The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center
ASEE
Portland, OR . June 23 - 26, 2024
Paper ID #44038

Expanding Support for Engaged Remote Student Learning of Internet of Things Concepts and Technology

Dr. David Hicks, Texas A&M University, Kingsville

David Hicks is an Associate Professor in the Electrical Engineering and Computer Science Department at Texas A&M University-Kingsville. Before joining TAMU-K he served as Associate Professor and Department Head at Aalborg University in Esbjerg, Denmark. He has also held positions in research labs in the U.S. as well as Europe, and spent time as a researcher in the software industry.

Dr. Lifford McLauchlan, Texas A&M University, Kingsville

Dr. Lifford McLauchlan is an Associate Professor in the Electrical Engineering and Computer Science Department at Texas A&M University - Kingsville, and has also worked for Raytheon, Microvision, AT&T Bell Labs, and as an ONR Distinguished Summer Faculty at SPAWAR San Diego, CA. He has over 55 publications covering areas such as adaptive and intelligent controls, robotics, an ocean wave energy converter, green technology, education, wireless sensor networks and image processing. He is a co-inventor on 3 US patents related to control systems. Dr. McLauchlan is a member of ASEE and was the 2012-2014 Chair of the Ocean and Marine Engineering Division. He is also a member of IEEE (senior member), SPIE, Eta Kappa Nu, ACES and Tau Beta Pi, and has served on the IEEE Corpus Christi Section Board in various capacities such as Chair, Vice Chair, Secretary and Membership Development Officer. Dr. McLauchlan has received the Dean's Distinguished Service Award twice and the Dean's Outstanding Teaching Award once for the College of Engineering at Texas A&M University-Kingsville.

Dr. Mehrube Mehrubeoglu, Texas A&M University, Corpus Christi

Dr. Mehrubeoglu received her B.S. degree in Electrical Engineering from The University of Texas at Austin. She earned an M.S. degree in Bioengineering and Ph.D. degree in Electrical Engineering from Texas A&M University. She is currently an associate prof

Hemanth Kumar Reddy Bhimavarapu

Hemanth Kumar Reddy is pursuing master's in computer science as well as working on campus as a Graduate Research Assistant in the Electrical Engineering and Computer Science Department at Texas A&M University-Kingsville. Before that he did his Bachelor's in Electronics and Communication Engineering at KL University-Andhra Pradesh, India. He is actively working on developing IOT applications and doing research on U3810A IOT Educational Kit.

Work In Progress: Expanding Support for Engaged Remote Student Learning of Internet of Things Concepts and Technology

Introduction

Internet of Things (IoT) based systems have proven to be effective solutions in a wide variety of application areas. The availability of low-cost versions of the hardware elements that form the basis of these systems, including processor boards, sensors, and communication devices, combined with expanding software support, such as cloud based IoT resources, ensure the popularity of IoT approaches will continue to expand further. Consequently, the inclusion of IoT concepts and technologies is becoming an increasingly important part of many engineering degree programs (e.g., computer engineering, electrical engineering, computer science, and mechanical engineering).

An ongoing NSF funded project at Texas A&M University-Kingsville and Texas A&M University-Corpus Christi has focused on expanding the coverage of IoT concepts and technology that is included in engineering degree programs. A particular emphasis of the project has been on the support of remote engaged student learning. To ensure students are given not only theoretical coverage of IoT concepts, but also receive valuable practical, hands-on experience, a learning toolkit approach has been utilized [8, 10]. Remotely learning students, who do not have easy access to a classroom laboratory in a university setting, are provided with an IoT toolkit they can utilize throughout the semester so they will be able to perform IoT exercises and laboratory assignments in order to fully engage with the material being covered [1, 4].

Supply chain issues encountered at the start of the project limited the availability of equipment that could be included in an IoT learning toolkit. Consequently, the initial IoT toolkit utilized for the project was a very basic one that included a single board computer, sensors, actuators, LEDs, a breadboard, and jumper wires [2]. More recently a more advanced toolkit has been assembled based on a commercially available IoT learning platform. This paper focuses on student exercises that have been developed to support remotely learning students that are utilizing the advanced IoT toolkit.

