AcquaSmart: An Environment Big Data Analytics and Internet of Things to Education and Research

Gustavo Scalabrini Sampaio, Gabriel Melo Francisco Correia, Leandro Augusto da Silva
Universidade Presbiteriana Mackenzie
Brazil

Abstract

Being an interdisciplinary area, Internet of Things presents great challenges to learning. However, it already is and will continue to be part of the daily life and thus requires qualified professionals to advance projects in this area. Apart from acquiring theoretical concepts, students need to put knowledge into practice. This practical learning aims to provide a means of easy assimilation to the student and that can mirror real situations of implementation. This work presents an Internet of Things learning methodology based on the development of environments that enable the student to put theoretical knowledge into practice in a scenario of easy assimilation. It is expected that the student will be able to understand the process of developing Internet of Things projects and the technologies involved in it. The proposed methodology is composed of 5 steps. The student analyzes the development environment, defines the type of implementation to be carried out, develops the hardware, the software and documents of the project. The data architecture together with the methodology allow the student to use and propose various types of development environments, controllers and web applications, being very flexible for learning. The implementation of temperature control was carried out in an aquarium environment. The proposed methodology proved to be efficient for the development of this project, so it can be applied in Internet of Things learning in educational institutions.

Keywords: Internet of Things; Education; Learning Methodology; Big Data.

1. Introduction

It is evident that the current generation of students has required hard work from teachers and a transformation in the way of teaching. The challenge for teaching is to quickly arouse the interest of the students so that a subject becomes attractive and keeps them engaged in classroom or research activities, especially for college students. Engaging the student in a research without a well-defined project can discourage the activity. This is related to, among other things, basically two aspects: access to information, which is widely facilitated in terms of hardware, software and communication systems; and the way each student finds to get their understanding, often diverging from teacher's planning. Most subjects of computing and even engineering areas are, to some extent, abstract to the students, what was verified through experiments in teaching activities in the higher education in a South America university. Thus, the elaboration of scenarios with real-life examples that connect with students' routines can spark their interest in learning and research.

The scientific community in areas related to exact sciences has observed the constant interest of research involving Internet of Things (IoT) and Big Data. These areas have been receiving a lot of attention, especially from development agencies, and contribute significantly to the teaching of science, technology, engineering and mathematics (STEM) (Glancy et al., 2017; Wu and Anderson, 2015). In the literature there are reports that the use of IoT and Big Data in the classroom stimulate students' interest in learning (Glancy et al., 2017; Marquez et al., 2016; Oh et al., 2017; Otieno et al., 2017). These works have motivated the present research and, at the same time, posed a challenge as to how such concepts can be introduced in the classroom and which platform for stimulating research can be constructed.

In view of this, the proposal of a teaching methodology based on development environments was elaborated, taking an aquarium as an example of contextualization. This element is known to students, all its environmental boundary conditions are known a priori and, at the same time, fundamental concepts of IoT or Big Data are placed in it. Students can quickly make a connection with the scenario, that is, what is being proposed is a strategy to bring their daily life closer to practical applications. In this proposal, the aquarium serves as a playful context, in the sense that it is concrete rather than abstract, which enables the teacher to clearly guide the student through the activity and, more interestingly, the way which it was designed, stimulates the student's creativity for the development of research. Therefore, the general objective of this work is to present a teaching methodology based on development environments in the area of IoT and Big Data, in the form of a laboratory, that can serve for the development of learning and research of undergraduate and postgraduate students in engineering and computing, possibly extending to elementary education. The specific objective is to propose a relatively playful context that enhances easy assimilation of activities to be executed in an aquarium environment, activities that involve IoT, Big Data and other disciplines such as network architecture, mathematics and statistics, computer vision, and artificial intelligence.

The environment of the project, named AcquaSmart, came from the need to turn the research of Big Data and Applied Analytical Methods into an environment that is easily assimilated by students, showing how scientific research can be applied to real problems and also be of public use. The AcquaSmart does not intend to automate an aquarium, but to create a framework in which different sensors are coupled on an Arduino board, generating measures to be later translated into strategic information for decision making, modeling of inference based on artificial intelligence, methods of data visualization, architectures of Big Data etc. Will be presented the implementation of a single sensor to measure the temperature of the aquarium in real time and present analytical measures that are translated into graphic visualizations that provides a dimension of the water situation. An actuator to heat the water when needed will also be implemented. The aquarium was intended for a *Betta* fish. The required temperature conditions for this species were researched, which enabled the establishment of patterns of values to be monitored and used to perform temperature control. The collected data and the history of situations were stored in a data server, which can be used for analytical methods.

