An Introductory Internet of Things Curriculum for Grades 9-12 Computer Science Classes

Pramod Abichandani
Electrical and Computer Engineering
New Jersey Institute of Technology
Newark, NJ, USA
pva23@njit.edu

Deepan Lobo
Electrical and Computer Engineering
New Jersey Institute of Technology
Newark, NJ, USA

Prateek Shekhar
Applied Engineering and Technology
New Jersey Institute of Technology
Newark, NJ, USA

***Abstract*—The emergence of the Internet of Things (IoT) has had a transformative effect on our society and has inspired educators to develop innovative approaches to educate the next generation of Computer Science (CS) professionals. This paper presents the design and development of an introductory IoT course suitable for grades 9-12 Computer Science classes. Information about the course content, intended outcomes, and evaluation techniques are presented. The course was introduced in 2 high schools in the US. The course includes a capstone project where the students identified a real-world problem and developed an IoT-based solution to address it. Formative and summative technical evaluation results are presented and suggest that the course provided an effective learning experience for students. The information presented here provides guiding principles for developing an IoT-based curriculum geared towards 9-12 education while also exposing the students to CS fundamental.**

I. Introduction

Internet-of-Things (IoT) is a disruptive technology which has enabled ubiquitous and pervasive computing scenarios [1]. The IoT refers to the system of inter-connected computing devices, mechanical and digital devices, and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. The IoT market is continually growing and are expected to grow to 1.6 trillion U.S. dollars in size by 2025 [2]. There is an increasing interest in using IoT technologies in agriculture, food processing industry, environmental monitoring, security surveillance, robotics and drones, education, and other significant industries [3]–[5]. These industries correlate closely with current rapidly growing need for software developers equipped with skills of computer programming that leverage the intersection between hardware and software. Researchers have recognized this need to train future software developers equipped with the knowledge in IoT [6], [7].

A. The 4-Layer IoT Model

The multi-dimensional nature of IoT technologies have been succinctly captured in multi-layered conceptual architectures by several authors [4], [9]–[13]. A popular model is the IoT World Forum Reference Model that has been widely adopted in the literature [14]. While these conceptual architectures feature different numbers and names for IoT layers, they have strong overlaps in terms of the core IoT paradigms. In its most general form, IoT can be conceptually expressed as a system formed by the interaction of 4 key dimensions: real-world sensing, communication networks, software services, and user interfaces. These dimensions result in a 4-layer model that was proposed by *Da et. al.* in [4], and is depicted in Figure 1. When viewed through the lens of IoT/STEM education, this 4-layer model allows for educators to design curriculum, pedagogy, and assessment in a multi-disciplinary manner while immersing students in the technical and societal challenges associated with IoT [6]. The curriculum developed as part of this effort was contextualized in this 4 layer-model and is presented in the following. Each of the course modules were desgined around active learning approaches guided by the *Interactive Constructive Active Passive* (ICAP) modes of cognitive engagement [8]. The ICAP framework enables educators to design and examine active learning approaches due to the observable and assessable nature of the four student engagement modes it outlines. Table I summarizes the observed student engagement behaviors in the context of the ICAP framework during the 3 course modules.

One crucial skill of a software developer is effective computer programming. Characteristics that define a skilled computer programmers are assessed on four dimensions namely critical thinking, problem-solving, computational thinking, and system design [6]. The need for computational thinking and improvements in the programming skills of individuals continue to receive widespread attention to better prepare future citizens [15]. Today, basic knowledge of programming is part of almost all engineering programs curricula. At times, both students and instructors can find the programming concepts and language syntax cumbersome ultimately negatively affecting student motivation [16]. Researchers report that high number of concepts and skills to learn in a limited time, absence of incorporation of interactive media and instant feedback instruction, and lack of mathematical background are the main barriers in learning programming [17].

