INTEGRATION OF ENGINEERING SYSTEMS WITHIN A REMOTE LABORATORY FACILITY

Krishna C . Vakati, Abul K.M. Azad, and Reza Hashemian
College of Engineering and Engineering Technology
Northern Illinois University

Abstract

Remote laboratories provide a new way of supplementing traditional laboratories by permitting laboratory facilities to be utilized from any place and at any time. This paper focuses mainly on the customization of two engineering systems to be controlled and manipulated from remote locations and finally to integrate them within a remote laboratory facility. The systems are Roomba Create and a small scale model house. The development involved various technical designs and study of performance characteristics. Additionally, a novel method of implementing a safety system for the remote laboratory is discussed.

Introduction

Laboratory based courses are necessary as a basic component of any practical training. Research shows students who have hands on experience in laboratory facilities have more effective problem solving skills compared to the students who have only theoretical knowledge [1]. However for various reasons, most institutions do not have adequate laboratory facilities to support hands on activities.

Traditional laboratory classes have fixed times for students to perform experiments [2, 3]. Also sharing facilities among the students makes the situation even worse. It is difficult to keep a laboratory open all the time, even though most of the equipment lies idle. To overcome such a problem, remote experimentation was introduced [4]. These facilities are cost effective and also provide better utilization of available resources. They are also useful in terms of sharing expensive resources among universities at geographically distant locations [5]. Although this paper is mainly focused on engineering systems, remote laboratory development can easily be extended to other STEM (Science, Technology, Engineering and Math) areas such as physics, chemistry, biology, astronomy, and communication. By using a remote laboratory approach in these areas, one can share experimental systems from remote locations without any human intervention. Sometimes there may be a need for running a lengthy experiment, gathering periodic data, or changing the experiment setting. All of these can be facilitated using a remote laboratory facility without being present in the experiment location.

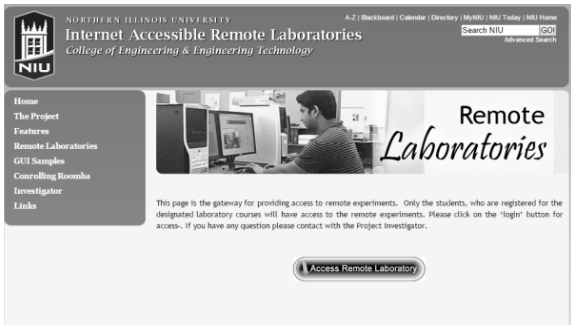
The first and foremost challenge of any remote laboratory development is identifying the experiments. Some experiments are not cost effective to convert into remote experiments [6]. Secondly, the remote user operates the laboratory equipment in all conceivable ways, so the designer needs to think of each experiment from the client’s perspective. As an example, consider a robot to be operated by a remote user. It can be damaged if the user does not operate it carefully. So it is the designer’s responsibility to think of every possible situation and provide appropriate measures to address those concerns. Furthermore, sometimes a laboratory manager needs to intervene for periodic preventative maintenance. Therefore, the challenges in the development of remote laboratory experiments vary depending on the nature of the experiment, available resources, and desired functionalities for remote users. Online laboratory learning is generally augmented by adding GUI (graphical user interface) and graphs, displaying the controller responses, and providing a real time video of the system [7].

This paper discusses the technical considerations and instructional issues that influence the complexity of the development of remote laboratories. The engineering systems considered for remote laboratory integration are Roomba

Create (RoCr) and a small scale model house. RoCr is a commercial product that was modified for its integration, while the model house was designed and developed from scratch. This paper highlights the issues and practical problems while developing remote laboratories. Also it demonstrates the developed GUI, secure login procedures, and laboratory safety features.