This paper is organized as follows: section 2 presents related works which explore technology as a teaching tool and applications of IoT and Big Data, and how these works contributed to the present work;

section 3 presents the proposed teaching methodology, using playful development environments and the standard technological architecture for these environments; section 4 presents the application of the methodology and architecture in the aquarium environment, the first implementation of sensor and actuator and use of a web application for data visualization, as well as the analysis of data flow processed by the data server; section 5 presents the conclusion and possible future research.

2. Related Works

The teaching and learning of IoT are beginning to be incorporated in the classroom and, with this, new methodologies or pedagogical practices are being developed to facilitate the learning and approach of theories and practices in an interdisciplinary activity. The pedagogical works in the area of IoT reinforce that this subject is present in people's daily life and can be an instrument for the automation of several activities. Burd et al. (2017) presents some motivations for IoT learning, one of which is to gain knowledge of contemporary computing areas such as embedded systems, networks, client-server architecture, service-oriented architecture, and human-computer interaction. In addition, through IoT one can learn how to develop useful systems for the industry, analyze data for decision making, as well as deal with ethical questions derived from data analysis. Some papers present methodologies or implementation of pedagogical practices describing a specific scenario for the learning of IoT (Farhat et al., 2018; Khan et al., 2017; Njeru et al., 2017). Other works present more open methodologies, where a learning framework is described (Akiyama et al., 2017; He et al., 2016; Maenpaa et al., 2017).

Njeru et al. (2017) proposed a system of data collection by IoT devices. The data collected refer to heart rate, brain signals, temperature, body composition and eye movement. From these data, together with questionnaire responses on the content addressed with the students, analyzes and graphs are generated in order to produce reports to assess the efficiency of the adopted teaching methods. Some of the techniques used during the analysis of the data are Decision tree, Artificial Neural Networks and Support Vector Machine. Although this work presents only the application of IoT to reach its objectives, the use of techniques such as Neural Networks and Decision tree to extract information from the data demonstrates the importance of the joint use of IoT with Data Science techniques. Khan et al. (2017) presents a digital learning environment based on a Quadrotor drone, where students learn about control systems. Students develop necessary skills for control systems, working with the support of manuals that deal with mathematical modeling, transfer functions, performance analysis and other topics. An interesting result of the work was the increase in students' interest in developing their undergraduate dissertations in control systems, indicating efficiency in attracting students' attention to this area. The present work presents a similar approach to learning, where students have the opportunity to develop practical work in the area of IoT, however with the advantage of having more freedom in the development, stimulating creativity and allowing them to continue the work as research projects. Farhat et al. (2018) presents the application of IoT focused on health. It shows the implementation of a medication dispenser that uses an IoT architecture communicating the machine controller, a data server containing information about patients and physicians, and a mobile application to access the machine and the information it contains. The work presents the architecture used, which resembles the architecture of this work.

He et al. (2016) and Maenpaa et al. (2017) present a framework for learning IoT, containing several modules, from fundamental to advanced, to approach concepts involved in IoT such as embedded systems, microcontrollers and networks. He et al. (2016) discusses the importance of STEM students to have knowledge about IoT, since this area is expected to grow in the near future and requires qualified people. Maenpaa et al. (2017) presents very well-defined learning phases and proposes a detailed assessment method with clear criteria. Akiyama et al. (2017) presents a step-by-step teaching methodology is presented for the construction of a prototype IoT system. The steps performed with the students are: selection of sensors and actuators of the system; execution of the program to visualize the data; establishment of communication between the IoT device and the Internet; construction of the emission system of user commands; discussion of the project and possible applications of the prototype in real situations. It is a didactic method that can be reproduced by students from the humanities area. The present work also aimed at being didactic, however in the proposed contextual environment students will carry out their developments. It is believed that this way of delimiting the environment can facilitate the insertion of the student in an IoT context, not requiring idealizations that may be unnatural to students in the initial phase of learning.

The present paper has used some concepts described in the aforementioned works and sought to increase some of these concepts, as well as to introduce different views on the education methodology. The proposed methodology will be presented in following section.

