019). The MQTT is designed for constrained devices and low-bandwidth, high-latency network, so it is ideal for M2M (machine-to-machine) or IoT world of connected devices and for mobile applications where bandwidth and battery power are at a premium.

MQTT adopts a publish-subscribe model in which a publisher publishes messages on a topic (analogous to TV/radio channel) and a subscriber must subscribe to that topic to view the message. The study of MQTT will help students understand how messages transport in IoT applications. We also teach students how to implement MQTT protocols using Node.js. Node.js is an open-source, event-driven JavaScript runtime environment that executes JavaScript code outside a web browser. It is popularly used in the industry and students will gain this required skills by working on protocol implementation using Node.js. Brokers are required for MQTT implementation. There are many options for brokers including Mosquitto, HiveMQ, Apache ActiveMQ, RabbitMQ, etc. In class, we use Mosquitto as broker, which is the most popular open source MQTT broker.

In class, we demonstrated how to set up a MQTT broker to student. Then students are assigned a project to design and implement MQTT protocol using two computers and Arduino boards. In the first two years, students are asked to set up two clients/computers from each side, and then communicate with each other via Mosquitto Broker using MQTT protocol. The MQTT diagram is described in Fig. 1. Fig. 2 shows one student team's submission where two students work as a team. Student Salih serves as the publisher client and publish a message to the broker, while student Haolin serves as the subscriber client and is expected to receive message from Salih via the broker. Fig. 2 (a) is the screenshot of running pub.js file that contains the code of publisher client. Fig. 2 (b) is the screenshot of running sub.js file that contains the code of subscribe client. It shows "Hello Haolin" message received sent from the publisher via the broker.

Fig. 1 Diagram of MQTT

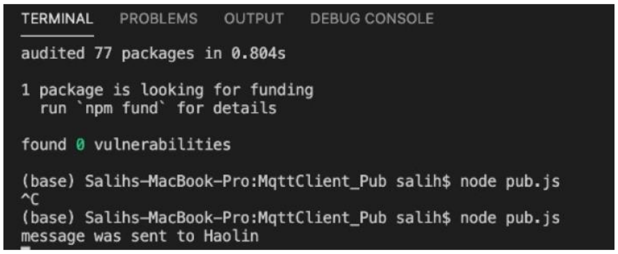

Fig. 2 (a)

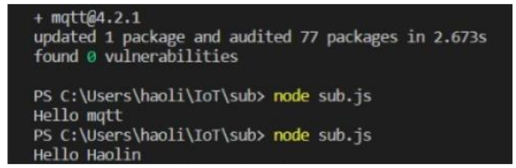

Fig. 2 (b)

Fig. 2 Student submission of MQTT implementation using Node.js

Practical hands-on experiences in engineering courses support student learning outcomes in both the affective and cognitive domains (Windsor, 2017). To give students more hands-on experiment experiences, the project assignment is extended to send and receive MQTT messages using Arduino WiFi 1010 in fall 2022. The following major steps are involved: (1) build a simple circuit that uses a pushbutton to turn on/off an LED; (2) Install WiFiNINA library and the ArduinoMqttClient library; (3) Program the Arduino board as an MQTT client that connects to a broker (https://shiftr.io/try can be used as a broker), subscribes to a topic, and both listens for messages on that topic and sends message to it. When a pushbutton is pressed, the client sends a message (e.g., a random number). When the client receives a message, and if the number is greater than 0, it sets an LED to full. When nothing happens, if the LED is not off, it's faded down one point every time through a loop. Students said they learned more from the improved version of MQTT implementation using Arduino WiFi 1010.

IoT Cyber Security

Cyber security is one of top concerns for any system including IoT-related ones. This topic is not covered in the first two years. In the third year, we address cyber security of IoT systems in Module 2 including both hardware and software attacks. In terms of security, existing attacks mainly focus on the vulnerabilities of networking and communication protocols, while neglecting the problems arising from attacks to IoT devices. In class, we use an attack to contact sensors in smart home as an example (Li, 2021). Contact sensors are security devices typically installed on doors and windows. They notify us about our system (e.g., a door, window, or gate) in the event of open, and send an alert during an intrusion. However, an adversary may use its own magnet (i.e., a malicious magnet) to control the state of the reed switch, further implement event-eliminating attack and event spoofing attack. Further, the communication attack of jamming to contact sensors is also introduced, which will quickly deplete sensor's battery. Transduction attack is also introduced, which is referred to an attacker using external signals (such as sound, electromagnetic waves, electric signals, etc.) to trick a sensor into reading incorrect data on purpose. A Youtube video of DolphinAttack (Zhang, 2017), one type of transduction, is played to students, in which attackers send inaudible ultra-sounds to trick speech recognition systems (Siri, Alexa, Google Now, etc.) into executing commands.

