Scientific Research Publishing

Advances in Internet of Things, 2021, 11, 95-112
https://www.scirp.org/journal/ait
ISSN Online: 2161-6825
ISSN Print: 2161-6817

Design and Development of Remote Laboratories with Internet of Things Setting

Abul Kashem Mohammed Azad

College of Engineering and Engineering Technology, Northern Illinois University, DeKalb, USA
Email: aazad@niu.edu

How to cite this paper: Azad, A.K.M. (2021) Design and Development of Remote Laboratories with Internet of Things Setting. *Advances in Internet of Things*, **11**, 95-112.
https://doi.org/10.4236/ait.2021.113007

Received: April 1, 2021
Accepted: July 3, 2021
Published: July 6, 2021

Abstract

Given the recent developments of Internet of Things (IoT) researchers and academics are trying to utilize this concept for different innovative applications. One of those applications of IoT is the remote laboratories, where real hardware/software experimentation systems can be controlled and manipulated over the web. During the COVID-19 pandemic, this has become almost essential for academic and research activities. The paper starts with highlighting the necessity and current status of remote laboratories in terms of shortcomings and possible recourse. With these in mind, the second part illustrates an infrastructure of a remote laboratory facility as well as describes the development of a number of prototype remote laboratories that are utilized for academic activities.

Keywords

Internet of Things, Remote Laboratories, Embedded Systems, Graphical User Interface, Cloud Technologies

1. Introduction

Providing laboratory activities is vital to optimize the learning in Science, Technology, Engineering, and Math (STEM) disciplines. In almost all cases, to conduct an experiment with a real hardware student must need to visit the laboratory where the system is physically located. Usually laboratory courses are planned only for a set time period. Given the nature of laboratory activities, most of the time fixed time allocation is not enough for all students to fulfil the laboratory outcomes effectively and efficiently [1] [2]. On top of this quite often students desire to conduct activities beyond their regular tasks. It is very much difficult, if not impossible, to accommodate any extra laboratory time due to various limitations. Additionally, due to geographical locations sometimes students from other academic units do not have access to the physical laboratory facilities. At the same time, with these limited opening/access hours, much of the experimental systems lie idle during a large part of their lifetime. All these issues indicate that conventional ways of laboratory course offerings are not always effective. In addition, during an unusual situation like COVID-19, regular laboratory access is also a major challenge.

A remote laboratory can be the effective way of delivering laboratory experience, where one can get unlimited access to perform experiments. This can also maximize the utilization of available resources without any geographical limitation [3] [4] [5]. Also, only remote laboratories can facilitate inter-laboratory collaboration among universities and research organizations, where students and researchers can share a variety of costly equipment at geographically distributed locations.

Due to the failure to deliver laboratory courses, the growth of distance-learning programs are also limited to certain disciplines. The inclusion of STEM disciplines for distance-learning is slowed down due to the lack of provision for laboratory course offerings [6] [7]. Simulation and multimedia can facilitate a level of educational experience; however, for effective and meaningful education within the STEM disciplines, a combination of theoretical and laboratory sessions is essential. As it is now, distance-learning programs need their students to visit a designated campus for the laboratory part of the program and most of the time these are for a limited period of time [8] [9]. These limited periods of laboratory exposure are insufficient to achieve their learning objectives [1]-[10]. Providing access to remote laboratory facilities over the web and/or cloud would address this problem. Given the necessity of remote laboratory facilities and emergence of Internet and cloud technologies, there is a number of initiatives to develop remote experimentation facilities [11]-[18].

With available emerging technologies and better knowledge on learning strategies, groups of academics and scientists are making significant progress in developing remote laboratories. Even after all these, acceptances of remote laboratories as a part of standard curriculum are still limited. There are a number of factors that are influencing this. These can be listed as: Involvement of various hardware and software technologies; Commercial products to support the developments; Learning management system integration; Awareness of benefits among the administrators; and Industry support.

Traditionally, personal computers and workstations are utilized as the main unit for the experiment part of remote laboratory developments and a local server for database and user management. With more powerful, faster, and resourceful embedded processors, remote laboratory developments are now tending to utilize these new technologies instead of traditional computers and servers. Developers are now using embedded processors for remote laboratory developments.

To illustrate this **Figure 1** provides a framework of a remote laboratory platform using an embedded system. The main components can be identified as the

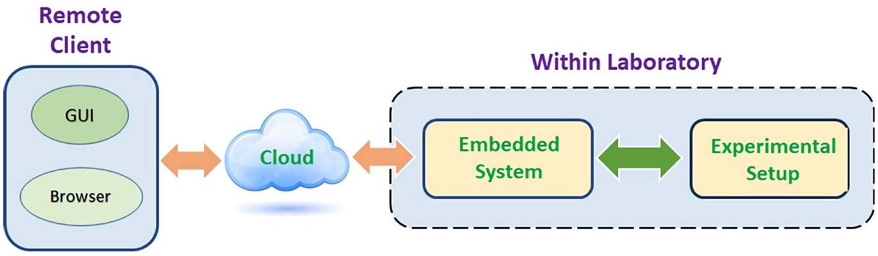


Figure 1. Embedded system integration for remote laboratory.

experimental setup, embedded system, cloud, and remote client. The experimental setup is connected with an embedded system, which plays the role of the gateway between the experiment and the remote client.

These facilities utilized embedded processors for facilitating a number of experiments, providing remote client to manipulate experiments, providing needed assessment, and implementing real-time learning management. The developed facilities are utilized for delivering laboratory classes and at the same time collected data for assessing effectiveness and system development.

The rest of the paper contains five additional sections. Section 2 discusses the state of remote laboratories highlight their limitations as well as potential discourse to address those. Section 3 describes the IoT structure for remote laboratories involving embedded processors as the main controller. Section 4 presents the development of three remote laboratory systems utilizing the framework mentioned earlier. Section 5 provides system utilization data and its analysis. At the end section 6 concludes with a discussion considering on what was presented through this paper.

2. State of Remote Laboratories

This section provides a discussion involving the current state of remote experimentation. The main focus is to identify the shortcoming and stumbling blocks towards the progress of remote experimentations. This is followed by an outline of potential strategies to address those issues.

2.1. Limitations

Conducting remote experimentation utilizing real physical systems over the web is a relatively new concept. Academics and researchers working on remote laboratory developments in an abrupt manner but yet to develop a sustainable solution that can establish their wide scale acceptance. The major shortcomings are concentration of a number of technologies, absence of modular design strategy, lack of commercial products, absence of learning management system integration, shortage of maintenance personnel, and insufficient of administrative support. This section will discuss each of these underlying issues.