Background and context

Evolution of the IoT Exercises: A series of exercises have been developed to accommodate the learning toolkits described in this paper and support remotely learning students. As reported previously, the first IoT related exercises that were developed during the project were introductory level ones intended for students utilizing the basic IoT toolkit to assist them in becoming familiar with its components and basic IoT concepts [2]. The introductory exercises were later adapted into a form that is appropriate for the more advanced IoT toolkit [3]. This paper reports on an additional set of exercises that have since been developed to teach more advanced IoT concepts to students utilizing the expanded features and capabilities of the advanced IoT toolkit.

Computer Languages Utilized: While the examples and solutions developed to accompany the first set of exercises developed for the advanced IoT platform were based on the C programming language, the materials created for the more advanced exercises described here utilize the Python language. The intent in using Python is to accommodate a broader variety of student backgrounds. It is anticipated that this will expand the potential use of the toolkit to include students from additional engineering disciplines such as mechanical engineering or industrial engineering where IoT technology plays an increasingly important role [9]. It could also enable the use of the toolkit beyond traditional engineering education programs to include a broader collection of STEM related disciplines.

IoT Toolkit Components: The initial IoT learning toolkit developed during the project was a basic one. Due to supply chain issues at the time, it was based on a simple processor board, the Raspberry Pi [12]. The kit also included sensors, actuators, resistors, LEDs, a breadboard, and jumper wires to connect components together [2]. Once supply chain issues were resolved a more elaborate IoT learning toolkit was developed based on an IoT learning platform, the Keysight U3810A [7]. This learning platform includes an integrated basic processor board, the BeagleBone Green, along with a variety of sensors and components mounted onto a larger circuit board. The U3810A IoT learning platform is pictured in Figure 1. In addition to the U3810A and its integrated basic processor board, the advanced learning toolkit includes jumper wires to make connections among its components. A breadboard is also included to enable students to incorporate additional sensors, actuators, etc. if needed in the development of an IoT solution.

Figure 1. The U3810A learning platform.

IoT Exercises

The computer language used in the development of the most recent exercises for the advanced IoT kit was Python. A series of seven exercises have been developed to date, designed to help students learn the basics of using the advanced IoT toolkit and the learning platform upon which it is based. The exercises were designed to be completed by students in sequence. To support the goal of broadening the applicability of the exercises beyond just those students studying in the electrical engineering, computer engineering, or computer science disciplines, the exercises start at a basic level and iteratively incorporate additional functionality and components of the IoT learning platform as they progress [5].

To utilize the learning platform of the advanced IoT kit, students are first required to connect a desktop or laptop computer to its integrated basic processor board (BeagleBone Green) via a USB cable. An SSH client application, such as PuTTY, can then be used to log in to the processor board from the laptop to create, edit, and debug programs [6]. As these programs will run on the integrated processor board they have ready access to the various components of the U3810A environment and can utilize them in the development of IoT solutions. Students can run their programs on the processor board utilizing the SSH client connection and view any outputs generated via their login shell or on a component of the IoT learning platform, such as an LCD display, depending on which devices have been included in an IoT solution.

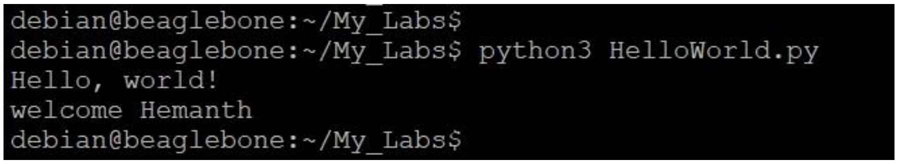
The seven Python-based IoT exercises that have been developed for the advanced IoT toolkit are summarized in Table 1. Having students utilize the advanced IoT toolkit based on the U3810A board to complete the exercises enables them to easily incorporate a variety of its available integrated input and output devices into their solutions. These include utilizing an LCD display for output and sampling input from buttons and an analog temperature sensor. Though equivalent IoT exercises could be designed for a basic IoT toolkit utilizing a simple processor board, students would be required to physically connect similar input or output devices in order to include them in an IoT solution, significantly complicating exercise tasks intended for novice students studying remotely. The remainder of this section describes each of the exercises in more detail.

