A Comprehensive Approach to Educating Students About the Internet-of-Things

Samuel J. Dickerson
Department of Electrical and Computer Engineering
Swanson School of Engineering
University of Pittsburgh
Pittsburgh, PA, USA
dickerson@pitt.edu

***Abstract*—The Internet-of-Things (IoT) is an new area where everyday objects are embedded with intelligence and the ability to transmit and receive information. The arrival of this emerging topic presents a pressing educational need. In this paper, a new, innovative course on IoT is presented. The pedagogical approach is to guide students in developing a conceptual "toolbox" of necessary IoT skills centered on the pillars of IoT systems: sensors, embedding computing, wireless networking and cloud-computing. Fundamental topics associated with these areas are taught to students all in one, comprehensive, hands-on course rather than focusing solely on one aspect. Results from the course are presented in the form of direct assessment of student projects. The results show that the course curriculum is effective in reaching the desired student outcomes. Recommendations for improving future offerings are given.**

Keywords—Internet-of-Things; Electrical and Computer Engineering Education; Embedded Systems; Networking; Cloud-Computing;

I. Introduction

The Internet-of-Things (IoT) is an emerging area that combines technical areas spanning a wide range, from sensing and embedded computing, to networking protocols and sophisticated cloud-computing services. The breadth of technical skills required to embody these areas in a single electronic device is in contrast with the sometimes narrow focus of Electrical and Computer Engineering programs. Many refer to IoT as the center of the next industrial revolution. Thus, preparing undergraduate students to become contributors in this new field is a pressing educational challenge. While IoT educational materials and courses are now starting to appear, much of what is available tends to focus on just one particular aspect of IoT rather than providing a comprehensive IoT skillset. In educating students about IoT, some questions that arise are: What core skills do students need to learn in order to have a sufficiently equipped 'toolbox' of IoT skills? What makes a course on IoT something new and different? How can we differentiate an "IoT" course from other commonly offered courses (e.g. embedded systems) that may seemingly have a lot of overlap?

In this work, efforts to address these IoT educational challenges by way of a newly developed, innovative course on the Internet-of-Things is presented. The course is an elective, targeting upper-class electrical and computer engineering students. In this course students participate in a series of hands-on laboratory assignments using an embedded computing platform along with various wireless microcontrollers, exposing them to all of the primary aspects of IoT in a single semester long experience.

In this paper, we report on the offering of a newly developed, IoT specific course. The paper is organized as follows: In the next section we give an overview on the current state of IoT education. That is followed by a detailed description of the course curriculum design. Results are presented in the form of direct assessments of student projects. The paper concludes with lessons learned, suggestions for improving future offerings and recommended guidelines for other educators who wish to offer IoT courses.

II. IoT in Education

IoT in education takes on many forms. Courses on IoT can focus on technological, engineering aspects or even focus on the application area and not as much the technology [1]. Course offerings that focus on IoT technology have become popular with the advent of low-cost microcontroller platforms [2,3]. This has even led many popular online venues to teach IoT related material [4]. However, IoT courses and materials that make the microcontroller the principal focus have the disadvantage that many low-level details are abstracted away. Courses of this type are excellent for prototyping and "making" but are not suitable for an advanced electrical or computer engineering student. In some reports of IoT courses, the material reported on is suitably advanced but does not cover a wide enough range of material. As a result, such courses have substantial overlap with other common engineering courses and go in depth in the areas of embedded systems [5] ,computer networking [6] and cloud-computing [7] and at the expense of breadth. What makes an educational offering on IoT unique is that it combines many of these areas and more, therefore breadth is a critical component. It has been proposed by some that the core IoT educational pillars broadly includes sensing, communication of data, aggregation of data, and decision making based on data [8]. In this work, we specifically define key technologies that should be included in the design of an Iot course and provide a detailed curriculum design.