Expertises from various disciplines are essential to facilitate any development with remote laboratory systems. This includes hardware interfacing, data gathering, application development, networking, cyber-security, and real-time control. Also the nature of experiments differs from one discipline to another. All these factors hinder the modularity in system design and the development of a common framework. Each development initiative in this area tends to start from a scratch, like reinventing the wheel. This introduces a challenge in transferability of one component of a system to another or inter-system integration. This becomes almost impossible for remote laboratory systems to interface with another system. Sometime, one has to redesign the system to accommodate new experiments [19].

There is an extensive development hardware and software technologies in recent years. This opens up an opportunity to the remote laboratory system designers to develop powerful systems that were unthinkable in the recent past. However, implementation of these needed some level of expertise which is not readily available within an institution. Also, most of the commercial products and tools are not designed with remote laboratories in mind. On the other hand, the traditional learning management systems such as Blackboard, is not designed to be incorporate the delivery of remote laboratories. This is a major stumbling block for offering remote laboratories as a part of a regular curriculum.

Considering the complexity of remote laboratory systems, getting a suitable maintenance technician can be challenge. Most of the time these are developed and managed by graduate students and/or entry level researchers. Most of the organizations can't afford to have a permanent staff to maintain these facilities.

Another major drawback is the lack of administrative support for the remote laboratory developments and their maintenance. The potential of remote laboratories were not fully realized by administrators in academic institutions. As a result remote laboratory development efforts are not getting enough attention from administration and solely depend upon external grants and projects, which are support any sustainability.

2.2. Potential Discourse

In light of the above discussion it is clear that there is a major technological progress in remote laboratory developments. However, most of the cases these are used as a pilot projects mainly for in-house development purpose. Not many of these developed systems were included within the regular educational programs. It would be a good idea to address few key questions before we proceed further, so that the development of remote laboratories would be widespread and more acceptable to ensure sustainability.

Figure 2 illustrates the modularity in remote laboratory designs. The main component of this design is Interface module, Experiment server, Learning Management System (LMS) server, and Local network. To have a level of flexibility in operation, future expansion, and collaboration, there should be an understanding within the community for an interfacing standard between the modules. In this regard, recently, academic and researchers around the world took an initiative and formed Global Online Laboratory Consortium (GOLC).

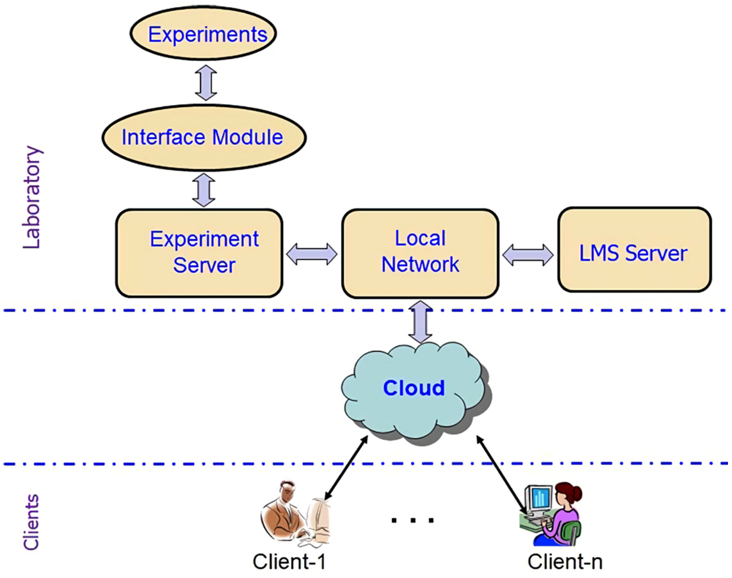


Figure 2. Design modularity for remote laboratories.

The role of GOLC is to discuss and explore sustainable structure for remote laboratory developments among the partners. This will streamline future developments through a modular design of system components with pre-determined inputs/outputs for each module. This platform will foster developments that will facilitate interfacing between the systems, ability to handle different kinds of experiments, and capacity to share resources to maximize the use of available resources.

In terms of development of commercial products, it is important to culminate collaboration between academia and industry with a specific goad to design and prototype custom products that can be utilized for remote laboratories. In this regard one can look for federal, state, and industry funding. In U.S. there are funding for small business initiatives from federal agencies. To get attention, it is important to highlight the necessity and potential of remote laboratories through lectures and presentations in various forums so that it can get attention of administrators and decision makers.

3. IoT Structure for Remote Laboratories

A conventional remote laboratory development involves a personal computer along with associated interfacing and web/cloud hosting technologies. This involves a considerable investment for commissioning and further maintenance [20] [21]. To mitigate this, the author took an initiative to utilize an embedded processor for the development of a remote laboratory facility. With this arrangement the function of personal computer and server is taken over by the embedded processor [22]. Within this project a Raspberry Pi was utilized as the system controller as well as the web server. Raspberry Pi, a small-scale computing system with its own operating system, replaced a full-scale computer/server, thereby reducing the cost and complexity of the remote laboratory design [23]

[24]. The developed experiments can be accessed remotely over the web using a suitable graphical user interface (GUI). The GUI is accompanied by a live video feed, using a Raspberry Pi Camera, so the user can have a real time video of the experimental facility. The system includes a provision so users can download data to a local system for offline analysis. **Figure 3** illustrates this layout with Raspberry Pi at its core. Raspberry Pi is a cost effective yet a very efficient solution to set up a remote laboratory system. As a single board computer, it functions as a web server through which remote users can access the laboratory through the internet. It also functions as an embedded processor to interact with the hardware of the experiment modules residing in the laboratory.

It should be noted that although we have used Raspberry Pi in this project, any other single board computer, such as Beagle Bone Black, can be used equally to develop such a system. Such computers, also referred to as microprocessor breakouts, at times may even be preferred over Raspberry Pi if higher capacity or performance is sought.

A number of Raspberry Pis are used in this project, each of which runs its own circuit and controls an experiment. An additional Raspberry Pi was used in this configuration to run a web server that posts an entry webpage to the remote laboratory. This was an initial stop point or welcome page for remote users. Then the webpage served as a gateway to direct a remote user to the web server of the experiment as desired by a client.

Python was used as the main programming language to develop the required software for this project. This is one of the most widely-used high-level programming languages with an extensive knowledge base for developers. Additionally, it has a simple and easy to understand syntax that emphasizes readability and thereby reduces the cost of program maintenance. It is easy to develop internet applications with the aid of open source modules and packages.