Table 1. Python-based IoT Exercises for Advanced IoT Learning Toolkit

Exercise	Topics Covered
1. Introduction and orientation	Connecting laptop to processor board and orientation to advanced IoT platform.
2. Using LCD display	Learning to configure a circuit board bus and how to access LCD display.
3. Integrating additional input devices	Learning how to access and monitor button inputs located on IoT platform circuit board.
4. Integrating analog temperature sensor	Learning how to access and monitor an analog temperature sensor of the IoT platform.
5. Dual display of temperature readings	Displaying temperature readings in two places: the SSH shell output window and LCD display on IoT platform circuit board.
6. Cloud-based IoT Resources	Integrating cloud-based resources into an IoT solution to store, display, and graph temperature readings.

7. Introducing the XBee communication module	Learning how to connect, configure, and control a XBee communication module as part of an IoT solution.

The first exercise is primarily software focused as it assists students in learning how to connect to and interact with the integrated BeagleBone Green processor board in order to create and run Python programs. It instructs students on the needed connections including a USB connection from their laptop or desktop machine to the processor board and a power connection for the learning platform circuit board. Students are then instructed on how to open an SSH connection from their laptop and create and run Python programs. The first program students are asked to create is an IoT version of the classic HelloWorld program that will run on the integrated processor board and display the familiar “Hello World” output message in the SSH shell window. Figure 2 illustrates the HelloWorld program being invoked in a PuTTY window and the resultant output message.



```
debian@beaglebone:~/My_Labs$
debian@beaglebone:~/My_Labs$ python3 HelloWorld.py
Hello, world!
welcome Hemanth
debian@beaglebone:~/My_Labs$
```

Figure 2. Running the HelloWorld program.

The second exercise builds directly on the first and integrates an output component available on the IoT learning platform circuit board. In addition to displaying the output of the HelloWorld program in the SSH shell window students are asked to also output the message to an LCD display located on the circuit board. Students are provided with the memory mapped address of the LCD display and instructions on how to configure a bus of the learning platform to establish access to the display. They are then given instructions on how to initialize and send characters for a message to be written to the LCD display. Figure 3 shows the output of the HelloWorld program being displayed on the LCD of the learning platform.

Figure 3. Program output on LCD display.

The third exercise extends the running theme further by asking students to incorporate another component of the IoT learning platform, this time for input purposes. Students are instructed to write a program that will monitor a set of four buttons located on the learning platform and, when one of the buttons is pressed, display an appropriate message to both the SSH shell window and the LCD display. Instructions are provided on how to access and monitor the status of the buttons utilizing the GPIO (general purpose input and output) pins of the basic processor board. Students must then create an appropriate program control structure in their Python code to continually sample and display the status of the buttons. In Figure 4 a message can be seen in the

LCD display of the learning platform indicating that button 4 was pressed. The four input buttons can also be seen to the right side of the LCD display (labeled B1, B2, B3, and B4).

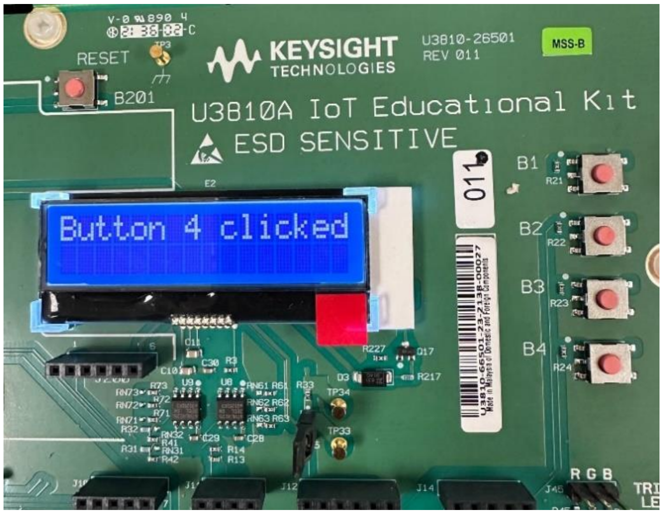
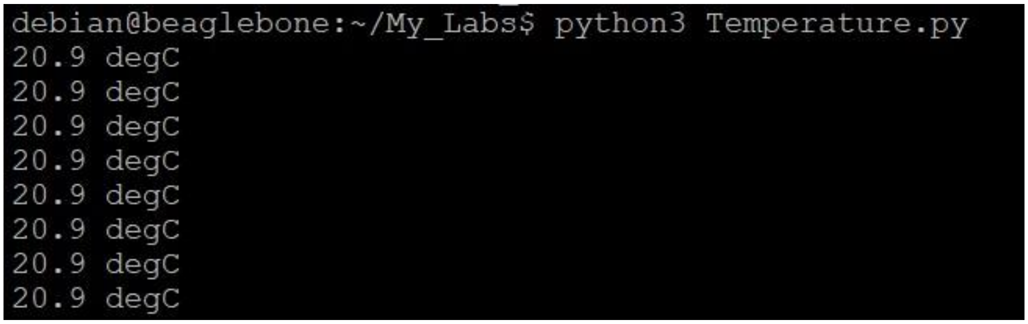