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

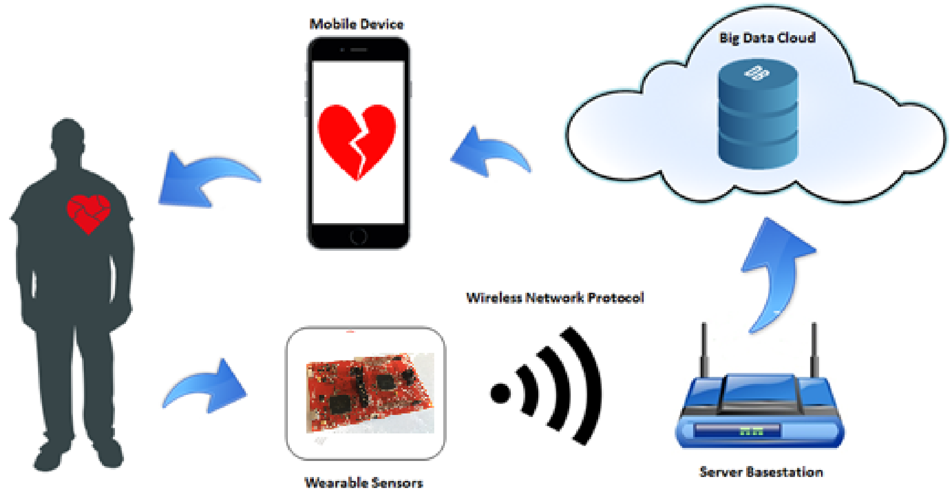

Figure 1. Data flow for an example IoT device, a wearable health monitor

III. IoT Curriculum Design

Since IoT is broad and has numerous application areas, an IoT course can be structured in many different ways and for a wide variety of audiences. The course offering presented here has been designed for electrical and computer engineering undergraduate students, therefore the educational focus is on creation the of underlying IoT technologies. The key challenge in creating an IoT course for electrical and computer engineering students is making sure that it is presented in a way that shows minimizes overlap with other common courses such as embedded systems and networking. To achieve this goal, four core technologies that are embodied in most IoT systems:

- **Embedded Computing**
- **Sensors**
- **Networking**
- **Cloud Computing**

Figure 1. illustrates these aspects in the context of a common, modern IoT device, a wearable device for health care monitoring. This device is composed of a microcontroller that interfaces to multiple sensors and has the ability to acquire physiological data measurements such as heart rate, temperature or tremors. That sensor data is then processed and transmitted wirelessly over a network to a server which then pushes the data to the internet "Cloud". At that point, "Big Data" analytics are used to assess the state of the person and any necessary warnings or recommended courses of action are then transmitted back to the person. Data having to traverse this loop, originating from a sensor, going to a microcontroller, through network and then to the cloud (and back) is what makes IoT new and exciting. The educational goal is not to cover all aspects of this loop, but rather key features that prepare students.

The course was designed to have two distinct components: Lab assignments and a course project. The lab assignments are structured, prescribed labs that have directed learning objectives. The purpose of these assignments is to equip students with basic tools and skills needed to create IoT designs. The semester ends with a term project were students are tasked with conceiving and implementing their very own IoT innovation.

In the sections that follow, the hardware platform selected for instruction is presented and then the approach taken with lab assignments built around the four core areas is described. The concluding sections of this paper presents results from the open-ended student projects. These projects are directly assessed in terms of the course learning objectives.

A. Hardware Platform for Instruction

Of critical importance to the course and learning outcomes is the prototyping platform provided to the students. In designing this course many options were considered as there are a wide variety of options to choose from. Currently, platforms such as the Arduino [10] and the Raspberry Pi [11] are very

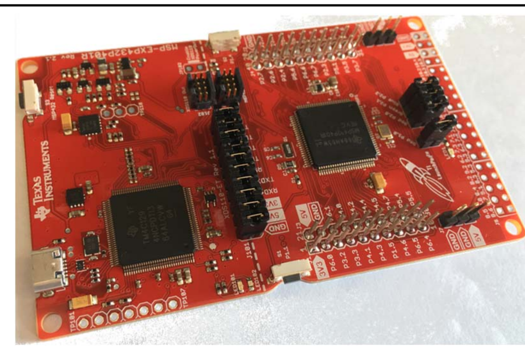

Figure 2. Microcontroller platform used for instruction, the MSP-432 Launchpad by Texas Instruments [9]

popular due to their ease of use. However, from the standpoint of educating electrical and computer engineering students, such platforms can at times provide so many layers of abstraction that it hinders learning of important low-level hardware/software concepts (e.g. sensor interfacing protocols). Conversely at the same time, the ability to have such abstraction available is useful when students need dedicate time on advanced topics that focus on the application more than the technology (e.g. cloud-computing). Therefore, since the goal was to create a comprehensive IoT course, a hardware platform with facilities and support for both types of usage models was selected, the MSP-432 LaunchPad by Texas Instruments (figure 2). The platform supports development environments that are geared towards low-level programming and debugging such as Code Composer Studio or Keil that allow for C/Assembly programming, viewing register values and memory maps). The platform also supports a high-level, abstracted development environment (i.e. Energia [12])) that is identical to the easy to use Arduino environment.