The Raspberry Pi uses Raspbian as its operating system, which is a Debian Linux based operating system customized for Raspberry Pi. Given a complete system Raspberry Pi can run its own web server, a built-in Apache web server. This allows Raspberry Pi to render web pages for the client-side graphical user interfaces (GUIs) and facilitate remote interaction with hardware that are attached to the Raspberry Pi. Python is used to program the web server through

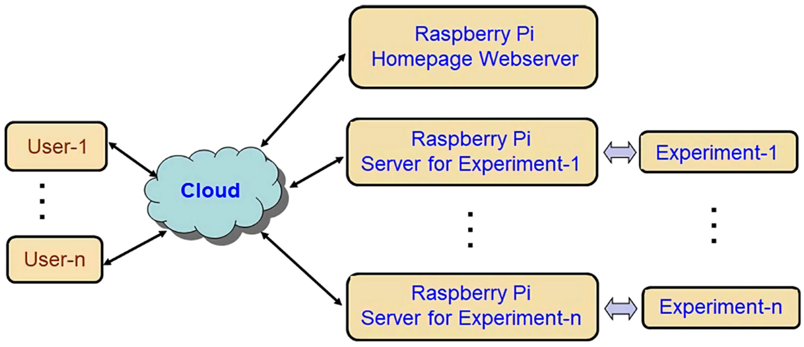


Figure 3. Modular layout of the remote laboratory.

Flask to facilitate this communication and execution procedure. Flask is one of the popular web frameworks, such as Django, that expedites development of web applications using Python by providing a solid core with basic services. Moreover, Flask supports databases, web forms, authentication, and other high-level tasks by the application of its extensions. Flask extensions are readily available to integrate with the core packages, and not all extensions need to be installed when developing an application. The web technologies used in this project are illustrated in Figure 4.

4. Experimental Systems

This section provides some details of three remote experimentation facilities developed utilizing the structure that is proposed in the previous section. The first one is the remote vacuum cleaner, where one can program an embedded processor to control the vacuum cleaner. The second experiment is the remote experimentation with an embedded processor system. This involves the development of experiments where students used Python to program a Raspberry Pi over the web to work with the sensors and actuators. The third experiment is a structural health monitoring system, which monitor structural health of an infrastructure through real-time diagnosis. A laboratory scale suspended bridge was designed and constructed along with a set of sensors and embedded system for data collection and processing.

4.1. Remote Vacuum Cleaner

The first experimental system is a vacuum cleaner fitted with a mobile platform with self-navigation capability. A server is wirelessly connected with the mobile platform to control the operation. A range of sensors and actuators are deployed within the system to facilitate obstacle avoidance with self-navigation. There is an onboard embedded processor to control the whole system. To facilitate interaction between the remote user and the system a suitable GUI was designed and developed. Overall system diagram of this development is provided in Figure 5.

The system has three main components: a mobile platform as the remote testbed system, a local server is the gateway between the testbed and remote

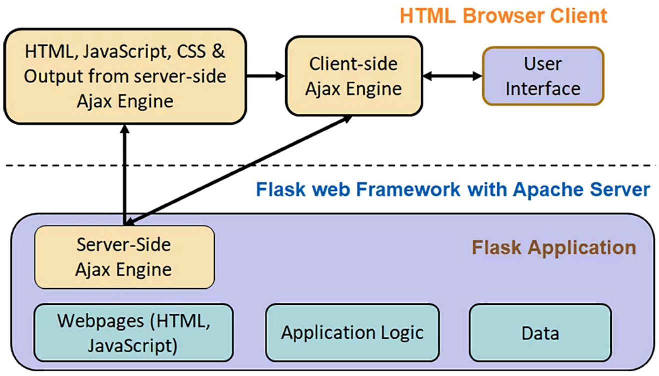

HTML Browser Client
HTML, JavaScript, CSS & Output from server-side Ajax Engine
Client-side Ajax Engine
User Interface
Flask web Framework with Apache Server
Flask Application
Server-Side Ajax Engine
Webpages (HTML, JavaScript)
Application Logic
Data

Figure 4. Software interaction and information flow.

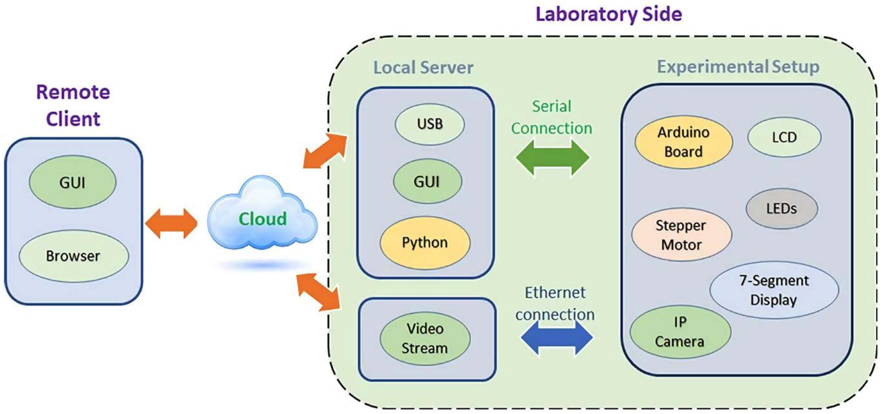
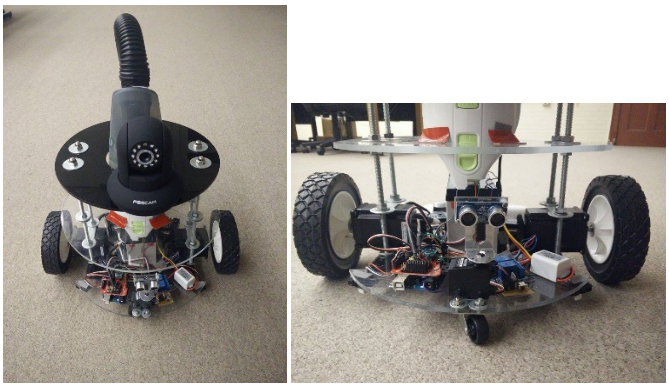


Figure 5. Overall system diagram for remote vacuum cleaner.

clients and the remote client. The mobile platform consists of a drive system, sensors for navigation, an embedded processor (Arduino board) for local control and data management, an XBee for wireless communication with the local server, and an IP camera for real time video. The IP camera has its own communication route via a WiFi channel. The video is then embedded within the GUI for user monitoring. Images of completed mobile platform are shown in Figure 6.