3. Proposed Teaching Methodology

During the learning process, everyday examples are commonly used to explain certain concepts and make the topic less abstract to students. In database teaching, for example, it is common to use the context of banking transactions; in the teaching of graphs, there is the example of the traveling salesman, and several other cases of contextualization to exemplify theories. The present work demonstrates the use of a teaching methodology that involves development environments, contextualizing and making the learning of IoT and Big Data less abstract.

The proposed methodology involves three elements: the development environment, the data architecture and a sequence of steps or tasks that the student must follow to use the first two elements and develop applications with IoT and Big Data. This methodology seeks flexibility and allows the application of the same data architecture and tasks to different environments.

3.1 Development environment

The development environment is the element where the student can interact and put the acquired theoretical knowledge into practice. It is the “T” in IoT, to which the sensors and actuators are coupled, allowing to collect data and control. To a lesser extent, development environment is intended to provide students with situations where IoT can be applied in real life. Some examples of development environments that are similar to the aquarium and where future applications of the proposed methodology can be performed are: reptile nurseries; work offices; museums, for the preservation of works and artifacts; robots; autonomous vehicles; access control systems; support systems, for people with disabilities; tourist assistance systems; food storage systems.

3.2 Data architecture

The data architecture used in the proposed methodology has three components: the controller, the IoT data server and the web application. Figure 1 presents a data flow diagram of these components. The hardware system, represented by the controller, is responsible for collecting data in the environment by means of sensors, control actions and transmission of data. The server, in turn, stores the data in a structured way serving as an integrator between hardware and software. The software system, represented by the web application, allows users to visualize the behavior of the environment and access the data collected to conduct future studies. Moreover, the web application can have hardware control functions, for example passing control parameters, actuation commands, among others, sending these signals to the data server. In this context, the controller must be developed considering the reading of these commands on the server. The components can be constructed and developed according to the project's objectives. The structured scenario for students of the IoT laboratory uses the Arduino UNO microcontroller, a data server implemented in NodeJS with NeDB database and the web application developed in R.

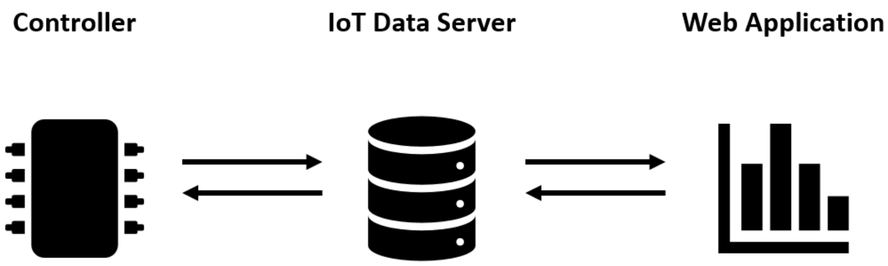

Figure 1. Data architecture.

The controller also has many choice alternatives, since there are several options in the market, such as PIC, Raspberry and BeagleBone. However, the basic architecture proposed in the laboratory relies on the Arduino UNO. The first reason for choosing this microcontroller was the fact that its features are easy to learn and operate, being a good option to work with students. The second reason was the affordable price, around US$20.00, making it easier to replicate the development environment. Finally, it was considered that it can be easily found and purchased. This microcontroller, has several digital and analogue ports and can be programmed in C language making it flexible for several applications. Today, it is a widely used microcontroller for the development of IoT solutions and one of the most successful as a teaching tool for STEM (Arduino, 2018).

The purpose of the IoT data server was to provide a collection, aggregation and data availability service capable of communicating over HTTP (Hypertext Transfer Protocol) via TCP/IP (Transmission Control Protocol/Internet Protocol). Due to the nature of the sensors and the potential heterogeneity of the technologies employed, the developed service must be capable of attending as many sensor implementations as possible. Given the number of designs that can be implemented and the number of sensors used, the sensor registration and communication configuration has been simplified to facilitate the development (Fall and Stevens, 2011; Fielding et al., 1999; Gluhak et al., 2011).