IoT presents a novel paradigm for enhancing computer science education. An IoT system involves physical computing at its core – the use of inexpensive microcomputers that run optimized code to collect, analyze, and share data with other devices or a cloud server. Due to the inherent integration between hardware, software, real-time data, software processes, and human interaction, IoT has the potential to serve as an

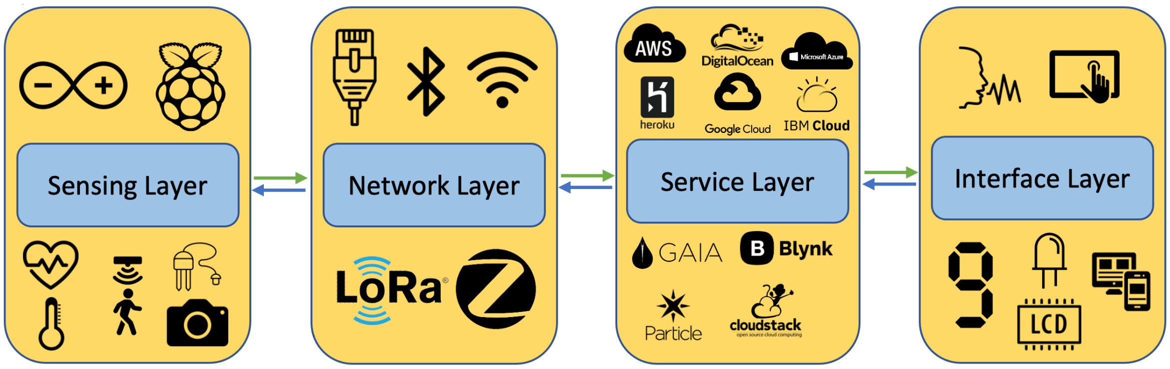

Fig. 1: The IoT 4-layer architecture encompassing IOT systems [6].

excellent platform for teaching CS topics in an experiential format. IoT can be used to introduce CS principles in 9-12 due to its heavy focus on physical computing, and the availability of easy-to-use programming languages such as Python and NodeMCU/Lua [18]. These modern programming languages reduce the barrier to entry in C.S. for 9-12 students. For example, the authors in [19] report on the use of IoT hardware and software for 6-12 summer programs. Students focused on creating IoT prototypes using pre-made code templates. Similarly, recent projects from the U.S. National Science Foundation (NSF) are also creating an IoT pedagogical ecosystem for integrated C.S. and Software Engineering education for 9-12 students [20]. Architecturally, IoT-based curricula can leverage learning frameworks consisting of a collection of integrated computing devices and instruments for mapping abstract computer programming concepts and theories into practical, hands-on experiential knowledge [21]. Readers interested in a comprehensive understanding of the literature covering IoT based curriculum, pedagogy, and assessment are referred to [6] and its references.

This paper presents the design and development of an IoT course that introduces computer programming concepts for grades 9-12. The coursework is designed around the interactive-constructive-active-passive (ICAP) cognitive modes of engagement presented in the ICAP framework [8]. The curriculum was tested at 2 high schools located in the US. The key contributions are highlighted as follows:

- A course design that introduces CS concepts to students of grades 9 - 12 contextualized in the sensing, network, service, and interface layers of IoT.
- Quantitative formative and summative evaluation results of the first iteration of this course that provide insights about student performance.

II. Methodology

A. Curricular Modules

The accessibility of single-board computing devices such as Arduino and Raspberry Pi, and the abundance of online resources has lowered the barrier of entry for students interested in learning programming. Thus, one of the main motivations of this study is to present an organized course structure that leverages the experiential component of IoT devices with the software concepts of programming. The curriculum is designed around programming IoT sensor hardware with a modern computer programming language. Figure 2 depicts the essential elements of this curriculum. Table I depicts the ICAP framework mapping.

TABLE I: Student engagement modes observed in the context of the ICAP framework during the course modules [8].