One of the important components of remote testbed designs is to provide the user with a GUI with excellent visual composition. The purpose is to provide better visual experience for the user through coordination between human eye and computer screen. Considering these issues for an effective GUI, HTML was used as a software tool for testbed GUI development. Along with HTML, Cascaded Style Sheets (CSS) were provided to improve the overall visual experience. The GUI contains all the input and output selection buttons and variables, which can be adjusted by a client. Figure 7 shows an image of the developed GUI showing all the functionalities. There is an on/off switch provided at the top right corner of the page. Just to the right side of this, there is a mode selection switch. The system can be operated in two modes: auto and manual. On the top right corner there is a vacuum on/off switch for that can activate and deactivate the vacuum cleaner. In the manual mode one can use the arrows for motion control and the red square button to stop. In the auto mode the system travels on its own guided by a set of IR sensors for wall detection and a pair of ultrasonic sensors for obstacle detection. In addition, users can adjust the speed and allow proximity to an obstacle. On the mid left, there is an arrow to determine the distance from the nearest obstacle at any point. The distance will display at the bottom. At the bottom right, the safety status is displayed to show the danger from a nearby obstacle.

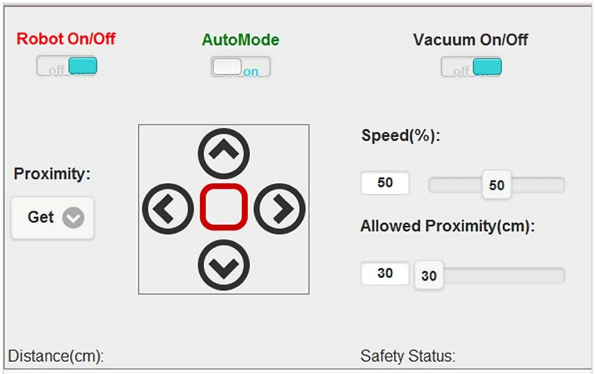
A high level view of the program execution cycle for the server system is shown flowchart in Figure 8. To start the server, the Tornado Python script should be executed [25] [26] [27]. At the beginning, Python takes in all the modules required for operation using import method. It initiates the opening of a serial port. There will be an error message if the serial port is already being



(a)

(b)

Figure 6. Images of remote vacuum cleaner. (a) Complete system; (b) Sensors and actuators.



Robot On/Off

AutoMode

Vacuum On/Off

Proximity:

Get

Speed(%): 50

Allowed Proximity(cm): 30

Distance(cm):

Safety Status:

Figure 7. Developed GUI for remote vacuum cleaner.

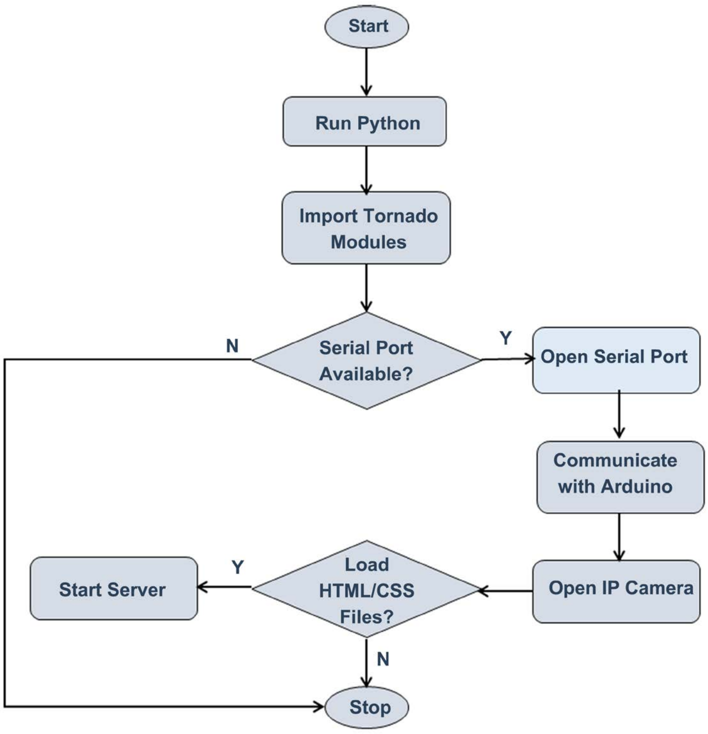


Figure 8. Server program flowchart.

used or is not available. The server communicates with the processor once it opens the serial port. The server script requires all of the HTML/CSS files when a request is made from the client side. The files are loaded before starting the server, and whenever a request is made, it will transfer the files to the client's browser as a webpage. In case of any irregularity in the procedure, it will exit the loop and will display an error message.

4.2. Embedded Processor Programming

This section illustrates a remote experimentation where one can program an embedded processor system over the Internet. Within this facility multiple input and output devices are connected to a Raspberry Pi, which serves as an embedded processor to manipulate these devices. The goal of the facility is to provide students with a platform to learn how to develop an embedded system using Raspberry Pi as the embedded processor and Python as the programming language. Figure 9 shows the general layout of this facility.

A student can remotely access this module through the web pages broadcast from the web server on the Raspberry Pi. The student can watch a live video stream that shows the entire module as well as observe the state of the devices attached to the module through custom displays on the GUIs of these web pages. The student can manipulate the state of these devices through the controls provided in the GUIs and immediately observe the changes made.

The Python codes that perform the manipulation of the devices are downloadable from the web pages together with relevant background information needed to program the Raspberry Pi for these manipulations. The student can download these code files and study them to learn how they work. Then the student can write, upload, and execute own Python program to manipulate the system and observe the result immediately. Thus, the main design aspect of this system is that it can be programmed from anywhere at any time.