Hands-on Experiment Design

From lecture study in three modules, we have taught sensors and the sensing principles for enabling IoT system. We also learned IoT framework to transform sensing data into insights and actions, as well as sending sensor data via networks mainly wireless and the protocols of networks. To help students deepen their understanding of how to design and implement a practical IoT system, we design several hands-on experiments using Arduino WiFi1010 and different type of grove sensors. This section takes the term project as an example to illustrate our experiment design on an IoT-enabled smart home. The purpose of this project is to provide hands-on experience in the design and implementation of an IoT application from the far edge to the cloud to the end-users using an IoT analytics platform service (ThingSpeak) and Matlab as main tools.

Students will gain experience with MatLab, ThingSpeak, Simulink and Restful API to design and analyze IoT systems. ThingSpeak is an open data platform for the IoT system. A device or application can communicate with ThingSpeak using a RESTful API, and students can either keep their data private, or make it public. In addition, ThingSpeak can be used to analyze and act on data (e.g., sending email notification). Specifically, in the hands-on term project, students need to put all these pieces together to simulate an IoT-enabled smart home. Students are provided different use cases that emulates IoT smart home features. Each use case is composed of inputs, actions, and outputs. The use case on Smart HVAC (heating, ventilation, and air conditioning) is designed as follows:

Use Case: Smart HVAC

In this use case, students simulate the operation of a super-smart, next-gen HVAC system. Because of physics, the dehumidifying functions of an air-conditioning system require the HVAC to be running cooler than the ambient temperature of the home. This is why we turn on the AC to dehumidify our home.

Table 2. Input, Action and Output for the Smart HVAC design

Input	Action	Output
1) Home Internal Temperature Data (Arduino Sensor) 2) Home Internal Humidity Data (Arduino Sensor) 3) UV Sensor (Arduino Sensor by a window to measure outside brightness)	1) Humidity Control If it is acceptable to cool the home, use the AC, otherwise, use the dehumidifier. 2) Temperature Control If you need to cool the home, use the AC; If you need to heat the home, use the heater; If it is daytime && you need to heat the home & it is bright outside (UV ensor), the blinds should be open; If it is daytime && it is not bright outside, the blinds should be open; For all other conditions, the blinds should be closed.	1) AC Control (Simulated Output) 2) Heater Control (Simulated Output) 3) Blinds Control (Simulated Output) 4) RHED Control (Simulated Output)

The next-generation smart home has a Revolutionary High-Energy Dehumidifier (RHED) that does not cool home down. However, to save cost, we only want to run it when lowering the temperature of home is not necessary. The control system for this smart HVAC is designed using ThingSpeak. The control system for this smart HVAC is designed using ThingSpeak. Temperature and humidity sensors distributed to students will be used. In addition, automated window blinds may provide an additional control factor. The input, action and output of this smart HVAC system is presented in Table 2.

Student Design

The design diagram for the use case Smart HVAC from one student team is presented in Fig. 2. A temperature sensor, a humidity sensor and a UV sensor on Arduino are used to collect environmental data. Specifically, the output voltage of UV sensor to define the brightness. The value '0' (voltage <50) indicates the outside is dark, the value '1' (50<voltage<227) indicates the outside is not bright and the value '2' (227<voltage) indicates the outside is bright. The working frequency of this sensor module is every 1 minute. These three types of data are sent to channels in ThingSpeak, which has 7 fields. In addition, React APP is used to perform a MATLAB analysis. In the MATLAB analysis, the humidity control is first implemented: If the humidity in home is greater than 60% and the temperature in home is greater than 78°F, '1' is written to the field 4 to turn on the AC; If the humidity in home is greater than 60% and temperature in home is not greater than 78°F, '1' is written to the field 7 to turn on the dehumidifier. Besides humidity control, temperature control, brightness control are also implemented in a similar way. Student have enjoyed working on this term project and have gained rich hands-on experiences from building such an IoT-enabled smart home.