Having this type of flexibility is important to the pedagogical approach of the class. The pedagogical approach is to teach students about the Internet-of-Things from the "Bottom-Up", starting with individual sensors operating in low-power computing environments and ending in the "Cloud". One of the objectives of this course is to remove abstraction where possible enabling students fully appreciate underlying theories and operating principles. As the laboratories progress in complexity and difficulty, concepts and ideas from preceding assignments are re-used, but the concepts are reintroduced using devices and components that allow for them to be presented in their abstracted forms.

B. Embedded Computing

The class was designed such that no background in embedded systems is assumed. Therefore, the first topics taught are the basics of programming for embedded devices [13]. Students are given an initial series of assignments where they must use assembly language to access various microcontroller resources. Assembly language is primarily used to reinforce concepts relating to memory-mapped I/O and to emphasize the challenges of programming for a resource-constrained device. After these initial assignments, students are provided with further assignments that instruct them on the basics of General-Purpose I/O, interrupts and microcontroller timing resources. For these assignments students are allowed to use the C programming language.

C. Sensors and Embedded Interfacing

The next section of the course focuses on the basics of sensors. There exists a wide variety of sensors that are readily available and could be used for instruction. However, an exhaustive overview of sensors is neither possible nor useful from an educational standpoint. The approach taken in this course is to give students experience with a small variety of carefully selected sensors that all vary in how they are interfaced to. For the first assignment, students are tasked with using an ultrasonic rangefinder as input to a traffic light controller (figure 3). This sensor serves as a good introduction because the input/output states are binary and reading information from it only requires the skills learned in the embedded computing portion of the course, in particular the GPIO interfaces and microcontroller facilities for controlling time. This particular device also serves as a good introduction to sensing because it requires students design and construct voltage-level shifting circuitry that allows for compatibility between the sensor and microcontroller. This is also a good learning experience for

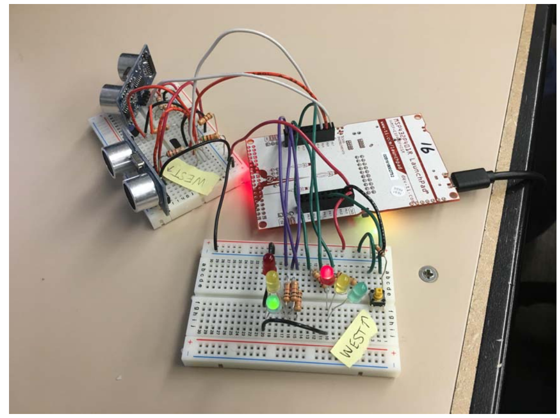

Figure 3. Example student assignment making use of ultrasonic distance sensors. Assignment requires students to construct voltage level shifting circuitry in order to properly interface the circuit.

students as often sensors used in IoT applications require additional peripheral circuitry for proper operation.

The subsequent sensor-focused assignments have students collect temperature data in two different ways. First, students use an analog temperature sensor that must be sampled. Afterwards, students use a digital temperature sensor that communicates over an I^2C serial interface. These assignments allow students to examine the tradeoffs (e.g. power consumption, complexity, cost, etc) in using analog versus digital sensors in IoT devices. Furthermore, since power consumption is of the utmost importance in battery-operated IoT devices, a significant portion of time is spent on examining digital serial interfaces in detail. The digital sensor interfacing assignments are good examples of where students can learn more by initially not relying on simplified, easy to use libraries. Having students access microcontroller registers to control every aspect of the serial data interchange gives them deep insight into the protocol behind the interaction. Having developed that insight is valuable to learning to design for IoT devices for commercial applications