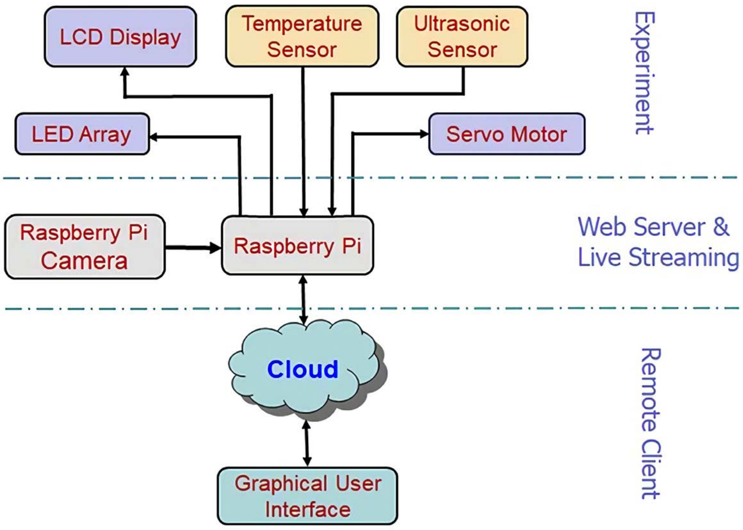
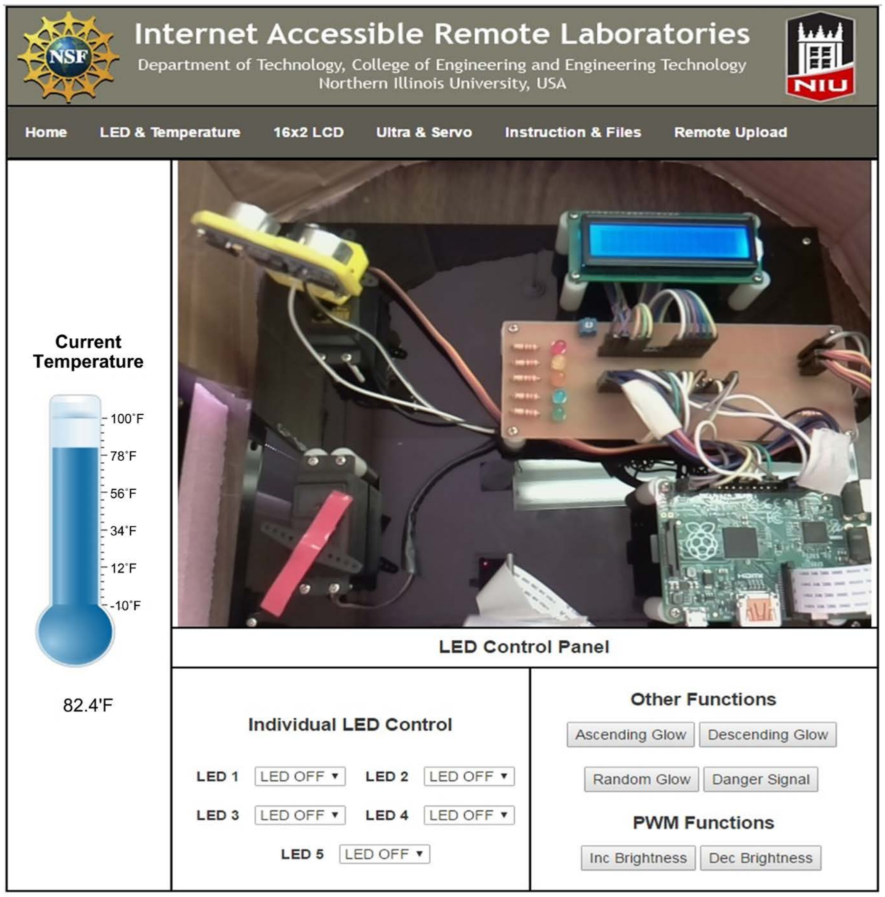


Figure 9. Block diagram of remotely programmable Raspberry Pi.

In addition to a live video camera attached to the Raspberry Pi, the setup includes two input devices: an ultrasonic sensor and a temperature sensor. There are three types of output devices: LEDs, a 16 × 2 LCD, and a servo motor.

Figure 10 shows an image of the webpage for LED control and temperature monitoring. The horizontal menu on the page has links to custom pages for other devices in the module. The picture on the image is the live video of the module from the camera, which shows the ultrasonic sensor at the top left, servo motor at the bottom left, LEDs at the center of the circuit board, and the LCD display above and Raspberry Pi below the circuit board. The GUI on the left of the page shows the temperature in the laboratory room using a graphical display and in text. The GUI at the bottom shows the state of the LEDs and provides controls to manipulate their states. Five LEDs are connected to the system, and LEDs can be controlled individually as well as together using specific functions, such as making LEDs glow in ascending or descending order or increasing and decreasing brightness levels by implementing pulse width modulation (PWM) through the controls on the GUI. Accessible through the "12 × 2 LCD" link on the horizontal menu, user can change the text interactively. The changes on the LCD can be monitored through live video stream. The LCD module is integrated with the Raspberry Pi using the Adafruit LCD library.



Internet Accessible Remote Laboratories
Department of Technology, College of Engineering and Engineering Technology
Northern Illinois University, USA