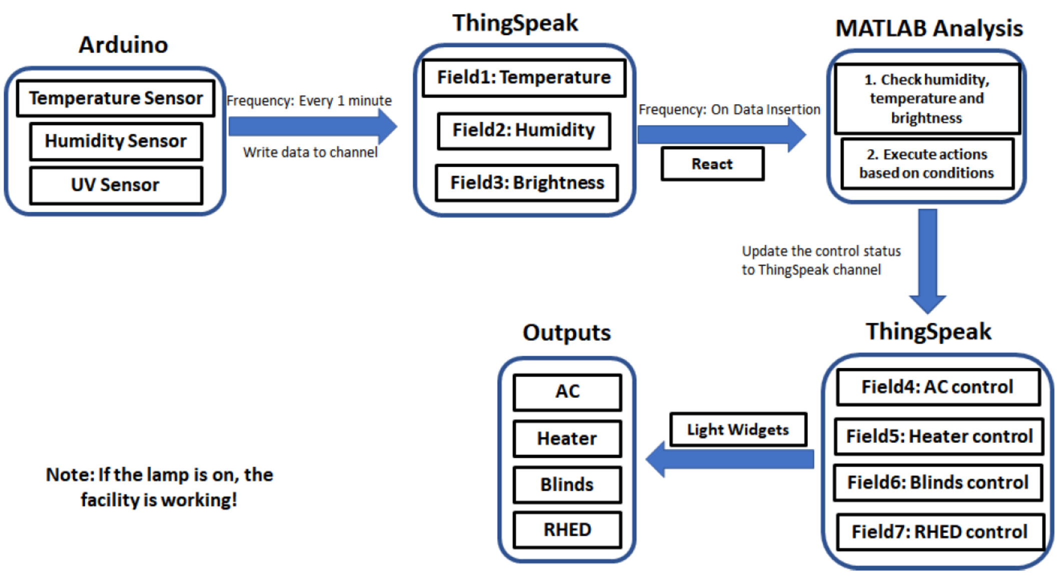

Fig. 2 Design Diagram of IoT-enabled Smart HVAC

Student Feedback

This course has received positive comments and a few constructive comments from students. This section summarizes both of them, and also propose actions to improve this course in future. Most students feel that they have gained a lot from three modules. The following positive comments are listed as follows.

--- *"I have gained a greater swath of background to pull from for materials science and even a greater understanding of semiconductors. I have gained far more experience in node.js than I would have otherwise as well as greater practice with the interesting ThingSpeak platform."*
---*"Everything the course set out to teach I have learned, honestly I feel that this course is very useful for ECE majors and should be a required course maybe."*
---*"I learned what an IOT system is. I learned a ton about how sensors and protocols work. Node JS and ThingSpeak were very cool to learn and I will most likely be using these tools in my future."*

In addition, students provide constructive comments on course design. From students' feedback, we identified issues in terms of student preparation, pace, and assessment

- *Student Preparation.* Although fulfilled the prerequisites of the course, some students, especially those in the Computer Engineering program, were not able to connect the new topics in module 1 to their prior knowledge system. As a result, the perceived difficulty level of module 1 was higher than other two modules.
- *Pace*. Modular courses do not necessarily have faster paces than traditional courses. However, the students felt they needed more time because the topics were "difficult" (mainly due to insufficient preparation as discussed above) and/or because the cognitive load of the course, especially module 1, is high.
- *Assessment*. Disparities between assessments in different modules were observed. This could have been caused by student understandings of the topics in each module and/or learning expectations of the instructors. Students also noticed their grades in the course relied heavily on their programming skills. This caused confusions to students and raised concerns about fair grading between modules.

Future Work

To address the three major concerns listed above and to enhance the teaching and learning quality, we will follow the guidance of constructive alignment (Biggs, 2014) in the future course development and improvement. More specifically, we will apply backward design and set clear and feasible learning outcomes for each module according to the zone of proximal development (Wertsch, 1984) and then modify the teaching activities and assessment strategies to support the learning outcomes. For example, some students especially in Computer Engineering felt challenging with the first module because they had no background on devices and electromagnetics. We plan to revise the lectures in module 1 to focus more on sensing technology and principles in IoT system. We will also provide reviews of the relevant topics at the beginning of each module to facilitate the instructional scaffolding. We will establish routine co-planning time (Walther-Thomas, 1996) to improve communication and coordination among modules. In addition, we will look into other commonly-used industrial IoT platforms. Google Cloud IoT platform might be a good candidate, in which MQTT and HTTP protocol bridges are used for device connection and communication with the Google Cloud Platform. MQTT and HTTP are already covered in our class with Node.js implementation.