The developed server service presents an Application Programming Interface (API) for the transmission the collected data by the sensors. The controller informs the server of a sensor identifier (ID) and the sensory value to be persisted by the service. The server then persists the data in a database along with the date and time of data receipt. The data recovery is done by the same API, specifying the ID of the sensor. The NeDB database used for storing data was developed in JavaScript, and takes up little disk space for simple installation and maintenance (Chatriot, 2018; MongoDB, 2018). This database persists the data in a text file in JavaScript Object Notation (JSON) format. If there is an update of a persisted data, a new entry containing the updated data is inserted at the end of the file. The server application follows the MVC (Model-View-Controller) structure, having a generic and flexible data model. Its implementation is done in NodeJS, which was chosen for its event-based framework, enabling the clustered implementation of the service (Leff and Rayfield, 2001; Tilkov and Vinoski, 2010). The data server has a portal where the user can register, manage their sensors and monitor data insertion in real time. The server also provides a security layer based on tokens (M'Raïhi et al., 2011). So, for a sensor to send data to the server, a unique token must be provided to the API. The user of the system has a tool that generates the sensor code, with the security tokens configured. This tool is capable of generating codes for the most used platforms (Arduino, ESP8266 among others). There are other options for managing data via API, such as making queries, analyzing time and quantity of entries. The server has an internal management module to monitor all real-time sensor data reception actions based on websockets, allowing to generate fault alerts.

The web application is the component of architecture presented with greater development flexibility. There are several programming languages that can meet the proposed functionality for the application, such as parameterization menus of the controller and data visualization. However, the R language was used as it is a free development platform for statistics and data visualization (The R Foundation, 2018). With the incorporation in R of the Rshiny and shinydashboard libraries, web applications can be simply and quickly developed and made available for users on the internet. In addition to these factors, R language allows the development of data mining, machine learning and other statistical analyzes.

3.3 Learning Methodology

When faced with the architecture described above, the student has an overview of how data flow happens between the components of this architecture. Considering the development environment, the student can analyze possible sets of data. In the case of the aquarium, they can be water temperature, luminosity, automatic fish feeding, among others. This is the role of the development environment, allowing a quick assimilation of the tasks that will be performed and contextualization of IoT and Big Data. With this scenario in mind, the student can execute a sequence of tasks to put into practice the acquired IoT knowledge, enhancing the environment or initiating development in a new. The proposed steps are:

1) Analyze the current state of the environment and define new types of sensors and actuators: the first procedure is to analyze the functioning of the environment where the new implementations will take place, to assist in deciding the types of elements that will be inserted. In this phase, the attention must be focused on the variable that must be controlled or on the functionality that will be added, leaving aside technical issues such as sensor or actuator model.

2) Define and install the chosen sensors and actuators: after defining the type of implementation that will be performed in the environment, the student must specify, now in a more technical way, the elements and functionalities that will be added. For the sensors it is ideal to carry out researches analyzing their characteristics and checking questions such as: advantages and disadvantages of the digital and analogue sensors of the variable; feasibility of physical and logical implementation; accuracy of measurements; cost and possibility of purchase. There are several sensors developed for microcontrollers that require low voltage and provide data exchange by communication protocols, such as SPI (Serial Peripheral Interface) and I^2C (Intermediate Integrated Circuit), being therefore easier to implement than actuators. Some actuators such as resistive heaters and lamps have no direct interface with microcontrollers, requiring circuits and shields to works properly. Thus, actuators and auxiliary components must be well-defined to avoid setbacks.

3) Increase or develop a new controller software according to the chosen sensors and actuators: because it is a modular methodology, that is, one that can have several inserted functionalities over time in different development environments, the student should consider the present moment of the environment. Initially, at a time when the environment does not have elements or has few elements, the implementation of a new functionality can only be accomplished by incrementing the controller software. As the environment gains new features and the complexity of control increases, it may be necessary to insert a new controller, as well as to develop a new software that can deal with the new elements and consider behaviors of variables from other controllers. This modularity has the advantage of allowing the development of new implementations in parallel by different students, making feasible the integration between the students.

4) Develop web application for implementation: similar to the implementation of new physical components, the development of the web application should be carried out considering the new features proposed by the student. The implementation may be incremental, adding visualization features or extracting data from a new variable in an existing web application, or a new web application can be developed. Although this is the fourth step, this phase can be seen as a first step towards an analytical implementation in the environment. Students can also perform only analytical developments, such as data mining, Big Data, or artificial intelligence considering the elements and data already present in the environment. In this case, students should analyze the characteristics of the environment, execute the development and propose improvements or modifications in the way of controlling the variables.