Current Temperature
82.4'F

LED Control Panel

Individual LED Control

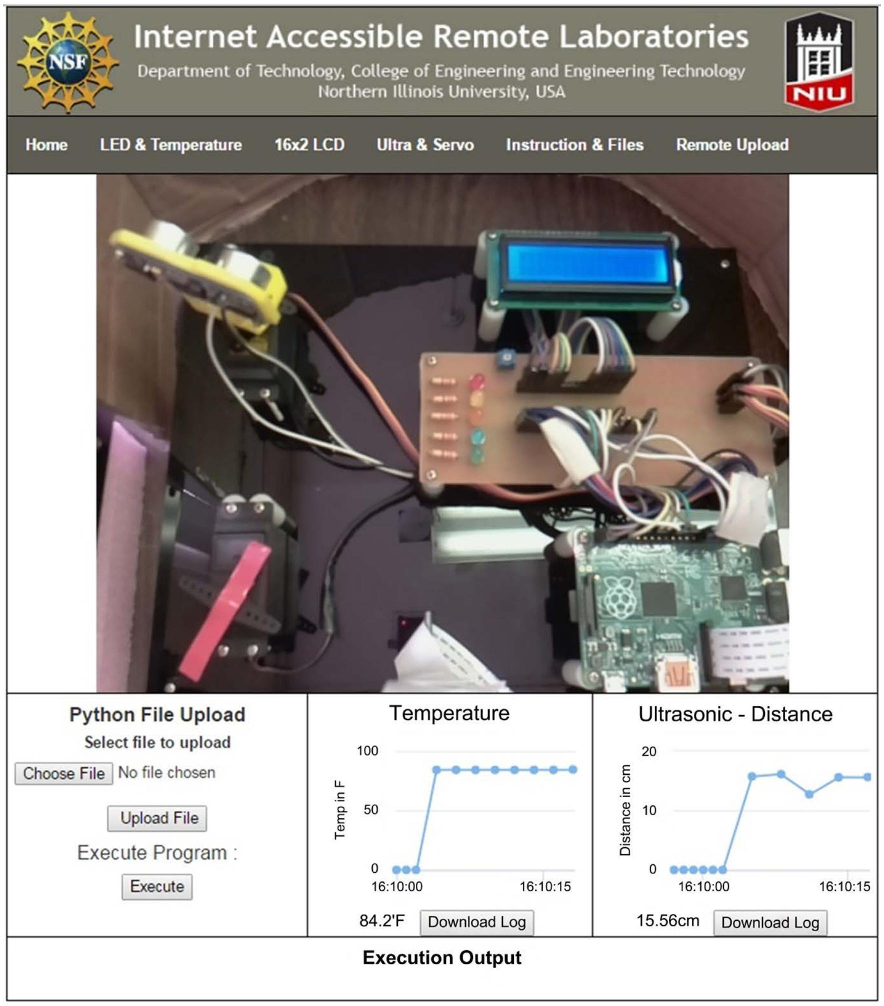
LED 1: LED OFF
LED 2: LED OFF
LED 3: LED OFF
LED 4: LED OFF
LED 5: LED OFF

Figure 10. GUI for LEDs and temperature sensor.

Another important feature of this system is remote upload and execution. The user can upload his/her own program to Raspberry Pi using the GUI shown in Figure 11. The upload form accepts only *.py format for it to execute the program file once it is uploaded. Once the "Execute" button is clicked, the uploaded file is executed and either the error or the program output is displayed in the "Execution Output" tab in the GUI. Response to the execution can be observed live in the video stream as well. The two graphs (Figure 11) show data from the temperature and the ultrasonic sensor captured over time. These data are logged into a file and are available for download by remote user to perform offline analysis of an experiment.

4.3. Structural Health Monitoring

A structural health monitoring system is illustrated within this section. The system provides an automated real-time diagnosis of structural health of an infrastructure. To implement this project a custom made laboratory scale suspended-bridge was used. Along with the physical structure, the bridge was equipped with



Internet Accessible Remote Laboratories

Department of Technology, College of Engineering and Engineering Technology
Northern Illinois University, USA

Figure 11. GUI for remote upload and data log.

a range of sensors, actuators, and an embedded processor. Figure 12 shows the overall system diagram. On-site data collection and pre-processing tasks are performed with the embedded processor. To provide client access the processed data are simultaneously passed to the cloud as well as a remote server.

The center of the cloud integration is the NodeJS server program running Heroku cloud. This server is connected with the NVIDIA TK-1 board as well as a local web server and Datastore (Figure 13). The Heroku cloud accepts/requests data from all the sensors, TK-1 board's, supporting programs, and other users [28] [29]. Sensor data are stored in Elasticsearch (Lucene) and is running in the local Datastror. Elasticsearch is a Non-SQL Datastore, which helps to provide faster indexing, fuzzy searching, and analytics of very large data. It is highly scalable compared to SQL [30].

To extract vibration data (x, y, and z) there were three accelerometers fitted with the bridge. Data from accelerometers are used to plot two graphs, one is the

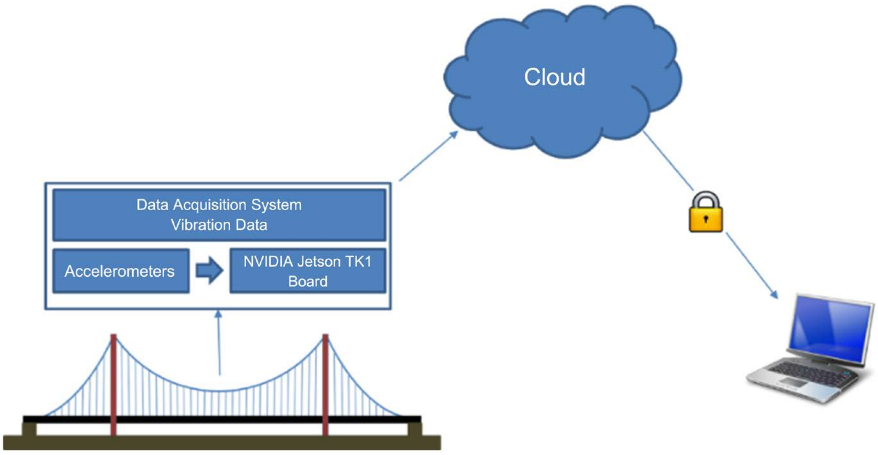


Figure 12. Overall system diagram.

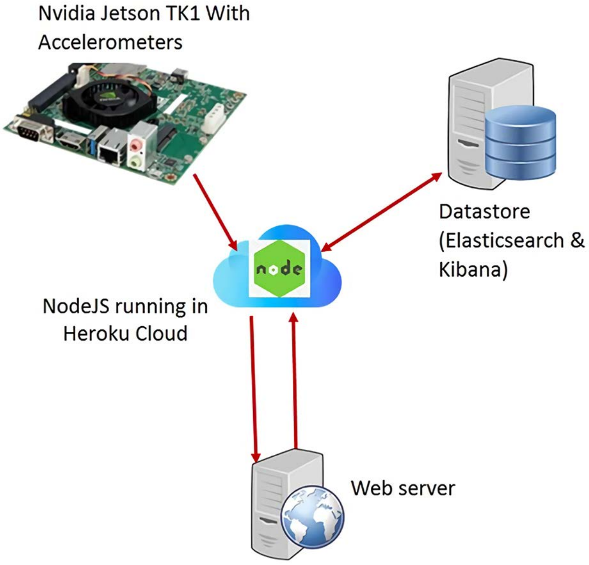


Figure 13. System integration using Cloud.

time series and the other is the spectral analysis. Graphs are plotted using Kibana, which is an open source analytics and visualization platform designed to work with Elasticsearch [31]. Kibana graphs are embedded with the web page using "IFrames". Kibana also provides advanced filtering options in the graph. In addition, it can run Elasticsearch Query DSL within the graph to apply custom filter.

Figure 14 shows two graphs for one accelerometer that will be displayed within a web browser. The left-hand side graph is showing time series for one axes and the right-hand side graph is showing its spectral analysis. Within the web page the users can choose the start date for plotting. The time series graphs are dynamic and fetch new data from the Elasticsearch Datastore every 5 seconds. Spectral analysis is done by NodeJS server running in the cloud and pushed into Elasticsearch every minute. Each time NodeJS takes recent 2048 data points from time-series data to perform spectral analysis and pushes them to Elasticsearch to be plotted by Kibana via the GUI.

5. System Utilization

The systems are built with a capacity to collect students' login and logout times along with the time taken to perform each experiment. These are used to analyze the level and timing of the facility use. This information provides a broader understanding of the students' behavior in terms of facility usage.