D. Networking

Machine-to-Machine (M2M) communications is a principle feature of IoT technologies and an important part of any IoT course. M2M communications are not tied to or defined by any one particular protocol. In this course, M2M is taught to students by way of a protocol that was developed specifically for IoT, Message Queue Telemetry Transport otherwise known as MQTT [14]. MQTT was designed to be lightweight and is ideal for IoT due to its small code footprint that reduces use of embedded computing resources. Students connect their MSP-432 prototyping platform to the internet via an 802.11 wireless network processor (CC3100 development board by Texas

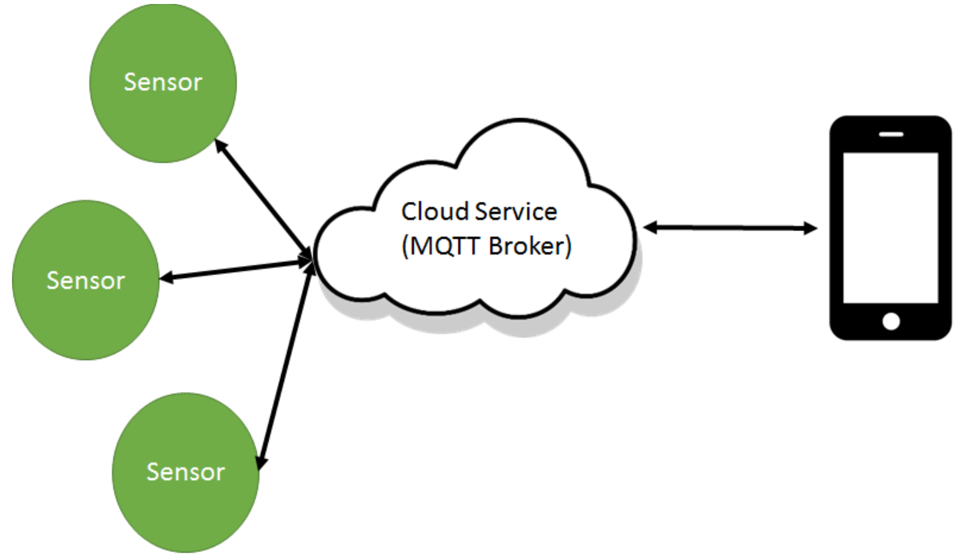

Figure 4. MQTT Publish-Suscribe Network Architecture

Instruments [15]). MQTT runs on top of TCP/IP and has a publish-subscribe architecture[14] (figure 4). At this stage of the course, students are transitioned from using low-level C commands to using the higher-level, abstracted software development environment. This enables students to focus on various aspects of the MQTT protocol rather than being bogged down with the potential difficulties of trying connect a device to the Internet via low-level software commands. In this way, students experience MQTT's suitability for IoT applications and see the advantages to using an MQTT broker as a central point for publishing and subscribing to messages generated by connected devices, thereby minimizing parameters critical to IoT applications such as power consumption and network bandwidth.

E. Cloud Computing

MQTT also serves as a bridge between the embedded world and cloud computing, the final lab portion of the course. Many modern cloud-computing services (e.g. IBM Bluemix [16], Amazon AWS [17]) use MQTT as the protocol of choice for interfacing embedded devices to their cloud computing services. One of the most powerful aspects of the Internet-Of-Things is that once we have a device with the ability to acquire, process and transmit data that data can be sent to the "Cloud" for further analysis. Once in the internet cloud, we can carry out even more sophisticated analyses of the data, as computation is no longer bounded by the limitations of the embedded device. Moreover, vast amounts additional data can be taken into consideration such as information from multiple sensors or historical records and databases. With the use of those data sources, we can translate the original sensed data into information that is meaningful to the end-user.

Students are provided with modules that teach them about various cloud-computing paradigms such as Platform as a Service (PaaS) and Software as a Service (SaaS) and how to make use of specific cloud-computing APIs. In particular, students are shown examples of cloud-computing services being used to carry out complex functions such as textual analysis, speech analysis, and machine learning based big-data analytics. The development environment used for the cloud computing portion of the course is Node-Red[18]. Node-Red is a graphical programming environment that allows students to create JavaScript functions and call Cloud-Computing APIs using graphical schematic entry. Figure 5 shows an example student project that made use of cloud-computing services in conjunction with sensor data. The team used a combination of temperature data and the sentiment discerned from a particular celebrity's most recent tweets to determine corresponding notifications and responses.