Figure 4. Button status shown in LCD display.

In the fourth exercise students are introduced to another IoT related input device, an analog temperature sensor located on the IoT learning platform. To utilize the sensor students are first required to use jumper wires to connect it to a bus on the IoT learning platform. Next, they are given information on how to address and query the sensor to obtain the current temperature reading. They are then asked to write a program that will continually sample the temperature reading, convert it to a value in degrees Celsius, and then output it to an SSH shell window. In Figure 5 a PuTTY window is shown running a solution program for the exercise that continually samples and outputs temperature readings.



```
debian@beaglebone:~/My_Labs$ python3 Temperature.py
20.9 degC
20.9 degC
20.9 degC
20.9 degC
20.9 degC
20.9 degC
20.9 degC
20.9 degC
```

Figure 5. Temperature readings output to a PuTTY window.

The fifth exercise builds directly on the fourth. Students are again tasked with taking continual temperature readings from an analog temperature sensor on the learning platform board. They are instructed to convert the readings to values in degrees Celsius, and to output those values to the LCD display of the learning platform circuit board as well as the SSH shell window. Figure 6 shows a temperature value in degrees Celsius being displayed in the LCD output of the learning platform. Also in Figure 6 the jumper wires can be seen that were used to connect the analog temperature sensor (lower left of figure) to a bus on the IoT learning platform (middle of figure).

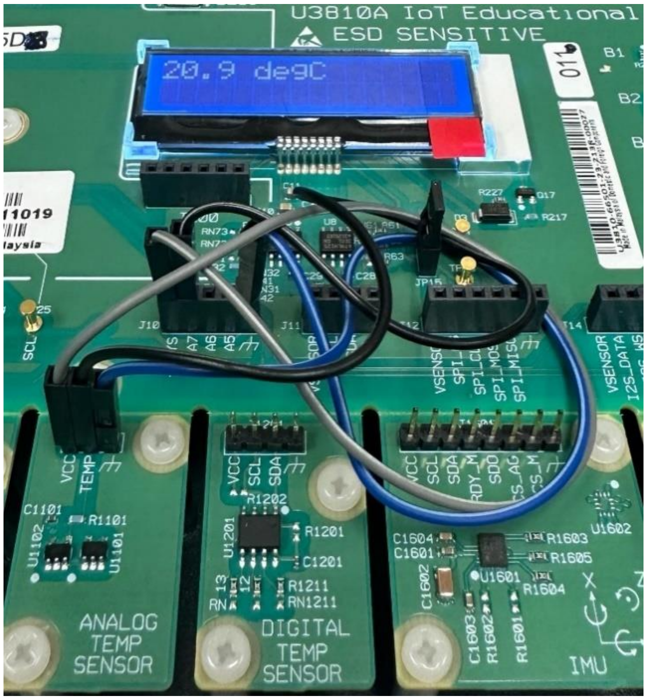
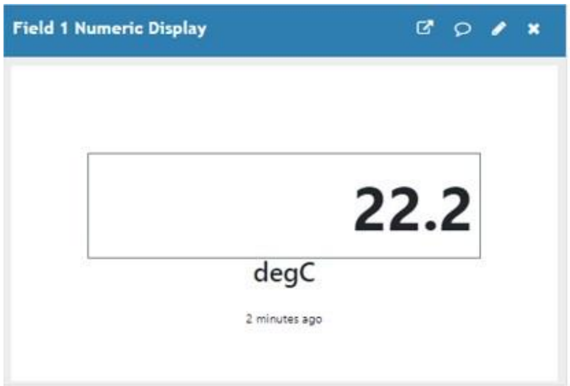