These data allowed comparing the learning efficiency of the control group and the test group and also the students' behavior in terms of the use of the facility. It was found that there were statistically significant differences between the test and the control groups in their time spent on the laboratory tasks, with the test

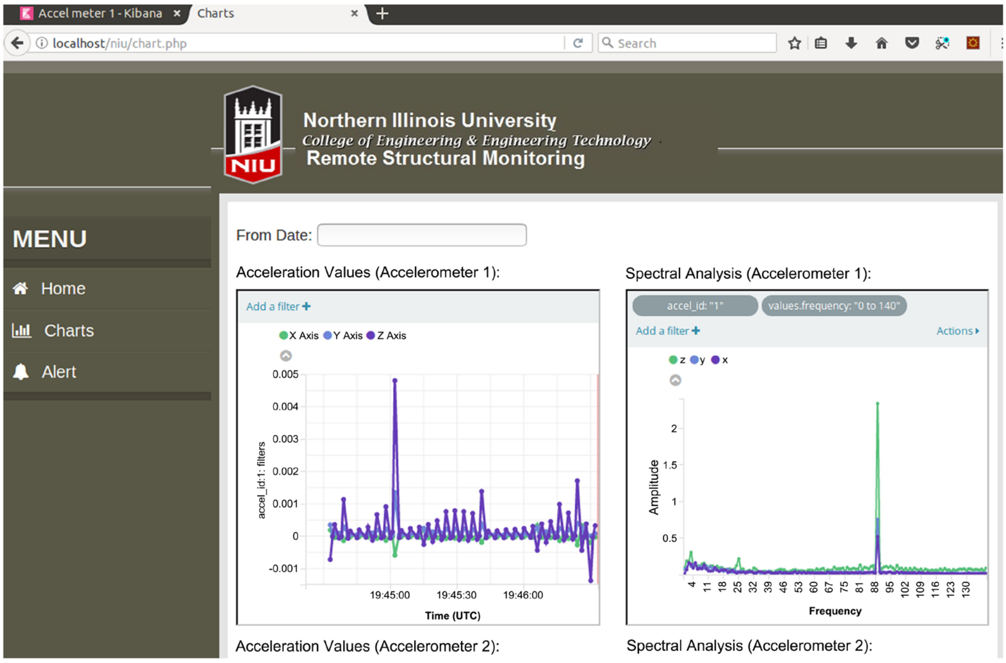


Figure 14. Vibration data with time series and spectral analysis.

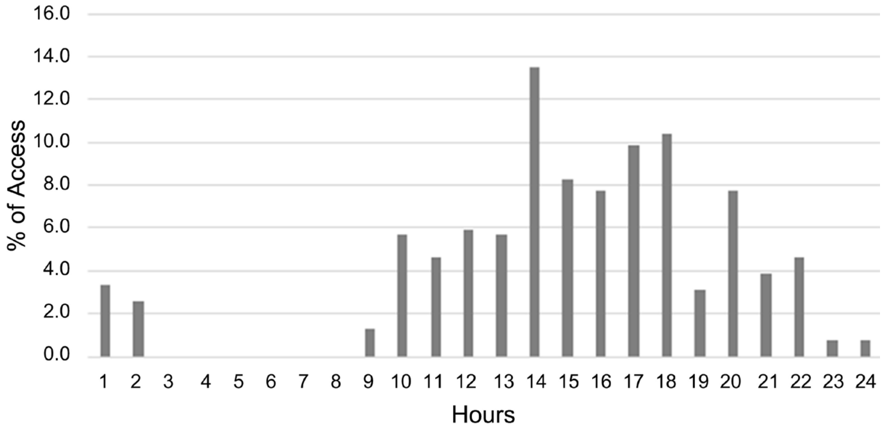


Figure 15. User access profile in terms of time of the day.

group spending 69% less time than the control group on average. It can be interpreted that the test group learned more efficiently than the control group, where the control group is performing experiments physically inside the laboratory. In terms of access time to the facility, it was found that the time of the day when students in the test group performed their laboratory tasks ranged between 9am and 2am of the next day, which indicates great flexibility and convenience for students that is otherwise impossible because of the cost and administrative limitations under a traditional laboratory configuration. Figure 15 shows the access profile to the remote laboratory experiments in terms of percentage of access.

6. Conclusion

The first part of the paper highlights the issues that need to be addressed to make the remote laboratory facilities more acceptable for the academics and organizations, as well as make these sustainable. The main factors are design with modularity, off-the-shelf products and systems to support the remote laboratory developments, and enhanced support from administration. The second part of the paper presents three remote experimentation facilities utilizing the embedded processor as the main controller. One is a remotely controlled vacuum cleaner with a remotely programmable embedded processor. The second one is a sensor and actuator system that the user can manipulate to conduct experiments within a laboratory course. This system complements a Python programming course utilizing a Raspberry Pi. Students can upload their developed programs on the remote laboratory system and verify their effectiveness. The sensors and actuators include an array of light emitting diodes, a temperature sensor, a liquid crystal display, a servo motor, and an ultrasonic sensor. A custom designed GUI was also developed so that the remote users can manipulate the sensors and actuators as desired. The third system provides an automated real-time diagnosis of the structural health of an infrastructure. For this study, a laboratory scale suspended-bridge was used along with accelerometers mounted for data collected.

Acknowledgements

The paper reports culmination of a number of projects funded by National Science Foundation (NSF) and the author likes to thank NSF for the support. The author also likes to acknowledge the valuable contribution of graduate students for these developments. Their meticulous work and dedication are keys for the success. Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the author and do not necessarily reflect the view of the NSF.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Terzis, V. and Economides, A.A. (2011) The Acceptance and Use of Computer Based Assessment. *Computers & Education*, **56**, 1032-1044. https://doi.org/10.1016/j.compedu.2010.11.017

[2] Papamitsiou, Z. and Economides, A.A. (2014) Learning Analytics and Educational Data Mining in Practice: A Systematic Literature Review of Empirical Evidence. *Journal of Educational Technology & Society*, **17**, 49-64.