#	Course Module	Interactive	Constructive	Active	Passive
1.	Introducing Python	Interact with instructional staff during the lecture, collaborate with team members on programming activities	Creating Python Programs based on instructions during class session	Commenting Python code demonstrated in the classroom	-
2.	4-Layer IoT Model (Sensing and Interface Layer Integration, and Network and Service Layer Concepts)	Interact with instructional staff during the lecture, collaborate with team members on mini IoT projects	Creating Python programs based on instructions during workshop session, Building circuits on breadboard	Debugging sensor hardware, electrical connections, and Python programs	-
3.	Final Project	Weekly group meetings, Discussions during progress presentations, project demo video creation	Open-ended programming in Python, constructing sensor housing, electro-mechanical assembly of the IoT project, and evaluating systems for correctness	Researching online resources such as YouTube videos and Online Blogs	-

1) Weeks 1 - 2: Introducing Python: The programming language used in the curriculum is Python3, which is an interpreted, high-level and general-purpose programming language that is concise and easy to read. Python is considered as one of the most popular programming languages and combined with its accessibility and versatility makes it the programming language of choice for first time users [22].

This module guides the students through the software installation process of the Python programming environment, code editor, and necessary software driver support for the IoT sensor hardware. Students were introduced to several Python Integrated Development Environments (IDEs). IDEs contain many useful features to write, edit and debug code, visualize and inspect data, store variables, and present results. While IDEs might differ visually, the programming language does not. Thus, the learning curve is limited to understanding Python's syntax.

After this introductory setup, students were introduced to fundamental programming constructs of Python. These programming constructs include variables, mathematical operations, input/output commands, conditionals, loops, and lists. These programming constructs introduced in the first two weeks of the course provide the building blocks for developing the first Python program to interface with sensor hardware.

2) Weeks 3-5: Sensing Layer Integration: During the next 3 weeks, students were introduced to IoT sensor hardware. The sensing layer is the primary layer in any IoT-based system and includes sensors or transducers embedded in real-world environments [6]. The sensors included in the curriculum are classified into the two categories namely environmental sensors and proximity/motion sensors. A key aspect of the curriculum was the use of easy-to-use Python Application Programming Interface (API) developed by the authors. These APIs enabled easy interface with the sensor hardware and actuators. Each lecture session began with the instructor first providing background about the hardware. Students were then asked to discuss the real-world applications of the sensor and identify any commercial products that use the particular sensor. This discussion was followed by hands-on sessions where the students were introduced to the template Python code needed to interface with the sensors. The Python code allowed students to acquire sensor data in real-time and display it to the user. After the students successfully interfaced with the hardware they moved on to associated coding exercises. These exercises allowed students to explore different logical/programming constructs associated with the sensors. Students used if/else conditions to display useful messages on the Python console based on sensor values, practiced average multiple data values, and visualized sensor data using Python plotting functionalities.

3) Weeks 6-7: Interface Layer Integration: This module was covered after students completed the exercises of the sensing layer and guided the students to create interfaces that allowed users to interact with an IoT system. This layer included the use of hardware displays such as LEDs (mono-colored or multi-colored), Liquid Crystal Displays (LCDs), touch screens, and actuating devices such as motors and buzzers. Students learned how to use Python programs to create real-time visualizations of the sensor data obtained from the sensing layer. On the software side, interfaces such as dashboards, text-box alerts and prompts, and text notifications formed a strong focus in this layer.

4) Week 8: Network Layer Concepts: The network layer module covered theoretical aspects of technologies that enable connection between IoT system elements and exchange information with each other. This layer is significant as IoT-based systems are connected through wired or wireless networks. Wi-Fi, Bluetooth, Zigbee, LoRa are the most popular wireless technologies for communication used by IoT devices and systems. The instructor covered the basic overview of these