Figure 6. Temperature values output to LCD display.

The sixth exercise builds on the previous two to provide students with an opportunity to learn about cloud-based IoT resources. Students are first provided with information about the general availability of cloud-based resources relevant to the design and development of IoT solutions. After studying the information students are instructed to create a free account with a specific cloud-based IoT resource provider, such as ThingSpeak [11]. The exercise next asks students to modify their solution to the previous exercise to integrate a cloud-based resource. Specifically, they are instructed to send the converted temperature values read from the analog temperature sensor to their new cloud-based IoT resource provider account (e.g., ThingSpeak channel). The data values should be stored for analysis on their cloud-based account.

Students are also instructed to configure their cloud-based account to process and display the received temperature readings in local windows on their laptop or desktop machine in two different formats. First, the current Celsius temperature reading (the most recently received value) should be displayed in an output window in digital form. This demonstrates the successful transmission of temperature values to their cloud-based account. Secondly, a graph should be displayed illustrating the temperature readings that have been received at the cloud-based account over time. This demonstrates the ability to store and utilize the temperature values to create a basic data visualization. The tasks of this exercise provide students with an introductory look at some of the IoT related cloud-based resources and the role they can play in an IoT solution. Figure 7 shows the most recent Celsius temperature reading received by a user's ThingSpeak account being displayed in a local output window on their laptop.



22.2

degC

2 minutes ago

Figure 7. Local display of temperature readings stored in cloud-based account.

In Figure 8 a graph is shown plotting the temperature readings received by a user's ThingSpeak account over time. The values are analyzed and the graph is produced by ThingSpeak and then displayed in a local output window on the user's laptop.

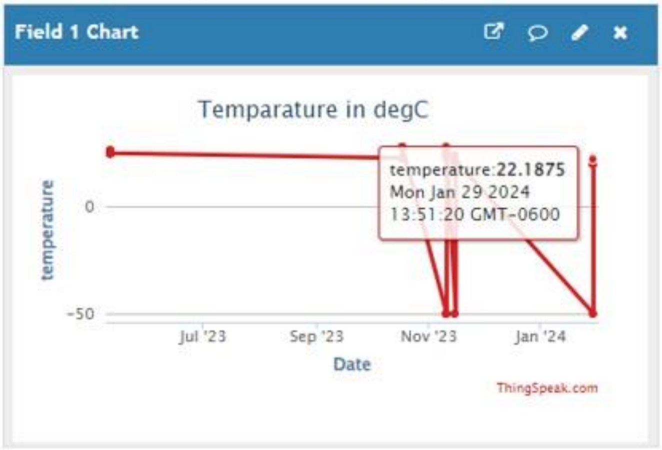
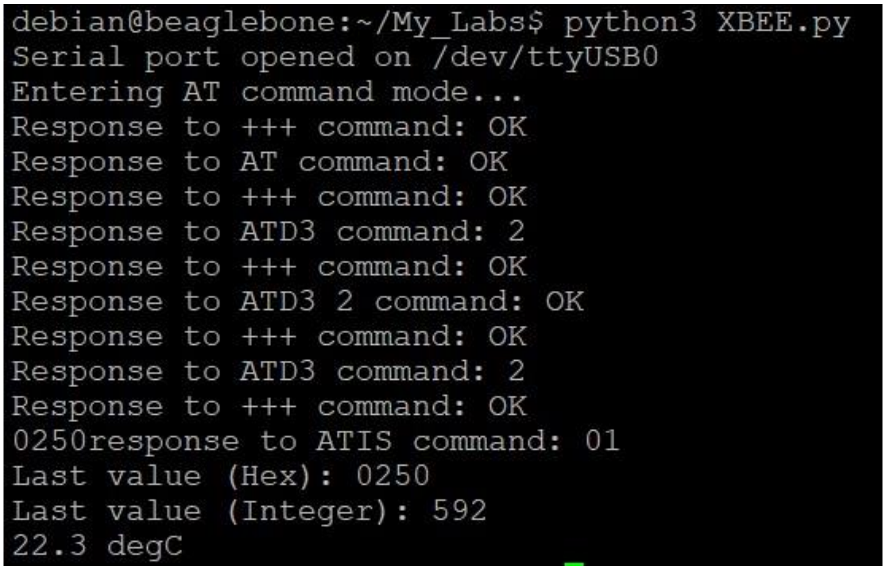