[3] Miguel, A., PradaJuan, J.F., Serafín, A., Sergio, G. and Manuel, D. (2015) Challenges and Solutions in Remote Laboratories. Application to a Remote Laboratory of an Electro-Pneumatic Classification Cell. *Computers & Education*, **85**, 180-190. https://doi.org/10.1016/j.compedu.2015.03.004

[4] Lowe, D., Newcombe, P. and Stumpers, B. (2013) Evaluation of the Use of Remote Laboratories for Secondary School Science Education. *Research in Science Education*, **43**, 1197-1219. https://doi.org/10.1007/s11165-012-9304-3

[5] Mitsea, E. and Drigas, A. (2019) A Journey into the Metacognitive Learning Strategies. *International Journal of Online and Biomedical Engineering*, **14**, 4-20. https://doi.org/10.3991/ijoe.v15i14.11379

[6] Appanna, S. (2008) A Review of Benefits and Limitations of Online Learning in the Context of the Student, the Instructor and the Tenured Faculty. *International Journal on E-Learning*, **7**, 5-22.

[7] Croft, N., Dalton, A. and Grant, M. (2015) Overcoming Isolation in Distance Learning: Building a Learning Community through Time and Space. *Journal for Education in the Built Environment*, **5**, 27-64. https://doi.org/10.11120/jebe.2010.05010027

[8] Gillet, D., Latchman, H.A., Salzmann, Ch. and Crisalle, O.D. (2013) Hands-On Laboratory Experiments in Flexible and Distance Learning. *Journal of Engineering Education*, **90**, 187-191. https://doi.org/10.1002/j.2168-9830.2001.tb00589.x

[9] Rivera, L.F.Z. and Petrie, M.M.L. (2016) Models of Collaborative Remote Laboratories and Integration with Learning Environments. *International Journal of Online Engineering*, **12**, 14-21. https://doi.org/10.3991/ijoe.v12i09.6129

[10] Tsiatsos, T., Douka, S., Mavridis, A., Tegos, S., Naddami, A., Zimmer, T. and Geoffroy, D. (2014) Evaluation Plan and Preliminary Evaluation of a Network of Remote Labs in the Maghrebian Countries. *International Journal of Online Engineering*, **10**,

15-20. https://doi.org/10.3991/ijoe.v10i5.3853

[11] Santana, I., Ferre, M., Izaguirre, E., Aracil, R. and Hernandez, L. (2013) Remote Laboratories for Education and Research Purposes in Automatic Control Systems. *IEEE Transactions on Industrial Informatics*, **9**, 547-556. https://doi.org/10.1109/TII.2011.2182518

[12] Heradio, R., Torre, L. and Dormido, S. (2016) Virtual and Remote Labs in Control Education: A Survey. *Annual Reviews in Control*, **42**, 1-10. https://doi.org/10.1016/j.arcontrol.2016.08.001

[13] Maiti, A. and Tripathy, B. (2013) Remote Laboratories: Design of Experiments and Their Web Implementation. *Journal of Educational Technology & Society*, **16**, 220-233.

[14] Tirado-Morueta, R., Sánchez-Herrera, R., Márquez-Sánchez, M.A., Mejías-Borrero, A. and Andujar-Márquez, J.M. (2018) Exploratory Study of the Acceptance of Two Individual Practical Classes with Remote Labs. *European Journal of Engineering Education*, **43**, 278-295. https://doi.org/10.1080/03043797.2017.1363719

[15] Li, W. and Bai, G. (2018) Internet of Things System Based on Mobile Communication Network. *International Journal of Online Engineering*, **14**, 64-76. https://doi.org/10.3991/ijoe.v14i11.9513

[16] Azad, A.K.A., Auer, M.E. and Howard, V.J. (Eds.) (2012) Internet Accessible Remote Laboratories: Scalable E-Learning Tools for Engineering and Science Disciplines. IGI Global, Hershey. http://doi:10.4018/978-1-61350-186-3

[17] Fidan, I. and Ghani, N. (2007) Acquisition Steps of a Remotely Accessible Rapid Prototyping Laboratory. *International Journal of Computer Applications in Technology* (*IJCAT*), **30**, 266-272. https://doi.org/10.1504/IJCAT.2007.017238

[18] Fidan, I., Elliott, A., Cossette, M., Singer, T. and Tackett, E. (2018) The Development and Implementation of Instruction and Remote Access Components of Additive Manufacturing. In: Auer, M., Azad, A., Edwards, A. and de Jong, T., Eds., *Cyber-Physical Laboratories in Engineering and Science Education*, Springer, Cham, 331-342. https://doi.org/10.1007/978-3-319-76935-6_13

[19] Auer, M.E., Azad, A.K.A., Edwards, A. and Jong, T.d., Eds. (2018) Cyber-Physical Laboratories in Engineering and Science Education. Springer, Cham. https://doi.org/10.1007/978-3-319-76935-6

[20] Magdum, D.D. and Patane, R.D. (2016) Raspberry Pi Based Remote Lab Implementation. *International Journal of Innovative Research in Computer*, **4**, 15181-15185.

[21] Kalúz, M., Cirka, L., Valo, R. and Fikar, M. (2014) ArPi Lab: A Low-Cost Remote Laboratory for Control Education. *IFAC Proceedings Volumes* (*IFAC-Papers Online*), **47**, 9057-9062. https://doi.org/10.3182/20140824-6-ZA-1003.00963

[22] Fallon, T. (2013) Survey of Existing Remote Laboratories used to Conduct Laboratory Exercises for Distance Learning Courses. 2013 *ASEE Annual Conference & Exposition*, Atlanta, 23-26 June 2013, Paper ID: 7055.

[23] Limpraptono, Y., Ratna, A.A.P. and Harry, S. (2013) New Architecture of Remote Laboratories Multiuser Based on Embedded Web Server. *Remote Engineering and Virtual Instrumentation* (*REV*), 2012 9*th International Conference*, Bilbao, 4-6 July 2012, 4-11.

[24] Angulo, I., García-Zubia, J., Orduña, P. and Dziabenko, O. (2013) Addressing Low Cost Remote Laboratories through Federation Protocols: Fish Tank Remote Laboratory. *Global Engineering Education Conference* (*EDUCON*), Berlin, 13-15 March 2013, 757-762. https://doi.org/10.1109/EduCon.2013.6530192

[25] Gravier, J., Fayolle, B., Bayard, M.A. and Lardon, J. (2008) State of the Art about Remote Laboratories. *International Journal of Online Engineering*, **4**, 19-25.

[26] Python (2015) Python Software Foundation. https://www.python.org/

[27] Tornado (2015) Tornado Stable. http://www.tornadoweb.org/en/stable/

[28] Arduino Mega (2015) Overview of Arduino Mega. http://arduino.cc/en/Main/ArduinoBoardMega

[29] NodeJs (2017) About Node.Js. https://nodejs.org/en/about/

[30] Heroku (2017). https://www.heroku.com/

[31] Elastic Search (2017). https://www.elastic.co/