Conclusions

As an emerging subject, IoT covers a broad spectrum of content and has been applied in various areas. However, there is no mature model available for teaching this subject. This paper presented our course design for IoT including topic selection and pedagogical practices. Anecdotal feedback from students suggested that our module-based course design open a new way to teaching courses. We have received constructive comments from students regarding the scope of the course and its modular structure. In the next iteration, we will strengthen instructor coordinations to improve course consistency and adjust certain topics in each module to better scaffold learning.

References

Biggs, J. (2014). Constructive alignment in university teaching. *HERDSA Review of Higher Education*, *1*, 5-22. Retrieved from https://www.herdsa.org.au/herdsa-review-higher-education-vol-1/5-22

Hassan, W. H. (2019). Current research on Internet of Things (IoT) security: A survey. *Computer networks*, *148*, 283-294.

Elnashar, A. (2019). IoT evolution towards a super-connected world. *arXiv preprint arXiv:1907.02589*.

IoT Job Market, https://www.datamation.com/careers/iot-job-market/

Vesikivi, P., Lakkala, M., Holvikivi, J., & Muukkonen, H. (2019). Team teaching implementation in engineering education: teacher perceptions and experiences. *European Journal of Engineering Education*, *44*(4), 519-534. https://doi.org/10.1080/03043797.2018.1446910

Steele, J. S., Cook, L., & Ok, M. W. (2021). What Makes Co-Teaching Work in Higher Education? Perspectives from a Merged Teacher Preparation Program. *Issues in Teacher Education*, *30*, 4-31. Retrieved from https://eric.ed.gov/?id=EJ1337912

Dehnad, A., Jalali, M., Shahabi, S., Mojgani, P., & Bigdeli, S. (2021). Students' view on supportive co-teaching in medical sciences: a systematic review. *BMC Medical Education*, *21*, 1-13. https://doi.org/10.1186/s12909-021-02958-4

Bacharach, N., Heck, T. W., & Dahlberg, K. (2007). Co-teaching in higher education. *Journal of College Teaching & Learning (TLC)*, *4*(10). Retrieved from https://www.clutejournals.com/index.php/TLC/article/download/1532/1512

Beck, A. (2006). Adventures in team teaching: Integrating communications into an engineering curriculum. *Teaching English in the Two Year College*, *34*(1), 59. Retrieved from https://www.proquest.com/docview/220967194?pq-origsite=gscholar&fromopenview=true

Walther-Thomas, C., Bryant, M., & Land, S. (1996). Planning for effective co-teaching the key to successful inclusion: The key to successful inclusion. *Remedial and special education*, *17*(4), 255-264. DOI: 10.1177/074193259601700408

Felder, R. M., Bernold, L. E., Burniston, E. E., Dail, P. R., & Gastineau, J. E. (1996, June). Team Teaching in An Integrated Freshman Engineering Curriculum. In *1996 Annual Conference* (pp. 1-434). DOI: 10.18260/1-2—6341

Ferguson, P. C., Kraemer, W., Nousiainen, M., Safir, O., Sonnadara, R., Alman, B., & Reznick, R. (2013). Three-year experience with an innovative, modular competency-based curriculum for orthopaedic training. *JBJS*, *95*(21), e166. DOI: 10.2106/JBJS.M.00314

French, S. (2015). The benefits and challenges of modular higher education curricula. *Issues and Ideas Paper*, (201), 1-12. https://melbourne_she.unimelb.edu.au/__data/assets/pdf_file/0006/2774391/Benefits_Challenges_Modular_Higher_Ed_Curricula_SFrench_v3-green-2.pdf

Hodgson, A., & Spours, K. (2014). Modularization and the 14-19 Qualifications System. In *Dearing and beyond* (pp. 111-126). Routledge.

Dinculeană, D., & Cheng, X. (2019). Vulnerabilities and limitations of MQTT protocol used between IoT devices. *Applied Sciences*, *9*(5), 848.

Li, T., Han, D., Li, J., Li, A., Zhang, Y., Zhang, R. & Zhang, Y. (2021). Your home is insecure: Practical attacks on wireless home alarm systems. *IEEE INFOCOM 2021-IEEE Conference on Computer Communications* (pp. 1-10).

Zhang, Guoming, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, & Wenyuan Xu. "Dolphinattack: Inaudible voice commands." In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, (pp. 103-117). 2017.

Wertsch, J. V. (1984). The zone of proximal development: Some conceptual issues. *New Directions for Child and Adolescent Development*, *1984*(23), 7-18.