Figure 8. Graph of temperature readings received over time.

The seventh exercise introduces students to another component of the IoT learning platform, the XBee communication module. Students are first provided with information about the communication module and its capabilities. They are then instructed to utilize jumper wires to connect an analog temperature sensor to a XBee communication module that is located on the IoT learning platform. A connection is also needed from a USB port on the integrated processor board of the IoT learning platform (BeagleBone Green) to the XBee module. Students are then instructed on how to utilize the XBee AT command set to configure and control the communication module. The programming task of this exercise is to utilize the analog to digital conversion capability of the XBee module to convert the temperature reading from the analog temperature sensor into a digital value. It should then be converted to a value in degrees Celsius, and output to an SSH shell window. The tasks of this exercise introduce students to the XBee module, the commands used to configure and control it, and its potential uses in an IoT solution. Figure 9 shows the output of a program running on the integrated processor board. It utilizes the analog to digital conversion facility of the XBee module and then finally outputs a temperature value in degrees Celsius.



```
debian@beaglebone:~/My_Labs$ python3 XBEE.py
Serial port opened on /dev/ttyUSB0
Entering AT command mode...
Response to +++ command: OK
Response to AT command: OK
Response to +++ command: OK
Response to ATD3 command: 2
Response to +++ command: OK
Response to ATD3 2 command: OK
Response to +++ command: OK
Response to ATD3 command: 2
Response to +++ command: OK
0250response to ATIS command: 01
Last value (Hex): 0250
Last value (Integer): 592
22.3 degC
```

Figure 9. Utilizing XBee module to convert analog reading to digital and calculate temperature value.

IoT Kit Selection Factors and Tradeoffs

There are naturally a variety of tradeoffs and factors that must be considered in selecting an appropriate IoT toolkit for a class (or lab). Cost is one of the most significant of those factors. While the advanced IoT toolkit does facilitate more easily including a variety of input and output devices into an IoT solution than would be possible with a basic toolkit, especially for novice students, the IoT learning platform upon which it is based can increase the price of the overall toolkit considerably. For the project described in this paper, emphasis was placed on supporting remotely learning students from a variety of STEM fields. This increased the importance of the "ease of use" of the toolkit, especially for novice students. In other scenarios such as deploying an IoT kit in a class or lab that is taught in person and that is attended by computer engineering or computer science students, a toolkit that is based on a basic processor board such as a Raspberry Pi or even a micro-controller such as an Arduino board might offer an appropriate and more affordable alternative.

Accessibility is another important factor. IoT learning platforms are invariably more intricate and include a larger number of components than basic processor boards. This can make them more susceptible to supply chain issues or manufacturing delays. Basing an IoT kit on such a platform can delay its availability for students, as experience at the start of this project.

Current Status and Deployment Plan

An exercise manual to accompany the advanced IoT learning toolkit is in the process of being completed. The manual will include descriptions of the seven exercises detailed in this paper. It will also include supplemental material to describe any new IoT components an exercise introduces to students as they progress through the series of exercises. It will also provide information on any necessary connections students should make along with configuration information for components and details on how they can be accessed through software.

The planned initial deployment environment for the advanced IoT toolkits are the capstone senior design projects conducted within engineering and computer science related departments at the two universities where this project is being conducted. Student projects in these courses frequently incorporate some degree of IoT technology. Teams working on such projects will be offered the use of an advanced IoT learning toolkit for initial use as a learning platform. Subsequently the IoT toolkit can serve as a prototype environment for the students as they design and implement their IoT solutions.