5) Implement and document: in the creation of study environments, aimed at the participation of several students, it is important to document what was implemented, describing the purpose, materials, algorithms, as well the analysis of the data collected. This documentation can be used by students to check the status of the environment and the details of the features already inserted, making it easier to define new functionalities or analyzes.

At the end of these steps, it is expected that the student will be able to understand the process of developing IoT systems and the technologies involved in it. The next section shows the implementation of an element in the aquarium by following the steps described in this section.

4. Application of the Proposed Methodology

To demonstrate the application of the proposed learning methodology and how development environment can be used in teaching and research, a first element of control in the aquarium was implemented. Initially a single sensor to measure temperature and an actuator to heat the water were inserted, besides that as a web application was developed to monitor the measurements and access the collected data. The procedures performed following the methodology steps presented in section 3 are presented as follows.

The first step was to analyze the current state of the environment and define the new types of sensors and actuators, leaving aside the technical details of the sensors and actuators for the moment. For the environment of the aquarium, it was considered that it was intended for a single fish, of the species *Betta*. This definition was relevant to determine the variables that should be controlled to provide a good quality of life for the fish. Research has shown that this species is found in hot locations and the ideal water temperature for captive breeding should range from 22°C to 28°C (Brammah, 2015; Masters, 2015; Srikrishnan et al., 2017). The temperature inside the aquarium is affected by the variation of room temperature due to use of air conditioner, season of the year or number of people in the place. Thus, the first controlled variable in this environment was the water temperature. To implement this control, a temperature sensor and a water heater as actuator were used.

After defining the controlled variable and the control components, the second step was to determine the type of sensors and actuators and to install them. The water temperature was measured by a Maxim DS18B20 sensor. As actuator, a resistive heater for aquariums was installed. The DS18B20 sensor, used to measure the water temperature, communicates with the controller via a proprietary Maxim protocol, the 1-Wire Bus Protocol. This communication protocol uses only one digital controller port for all devices, organized on the same data bus (Maxim Integrated, 2015). The heater used as a system actuator, suitable for aquariums, uses a resistance inserted in an acrylic envelope filled with sand. The energy dissipated by the resistance in the form of heat is transmitted by the enclosure to the water, causing the temperature to rise. Since this heater must be connected to an AC power source, it was necessary to add a relay to allow the controller to turn the actuator on and off according to the measured temperatures. Thus, a 4-channel relay module was inserted into the system.

The third step was to develop the controller software. The Internet communication functions in the Arduino UNO were only possible by implementing the Ethernet Shield that used the WIZnet W5100 controller (WIZnet Co., 2008). For temperature control, the system remains in infinite loop, executing the following operations: measuring water temperature; checking whether the temperature is between 22°C and 28°C. If yes, only the value of the measured temperature is sent to the data server, otherwise the state of the actuator must be changed and remain active until the temperature reaches 28°C. At this point the heater must be turned off; and send the temperature and heater status data to the server. The heater is controlled by a relay, an electromechanical device that has a coil that, when fed, forms a magnet by moving the relay contact (Panasonic Corporation, 2018). By means of digital signal, the controller feeds the coil contact, thus controlling the actuator. Every 10 seconds, the controller sends the temperature to the data server, as well as the state of the heater in case of change. This is done using the HTTP protocol GET method, sending the sensor ID and the measured value to the data server IP address.

The fourth step was to develop the web application. A web application in R programming language was developed to visualize and access temperature data. This application displayed temperature data in two forms: a dashboard and a timeline view. The dashboard presents the current state of measurement and control and consists of a chart with temperature values over the last 30min, minimum and maximum temperature range, last temperature value and current state of the heater. Temperature values presented in green indicate temperatures within the expected range; values in red indicate temperatures out of the expected range. This form of presentation, with elements of color, is intuitive and facilitates the interpretation by user. The values of the last measured temperature and the current state of the heater also followed a color scheme. The timeline view presents values over the time. The user can view the data stored on the server, filter the desired period, sensor or actuator. The application generates a graph with the selected data and presents a statistical summary. This summary has information on the number of measurements performed in the period, number of times the heater was turned on, as well as minimum, maximum and average temperature of the period. This view also provides the possibility of downloading the data in a spreadsheet format, allowing to perform analyzes in external tools or visualize in other formats. Data collection from the server was also performed using the GET method of HTTP protocol, passing as a parameter the ID of the sensor to be analyzed. Figure 2 present the two views.