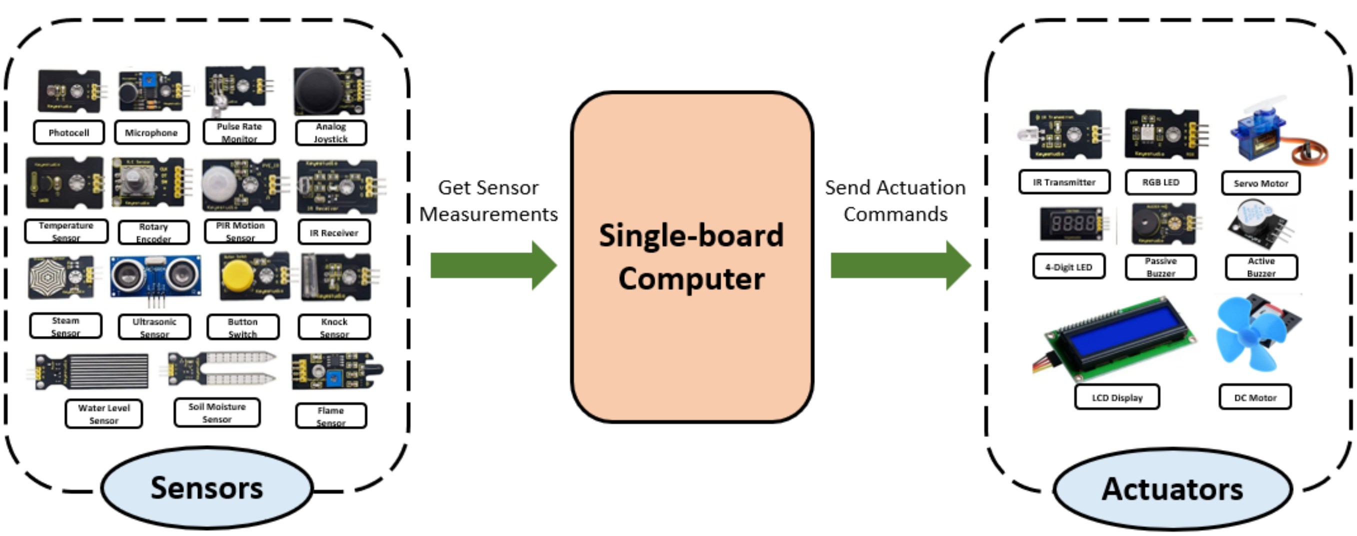

Fig. 2: Conceptual diagram of essential tasks in coding exercises

technologies in class while students discussed their experiences with these technologies. The instructor emphasized the fact that the choice of a network layer technology in an IoT system requires considerations of cost, data rates, ease of implementation, performance, battery life, reliability, and any governmental restrictions. Students also discussed the strengths and weaknesses of these technologies.

5) **Week 9: Service (Cloud) Layer Concepts:** The final module introduced concepts of cloud technologies and services. Cloud services provide the backbone to the IoT architecture and are crucial component in understanding the working of internet of things. This module covered theoretical concepts on how cloud services function followed by discussions in the classroom. Students were introduced to different vendors (Amazon, Google, Microsoft, and IBM among others) that provide cloud services and the breadth of cloud services provided by such vendors.

The assignments contained a variety of coding exercises involving the sensors and actuators. Through these programming exercises, students' ability to leverage the template Python code provided in the introduction and build sophisticated functionalities was evaluated. The coding exercises were designed according to the 3A and 3B standards (grade level 9 -12) outlined by the Computer Science Teachers Association (CSTA) [23]. Rubrics to formatively evaluate the student's comprehension of the CSTA standards-aligned concepts were used. These concepts included 1) Algorithms and Programming, 2) Computing Systems, 3) Data and Analysis, 4) Impacts of Computing, and 5) Network and the Internet .