IV. Results From Student Projects

At this stage of the course, students have been exposed to all of the previously mentioned core aspects of IoT, embedded computing, sensors, networking and the cloud by way of prescribed labs. It is at this junction that the course transitions from being lab-based to project based. Students are asked to form larger teams (3-4 students), conceive and then implement their very own IoT device.

The top of figure 6 shows the finalized project ideas and a short description of them. The bottom of figure 6 shows the rubric used to assess the learning outcomes from this course. There were four performance indicators used to assess the student's ability to conceive and implement an IoT device. The first indicator was the student's usage of the sensors. This indicator ranges from exemplary, students that made use of multiple sensors, down to unsatisfactory, students that did not make use of sensor or who's sensor usage was trivial. The next performance indicator related to the students efficient use of embedded computing resources. A student team who demonstrated exemplary performance in this aspect would have written customized functions to read and transmit data in such a way that improves some metric of interest to the embedded computing environment (e.g. bandwidth or power). The final two performance indicators relate to networking and cloud-

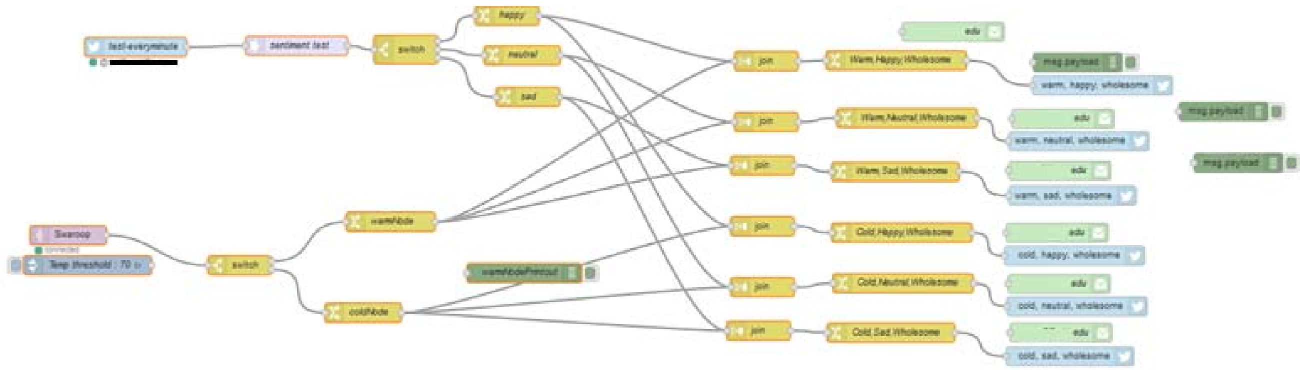

Figure 5. Example Student Node-Red schematic, enabling sensor data to be combined with advanced textual analysis of celebrity tweets

Team	IoT Project	Short Description
1	Coffee Pot	Coffee pot Tweets information on state of coffee brewing
2	Plant	Plant Tweets updates when watering is needed
3	Rhythm Game	Real-time analysis of button-pressing rhythm game
4	Water Bottle	Connected water bottle tracks water consumption statistics
5	Sensor Network	Distributed sensor network to analyzes room heating efficiency
6	Food Scale	Web-connected scale for analysis of refrigerator inventory
7	Seat Finder	Internet-enabled seats that track usage history
8	Basketball	Free throw shooting game collects/analyze accuracy statistics

	Unsatisfactory	Developing	Satisfactory	Exemplary
Sensors	Did not make use of sensors or sensor usage trivial	Made use of analog sensors	Utilized sensor that required digital serial interface	Made use of multiple types of sensors
Embedded Computing	No customizations made to embedded software	Wrote custom functions to read/transmit data	Preprocessed data using embedded platform prior to transmitting	Wrote low-level functions to efficiently use resources
Networking	Unable to transmit data to Internet	Made use of non-IoT specific protocols to transmit data	Able to use MQTT to publish data to a cloud-connected broker	Made use of multiple network protocols
Cloud-Computing	Unable to make use of cloud service	Able to display data to cloud service	Made use of at least one advanced Cloud API (e.g. textual analysis)	Included feedback from cloud back to embedded device

Figure 6. Top shows a brief listing of student conceived/designed final projects. Bottom shows rubric used to assess projects