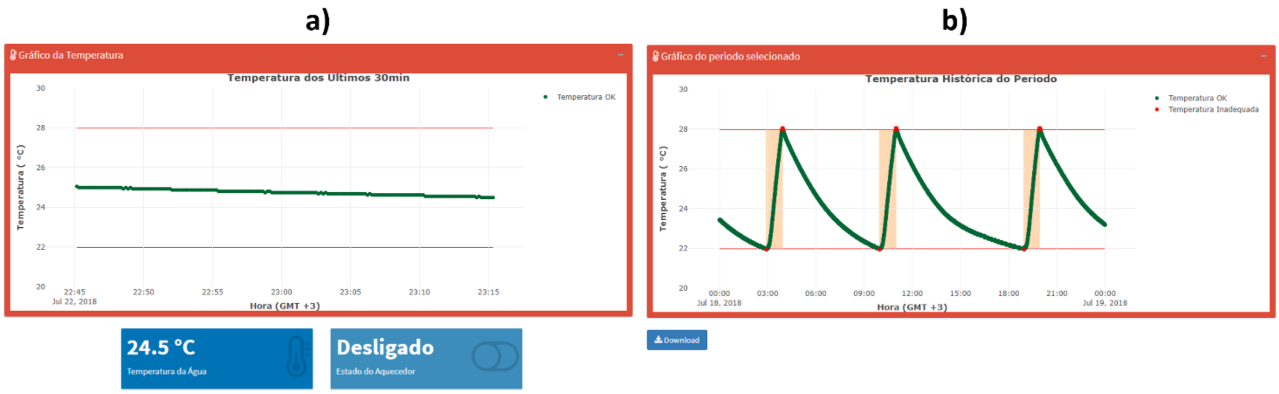

Figure 2. Views of the environment web application. **a)** dashboard view; **b)** timeline view.

Finally, the implementation documentation was produced describing the performed steps and the generated results. Figure 3 presents the development environment assembled with sensors and actuators, and the web application running the described dashboards.

Figure 3. AcquaSmart development environment.

5. Conclusion and Future Work

The implementation performed in the development environment showed that this learning methodology can be applied with students and improve the teaching and learning of IoT and Big Data. The contextualization of the development environment facilitates the understanding by student, as it materializes abstract concepts by putting into practice the knowledge acquired in a theoretical way. The data architecture presented makes it possible for both the controller and the web application to be flexible in terms of technological choice, and it allows several studies and analyzes to be performed with the data stored on the server, such as data mining and machine learning to change the behavior of the controller, among others. From the laboratory point of view, besides having a playful learning scenario from the development environment, students can produce parallel implementations, which stimulates a collaborative learning system and incremental research.

The development environment is also flexible. The present work has used an aquarium, however several other environments can be structured to enhance the learning of IoT and Big Data. Further research could explore this methodology in other environments and report the experience as well as the feedback of students, so that it would be possible to identify points to be improved or adjust some of the factors to the reality of IoT or Big Data learning. As future work, it would be interesting to assess the usability of the proposed data architecture for the study of data mining, machine learning and others, in order to verify whether the proposed methodology can also be useful as a platform for studying other concepts.

As attention to IoT and the demand for professionals who can work on projects in this field increases, the interest of educational institutions in providing that knowledge and of students in learning it grows. The proposal presented in this paper for teaching, learning and even researching offers a significant contribution to the quality of the educational process. Thus, it is important to put effort into the development of alternative learning forms that focus on practical and contextualized learning.

6. References

[1] A. Farhat, T. McNeill, and J. Raven. An interdisciplinary approach to developing an IoT healthcare solution in applied higher education. In 2018 Advances in Science and Engineering Technology International Conferences (ASET). IEEE, feb 2018.

[2] A. Leff and J. Rayfield. Web-application development using the Model/View/Controller design pattern. In Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference. IEEE Comput. Soc, 2001.

[3] A. M. Njeru, M. S. Omar, S. Yi, S. Paracha, and M. Wannous. Using IoT technology to improve online education through data mining. In 2017 International Conference on Applied System Innovation (ICASI). IEEE, may 2017.

[4] A. W. Glancy, T. J. Moore, S. Guzey, and K. A. Smith. Students' Successes and Challenges Applying Data Analysis and Measurement Skills in a Fifth-Grade Integrated STEM Unit. Journal of Pre-College Engineering Education Research (J-PEER), 7(1), jun 2017.