B. Final Project

The final component of the course featured student teams working on a capstone IoT project. The project spanned 6 weeks of the term and began as the students started completing their sensor layer assignments. The primary instructional objective of this project was to allow student teams to work together effectively and apply the skills/knowledge learned from the course to build a tangible solution to a real-world problem. Students were free to decide their project topic and project domains varied from environmental monitoring, home automation, connected infrastructure, and connected biomedical devices. The project was broken down into the following phases:

1) **Weeks 5 - 6: Team Formation and Project Proposal**: Leveraging their newly acquired knowledge and skills, students worked together and brainstormed ideas to solve a real-world problem through an IoT solution. Students were asked to form teams and propose their projects by Week 7 of the terms. Student teams could have between 2 and 3 members. Students were also free to work independently.
   A template of the proposal document was provided to the students. The template included prompts to engage students to think deeply about their project idea and team composition. The prompts included questions about them: 1) problem statement, 2) commercial/societal significance, 3) technical approach, 4) key innovation, 5) expected outcome and deliverables, and 6) resources needed.
   Students used their school's Learning Management Systems (LMS) to float ideas, discuss potential collaborations, and coalesce around a coherent project narrative with other classmates. During this period, students were required to engage with the instructors through weekly 1-on-1 meetings during their office hours to craft their projects' scope and technical deliverables appropriately.
2) **Weeks 7 - 9: Weekly In-Class Project Progress Presentations**: After the project proposal was approved by the instructors, students were expected to lead the project execution while having short interactions with the instructors. The key requirement during this phase was a 3-minute long weekly in-class progress presentation that outlined their weekly progress. The progress presentation featured three slides that covered the weekly progress, challenges, and expected milestones for next week.
3) **Week 10: Project Video**: The final deliverable for the capstone project was a presentation along with a technical report. The presentation was to include the following information: 1) Introduction and project statement, 2) Demonstration of the functions of the IoT project, 3) Description of design challenges and approaches adopted to solve them, 4) Learning resources referred to while working on the project, 5) Anticipated challenges at the beginning of the project, and 6) Approaches taken to solve the challenges.

III. Technical Evaluation

Formative assessments were used for regular evaluation of the technical topics module of the curriculum. Summative assessments were used to evaluate the final project. A total of 19 students were part of this implementation across the 2 high schools in their CS/STEM classes. Both classes featured students from multiple grade levels (9-12) They had some previous experience with block based coding but had not worked on Python. Approximately 60% of the participants identified as Caucasian whereas the remaining students identified as non-Caucasian.

A. Direct Formative Assessment

The course featured assignments that were due every week. The assignments consisted of a list of mini-projects featuring sensors and actuators being interfaced using Python. Each

TABLE II: Descriptive Statistics of grades obtained by 19 students on their assignments (A1 - A5). Each assignment was worth 10 points.

Assignment	1	2	3	4	5
Mean	8.47	7.18	8.58	9.42	8.89
Std. Dev.	2.52	1.52	2.13	2.20	1.41

assignment was designed around the list of sensor hardware discussed in class that week. The mini-project were formulated such that the students were met with an increasing level of difficulty, where the specifications for each project built on top of the previous project. The assignments were worth 10 points each. A detailed description of the assignments are provided as follows:

1) Basic components: This assignment featured the basic components of an embedded systems such as LEDs, RGB LED, push button, and analog sensors including the photo resistor and Thermistors. Students were guided through the steps of creating simple circuits on a breadboard. The program specifications introduced the students to essential Python programming concepts such as data types, print statements, and conditional logic.
2) Loops: This assignment introduced students to the concept of looping in programming. Looping is a beneficial feature of programming which allows the code to be written once and repeated as many times as needed, thus shortening the size of the program. The questions included hardware from the first assignment and required the students to reinforce programming concepts such as printing and conditional logic. Additionally, looping assisted students to perform sensor data acquisition.
3) Actuators: This assignment introduced a variety of actuators used in embedded systems such as DC motor, servo motor, and active/passive buzzers. Students were guided through the process of reading sensor measurements obtained from the environment designing logic to create real-world actuation in a meaningful way. For example, increasing the speed of DC motor based on increasing temperature measurements to simulate a cooling system.
4) Displays: This assignment introduced sophisticated display technologies found in real-world systems such as 7-segment LED, LCD, and Dot-Matrix LED. Creating expressive prompts, and warning messages to convey to users important information is a crucial component of the interface layer of an IoT system. This assignment also introduced additional sensor hardware to expand the available input devices for the project, namely water level sensor and a Passive Infrared (PIR) sensor. Students also attempted to build text box based notifications and on-screen displays.
5) Network Layer and Cloud Services: These assignments immersed students in the technical challenges of selecting an appropriate networking technology for a given IoT application. Additionally, students were asked to architect an IoT system end-to-end while using appropriate cloud services. Amazon Web Services (AWS) were used as the reference cloud services for this assignment.