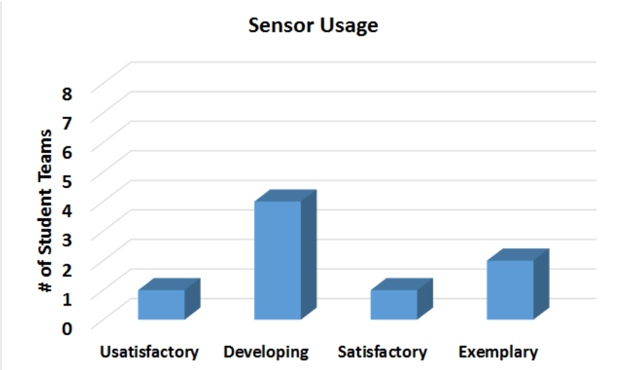

Figure 7. Sensor Usage assessment results

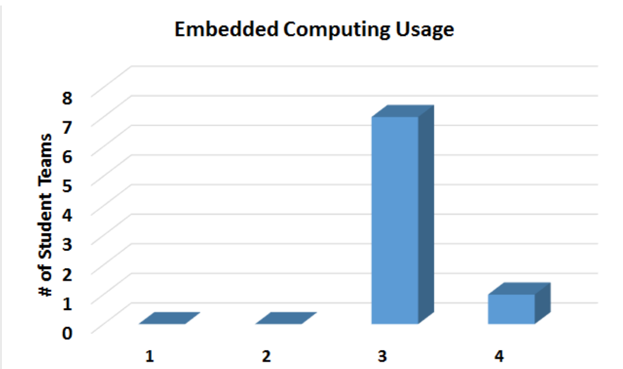

Figure 8. Embedded Computing Usage assessment results

computing. Exemplary students in these areas would have used MQTT, or some other IoT centric protocol to transmit data, made use of more than one network protocol en route to a cloud service, and then once in the cloud, would have used an advanced cloud computing API call that results in some physical action back on the embedded device.

A. Evaluation of Sensor Assessment

Figure 7 shows the result of the assessment of sensor usage in the final student projects. Exemplary projects would have made use of multiple sensors, including ones with a digital serial interface. The results shows that 37.5% of students were assessed at performing at a satisfactory or above level. The instructor feels that this is due to the majority of students basing their projects on the selection of a particular sensor of interest and not as much on applications that can make use of a wide variety of multi-modal sensor data. To improve this aspect, in the future the use of multiple sensors will be a project requirement. This requirement will better reflect the scenario seen in many modern Iot deployments.

B. Evaluation of Embedded Computing Assessment

Figure 8 shows the result of the assessment of embedded computing usage in the final student projects. Exemplary projects would have created their own low-level functions to optimize usage of some embedded computing resource such as network bandwidth or power. Most students were satisfactory in this regard, meaning that they performed some sort of preprocessing operation on the embedded side (e.g. filtering of data prior to sending to the cloud). The results shows that overall 100% of students were assessed at performing at a satisfactory or above level. The instructor feels that this shows that the coverage of embedded computing is sufficient for the course. This assessment also indicates that it may be possible to reduce the time and effort spent on embedded computing so as to leave more time for the other aspects.