Pre- and post-project surveys will be used to gather data on the efficacy of the learning toolkits in teaching students about IoT concepts and technology. Questions will also be included to assess the impact of the IoT toolkits on students' ability to recognize opportunities for exploiting IoT related technology in their capstone projects.

Conclusion

A collection of exercises has been developed to support remote engaged student learning of IoT concepts and technology. The exercises are intended to support students that are utilizing an advanced IoT toolkit to learn about IoT concepts and to conduct project work. To broaden the applicability of the exercises they are based on the Python programming language. The initial deployment environment for the advanced IoT toolkit and accompanying exercises will be in capstone senior design courses. Surveys are planned to collect information to be used in assessing the efficacy of the IoT toolkits and exercises.

Acknowledgement and Disclaimer

This material is based upon work supported by the National Science Foundation through Grant No. 2044255. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] N. E. Cagiltay, E. Aydin, C. C. Aydin, A. Kara and M. Alexandru, "Seven Principles of Instructional Content Design for a Remote Laboratory: A Case Study on ERRL," in IEEE Transactions on Education, vol. 54, no. 2, pp. 320-327, May 2011, doi: 10.1109/TE.2010.2058115.

[2] McLauchlan, L., & Hicks, D., & Mehrubeoglu, M., & Bhimavarapu, H. K. R. (2023, June), Enabling Remote Student Learning of IoT Technologies Paper presented at 2023 ASEE Annual Conference & Exposition, Baltimore , Maryland. https://peer.asee.org/43273

[3] Hicks, D. McLauchlan, L., Mehrubeoglu M., and Bhimavarpu, H. K. R, "Expanding Remote Student Learning-Internet of Things Applications and Exercises," 2023 IEEE Frontiers in Education Conference (FIE), College Station, TX, USA, 2023, pp. 1-8, doi: 10.1109/FIE58773.2023.10343065.

[4] Feisel, L.D. and Rosa, A.J. (2005), The Role of the Laboratory in Undergraduate Engineering Education. Journal of Engineering Education, 94: 121-130. https://doi.org/10.1002/j.2168-9830.2005.tb00833.x

[5] B. Adams, S. Jorgensen, A. Arce-Trigatti, P. Arce (2020) Innovative Curriculum Design for Enhancing Learning in Engineering Education: The Strategies, Principles and Challenges of an Inquiry-Guided Laboratory, INTED2020 Proceedings, pp. 8127-8135.

[6] PuTTY Project. putty.org (Accessed: February 6, 2024).

[7] Keysight U3810A Advanced IoT Teaching Solution Getting Started Guide, May 2021, [online]. Available at: https://www.keysight.com/us/en/assets/9018-70077/quick-start-guides/9018-70077.pdf. (Accessed: February 6, 2024).

[8] Chan-in, Attapan and Sipitakiat, Arnan. 2021. Designing an Educational Internet of Things Toolbox: A case-study of a learner-centric tool design. In Proceedings of the FabLearn 2020 – 9th Annual Conference on Maker Education (FabLearn '20). Association for Computing Machinery, New York, NY, USA.

[9] O. Ezenwoye, "What Language? - The Choice of an Introductory Programming Language," 2018 IEEE Frontiers in Education Conference (FIE), San Jose, CA, USA, 2018, pp. 1-8, doi: 10.1109/FIE.2018.8658592.

[10] De Roeck, Dries; Tanghe, Juergen; Jacoby, Alexis; Moons, Ingrid; and Slegers, Karin. 2019. Ideas of Things: The IOT Design Kit. In Companion Publication of the 2019 on Designing Interactive Systems Conference 2019 Companion (DIS '19 Companion). Association for Computing Machinery.

[11] IoT Analytics – ThingSpeak Internet of Things, [online] Available: https://thingspeak.com/pages /education. (Accessed: February 6, 2024).

[12] Raspberry Pi Official Documentation Site, [online]. Available at: https://www.raspberrypi.com/documentation/. (Accessed: March 29, 2024).