Table II depicts the mean and standard deviation of each assignment grade for 19 students. The mean of the grades received by the students for the IoT-based Python assignments varied from the 7.18 to 9.42. The standard deviation observed for the IoT-based Python assignments varied from 1.52, to 2.52. Several well-established rubrics from the literature were adapted to assess the CS, networking, and cloud aspects of the assignments [24]–[26].

B. Direct Summative Assessment

The project submission was evaluated according to the rubric provided in Table III. The rubric was adopted from resources available on [27] while keeping in mind the objectives of the capstone project. Based on the final presentations, 90% demonstrated a fully working project as per the proposed description, and only 10% presented a partially working project where they had a physical model, but the system was not working as intended.

C. Final project themes

The IoT projects developed by the students were classified based on their applications. Among the projects, 62% were based on the theme of environmental monitoring, which included technical challenges such as building a sensor dashboard to log temperature, light, humidity, and moisture levels in the environment. Approximately 23% of the projects were based on the recurring theme of home security and were designed around motion/proximity sensors such as ultrasonic sensor, Passive Infrared (PIR), and infrared. The remaining projects used a combination of environmental and motion sensors for a smart home application.

TABLE III: Rubric for IoT Capstone Project

#	Metric	Excellent	Average	Poor
1	Project Completion	All deliverables met	Some deliverables met	No deliverables met.
2	Video presentation	Meets all the video presentation requirements, excellent presentation skills	Partly meets the requirements of the video presentation, And mediocre presentation skills	Does not meet any of the requirements of the video presentation, or no video submitted.
3	Communication	Excellent at communicating challenges and progress made	Average at communicating challenges and progress made	Poor at communicating challenges faced and little to no progress weekly
4	Innovation	Implemented a unique feature or aspect to the project	Proposed an innovative concept but did not deliver	No innovation implemented
5	Realizability	Project is working, and real-world application demonstrated	Project is working, but real-world applicability, not possible	Project is not working, and real-world applicability is not possible.
6	Report organization	Excellent presentation demonstrates understanding of IoT technology and programming	Average presentation, satisfactory understanding of IoT technology and programming	Poor organization of report, and sub par understanding of IoT technology and programming

IV. Conclusion

This paper presented an introductory IoT-based, standards-aligned curriculum for CS classes for grades 9-12. The focus of the curriculum was on the use of IoT architectural layers to create hands-on, physical computing-centric learning experience. The final projects allowed students to explore team-based IoT solutions for real-world problems. The curriculum was implemented in 2 high schools in their CS/STEM classes. Formative and summative evaluations obtained during the first implementation of this curriculum suggested that students were able to grasp IoT and CS concepts. Subsequent to this initial implementation, future work will focus on refining the curriculum and its implementation, conducting experiments involving multiple control and experimental groups of students for data collection, and qualitative assessment via surveys and focus groups to codify student attitudes towards IoT curricular experiences. As IoT-centric education becomes more pervasive in high schools, the curricular information provided in this paper may serve as a guide for educators and researchers.

V. Acknowledgment

This work was supported by the National Science Foundation under Grant DRL-2010259.

References

[1] M. H. Miraz, M. Ali, P. S. Excell, and R. Picking, "A review on Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT)," in *2015 Internet Technologies and Applications (ITA)*. IEEE, 2015, pp. 219–224.

[2] L. S. Vailshery, "Forecast end-user spending on IoT solutions worldwide from 2017 to 2025," https://www.statista.com/statistics/976313/global-iot-market-size/, 2021.