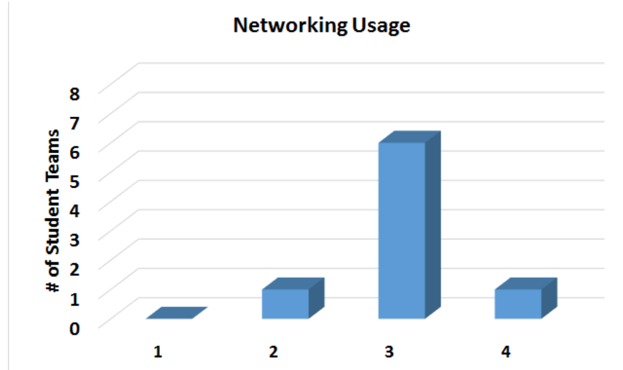

Figure 9. Networking Usage assesment results

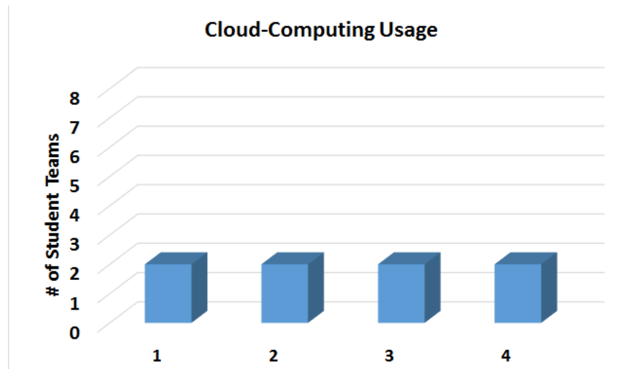

Figure 10. Cloud-Computing Usage assesment results

C. Evaluation of Networking Assessment

Figure 9 shows the result of the assessment of networking usage in the final student projects. Satisfactory projects would have made use of an IoT-centric networking protocol, such as MQTT in their project while exemplary projects would have use that in addition to multiple other protocols (e.g. Bluetooth to 802.11 then published via MQTT). A total of 87.5% of the students were assessed at a satisfactory or above level in this aspect. As with the sensors, in the future the final project requirements could be expanded as to make the use of multiple network protocols prior to transmission of data a requirement.

D. *Evaluation of Cloud-Computing Assessment*

Figure 10 shows the result of the assessment of cloud-computing usage in the final student projects. Satisfactory projects would have made use of at least one advanced cloud-computing API in their project (e.g. machine learning, textual analysis, etc). Exemplary projects would have used the results of analysis of that data to perform some sort of corresponding action back on the embedded device, completing the feeback loop. The assement shows that 50% of students were evaluated at satisfactory or above in this aspect. The recommendation to improve this outcome in the future is include additional lab assignments during the structured lab portion of the course that provide more examples of using these services so that students have the experience to use them in a more extensive way.

V. Summary and Conclusions

In summary, a new, innovative course on the Internet-of-Things is presented. The course is designed to be comprehensive, exposing students to the primary aspects of IoT, embedded computing, sensors, networking and cloud computing. The course is divided into two parts, hands on labs that provide necessary skills and then an open-ended semester project. The final projects were assessed and the results show that the course achieved varying levels of success in meeting the desired learning outcomes. Specifically, the results show that students are able to perform well at using embedded computing and networking in the context of IoT. There is room for improvement in student usage and understanding of sensors as well as cloud computing. It is recommend that these outcomes can be improved by incorporating more hands-on labs prior to projects and by including more requirements in the projects that will guide students towards more IoT centric solutions.

References

[1] Ali, Maqbool, et al. "IoTFLiP: IoT-based Flip Learning Platform for Medical Education." Digital Communications and Networks (2017).

[2] Ray, Santosh, and Ali Al Dhaheri. "Using Single Board Computers in University Education: A Case Study." World Conference on Information Systems and Technologies. Springer, Cham, 2017.I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[3] Lee, Mark JW. "Guest Editorial Special Issue on Maker Learning in Computer Science and Engineering Education." IEEE Transactions on Education 60.1 (2017): 38-39.

[4] https://www.coursera.org/learn/developer-iot.

[5] He, Jing, et al. "Integrating Internet of Things (IoT) into STEM undergraduate education: Case study of a modern technology infused courseware for embedded system course." Frontiers in Education Conference (FIE), 2016 IEEE. IEEE, 2016.

[6] Samuel J. Dickerson, "Preparing Undergraduate Engineering Students for the Internet of Things," American Society for Engineering Education (ASEE) Annual Conference and Exposition, New Orleans, LA, June 26-29, 2016.

[7] Jeong, Gu-Min, et al. "Course design for Internet of Things using Lab of Things of Microsoft Research." Frontiers in Education Conference (FIE), 2016 IEEE. IEEE, 2016.

[8] Voas, Jeffrey, and Philip Laplante. "Curriculum Considerations for the Internet of Things." Computer 50.1 (2017): 72-75.

[9] http://www.ti.com/tool/msp-exp432p401rM. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[10] https://www.arduino.cc/

[11] https://www.raspberrypi.org/

[12] http://energia.nu/

[13] Valvano, Jonathan. Introduction to the MSP432 Microcontroller. CreateSpace. 2015

[14] "MQTT 3.1.1 specification". OASIS. December 10, 2015.

[15] http://www.ti.com/tool/cc3100boost

[16] https://www.ibm.com/cloud-computing/bluemix/

[17] https://aws.amazon.com/iot/

[18] https://nodered.org/