[5] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton and T. Razafindralambo. A survey on facilities for experimental internet of things research. IEEE Communications Magazine, 49(11):58–67, nov 2011.

[6] Arduino. About Us. Available in: https://www.arduino.cc/en/Main/AboutUs, 2018. Access on July 24, 2018.

[7] B. Burd, L. Barker, M. Divitini, F. A. F. Perez, I. Russell, B. Siever, and L. Tudor. Courses, Content, and Tools for Internet of Things in Computer Science Education. In Proceedings of the 2017 ITiCSE Conference on Working Group Reports. ACM Press, 2017.

[8] D. M'Raïhi, S. Machani, M. Pei, and J. Rydell. TOTP: Time-Based One-Time Password Algorithm. RFC,6238:1–16, 2011.

[9] D. Masters. Betta Fish 101: The Complete Betta Fish Keeping Guide. 2015.

[10] H. Maenpaa, S. Varjonen, A. Hellas, S. Tarkoma, and T. Mannisto. Assessing IOT Projects in University Education - A Framework for Problem-Based Learning. In 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET). IEEE, may 2017.

[11] 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, oct 2016.

[12] J. Marquez, J. Villanueva, Z. Solarte, and A. Garcia. IoT in Education: Integration of Objects with Virtual Academic Communities. In New Advances in Information Systems and Technologies, pages 201–212. Springer International Publishing, 2016.

[13] K. Akiyama, M. Ishihara, N. Ohe, and M. Inoue. An education curriculum of IoT prototype construction system. In 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE). IEEE, oct 2017.

[14] K. Fall and W. Stevens. TCP/IP Illustrated, Volume 1: The Protocols. Addison-Wesley Professional Computing Series. Pearson Education, 2011.

[15] L. Chatriot. NeDB The JavaScript Database. Available in: https://github.com/louischatriot/nedb, 2018. Access on July 31, 2018.

[16] M. Brammah. The Betta Bible: The Art and Science of Keeping Bettas. Blurb, Incorporated, 2015.

[17] Maxim Integrated. DS18B20 Programmable Resolution 1-Wire Digital Thermometer, 2015.

[18] MongoDB. NeDB: a lightweight Javascript database using MongoDB's API. Available in: https: //www.mongodb.com/blog/post/nedb-a-lightweight-javascript-database-using, 2018. Access on July 31, 2018.

[19] Panasonic Corporation. What are Relays? Available in: https://www3.panasonic.biz/ac/e/corp/ nyumon/relay/index.jsp, 2018. Access on July 24, 2018.

[20] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. Technical report, jun 1999.

[21] R. Srikrishnan, N. Hirimuthugoda, and W. Rajapakshe. Evaluation of growth performance and breeding habits of fighting fish (Betta splendens) under 3 diets and shelters. Journal of Survey in Fisheries Sciences, 3(2):50–65, 2017.

[22] S. Khan, M. H. Jaffery, A. Hanif, and M. R. Asif. Teaching Tool for a Control Systems Laboratory Using a Quadrotor as a Plant in MATLAB. IEEE Transactions on Education, 60(4):249–256, nov 2017.

[23] S. Tilkov and S. Vinoski. Node.js: Using JavaScript to Build High-Performance Network Programs. IEEE Internet Computing, 14(6):80–83, nov 2010.

[24] T. T. Oh, S. Chung, B. Lunt, R. McMahon, and R. Rutherfoord. The Roles of IT Education in IoT and Data Analytics. In Proceedings of the 18th Annual Conference on Information Technology Education - SIGITE. ACM Press, 2017.

[25] The R Foundation. The R Project for Statistical Computing. Available in: https://www.rproject.org/, 2018. Access on July 24, 2018.

[26] W. Otieno, M. Cook, and N. Campbell-Kyureghyan. Novel approach to bridge the gaps of industrial and manufacturing engineering education: A case study of the connected enterprise concepts. In 2017 IEEE Frontiers in Education Conference (FIE). IEEE, oct 2017.

[27] WIZnet Co. W5100 Datasheet Version 1.1.6, 2008.

[28] Y. -T. Wu and O. R. Anderson.Technology-enhanced stem (science, technology, engineering, and mathematics) education. Journal of Computers in Education, 2(3):245–249, jul 2015.