[3] C. Iaboni, H. Patel, D. Lobo, J.-W. Choi, and P. Abichandani, "Event camera based real-time detection and tracking of indoor ground robots," *IEEE Access*, vol. 9, pp. 166 588–166 602, 2021.

[4] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[5] S. Li, L. Da Xu, and S. Zhao, "5G Internet of Things: A survey," *Journal of Industrial Information Integration*, vol. 10, pp. 1–9, 2018.

[6] 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*, 2022.

[7] P. Abichandani, D. Lobo, C. Berry, V. Parikh, W. Fligor, and W. McIntyre, "Matlabarduino. org: An open-source website and youtube channel for embedded systems education," in *2021 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2021, pp. 1–8.

[8] M. T. Chi and R. Wylie, "The ICAP framework: linking cognitive engagement to active learning outcomes," *Educational psychologist*, vol. 49, no. 4, pp. 219–243, 2014.

[9] J. Gómez, J. F. Huete, O. Hoyos, L. Perez, and D. Grigori, "Interaction system based on Internet of Things as support for education," *Procedia Computer Science*, vol. 21, pp. 132–139, 2013.

[10] S. Li, L. Da Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

[11] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys and tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[12] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future generation computer systems*, vol. 56, pp. 684–700, 2016.

[13] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[14] "IoT World Forum Reference Model," http://cdn.iotwf.com/resources/72/IoT_Reference_Model_04_June_2014.pdf, (last visited 2021-07-26).

[15] D. Pérez-Marín, R. Hijón-Neira, A. Bacelo, and C. Pizarro, "Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children?" *Computers in Human Behavior*, vol. 105, p. 105849, 2020.

[16] D. Topalli and N. E. Cagiltay, "Improving programming skills in engineering education through problem-based game projects with scratch," *Computers & Education*, vol. 120, pp. 64–74, 2018.

[17] M. Rizvi, T. Humphries, D. Major, M. Jones, and H. Lauzun, "A CS0 course using Scratch," *Journal of Computing Sciences in Colleges*, vol. 26, no. 3, pp. 19–27, 2011.

[18] "NodeMCU an open-source firmware based on ESP8266 Wi-Fi." [Online]. Available: https://www.nodemcu.com/index_en.html

[19] A. Jaklič, "IoT as an Introduction to Computer Science and Engineering:A case for NodeMCU in STEM-C education," in *2020 IEEE Global Engineering Education Conference(EDUCON)*. IEEE, 2020, pp. 91–95.

[20] "National Science Foundation(NSF)-Division of Research on Learning." [Online]. Available: "https://www.nsf.gov/awardsearch/showAward?AWD_ID=2010259"

[21] 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," in *2016 IEEE frontiers in education conference (FIE)*. IEEE, 2016, pp. 1–9.

[22] F. Klint, "Python is more popular than ever." [Online]. Available: "https://www.wired.com/story/python-language-more-popular-than-ever/"

[23] D. Seehorn, S. Carey, B. Fuschetto, I. Lee, D. Moix, D. O'Grady-Cunniff, B. B. Owens, C. Stephenson, and A. Verno, *CSTA K–12 Computer Science Standards: Revised 2011*. ACM, 2011.

[24] S. Basu, "Using rubrics integrating design and coding to assess middle school students' open-ended block-based programming projects," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 2019, pp. 1211–1217.

[25] E. Snow, D. Rutstein, M. Bienkowski, and Y. Xu, "Principled assessment of student learning in high school computer science," in *Proceedings of the 2017 ACM Conference on International Computing Education Research*, 2017, pp. 209–216.

[26] T. Ketenci, B. Calandra, L. Margulieux, and J. Cohen, "The relationship between learner characteristics and student outcomes in a middle school computing course: An exploratory analysis using structural equation modeling," *Journal of Research on Technology in Education*, vol. 51, no. 1, pp. 63–76, 2019.

[27] "Project based learning works," Buck Institute for Education, Last accessed on 05-20-2021. [Online]. Available: "https://my.pblworks.org/"