Related Work

The advent and wide use of the Internet has brought a new range of opportunities to sustain and expand existing networks and to create new ones. Various universities have been operating remote laboratory facilities all over the world, and new remote laboratory courses have been introduced. A few examples include work at MIT, University of Technology Sidney, University of Wyoming, Northwestern University, and Northern Illinois University [8 -10]. As an illustration, Figure 1 shows the portal for a remote laboratory facility developed within the Department of Technology at Northern Illinois University (NIU), USA [11].


NORTHERN ILLINOIS UNIVERSITY
Internet Accessible Remote Laboratories
College of Engineering & Engineering Technology
NIU
A-Z | Blackboard | Calendar | Directory | MyNIU | NIU Today | NIU Home
Search NIU
GO!
Advanced Search
Home
The Project
Features
Remote Laboratories
GUI Samples
Conrolling Roomba
Investigator
Links
Remote
Laboratories
Access Remote Laboratory

This page is the gateway for providing access to remote experiments. Only the students, who are registered for the designated laboratory courses will have access to the remote experiments. Please click on the 'login' button for access-. If you have any question please contact with the Project Investigator.

Figure 1. Remote laboratory web portal at Northern Illinois University.

The development process for an Internet accessible remote laboratory is complex, as it involves multiple disciplines: software, hardware, computer interfacing, Web development, Web security, user interface, and management [12, 13]. A number of research groups have developed remote laboratories and have demonstrated their potential in the fields of teaching, research, and industry [14]. However, there is a lack of common strategies in the developed approaches as well as difficulty transferring and integrating technologies from one laboratory implementation to another [15]. In addition, even incorporating these remote laboratories for a given STEM course requires expert skills, which makes it difficult for a faculty member to adopt them [16]. These problems affect sustainability and are a major challenge for the institutionalization of remote laboratories.

The strategies utilized for remote laboratory facility access can be differentiated into two classifications: those that need additional equipment and those that do not. Developing access to laboratory equipment such as signal generators, monitoring equipment, and oscilloscopes from the remote place by using the software and additional hardware come under the first category [17]. The second category involves completing the software base and using a client-server paradigm. Although the second one is more practical and expandable, it needs an expert developer to code the complex algorithms to maintain the laboratory programs [18]. The first category was chosen for this research.

Through research on remote laboratory software, it was decided to use the LabVIEW package from National Instruments for the remote laboratory development. LabVIEW is a user friendly environment that contains an interface for easy communication with sensors and actuators while integrating all the components as a single system [19].

In addition, it has a powerful GUI development tool along with an in-built web publication feature. The web publication feature allows the user to access a GUI for remote laboratory applications (Figure 2) [20, 21].

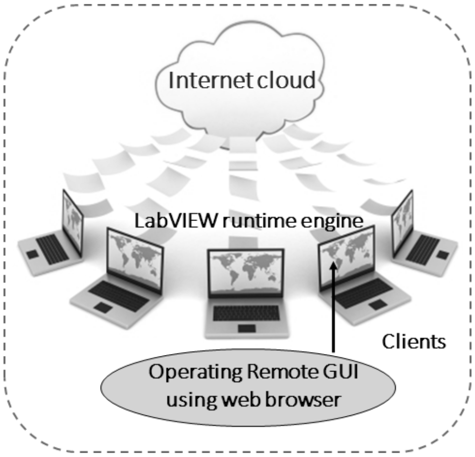


Figure 2. LabVIEW remote development feature.

Integration of Roomba Create

The RoCr is a complete robot development system that allows one to program new robot behaviours without thinking much about mechanical assembly and low-level code. The system was developed by iRobot RoCr's Open Interface (OI), which provides a set of commands, such as drive, demo commands, song, and sensor commands, etc [22]. These can be used to develop new behaviours, add third party electronics, and write Open Interface-based programs to control the RoCr using serial communication. Figure 3 shows the iRobot RoCr without any modification.

Figure 3. Roomba Create.

Modification of RoCr

The RoCr was modified so it could be controlled wirelessly from a computer via a GUI. Provisions were made to implement the GUI over the Internet so the user can control the RoCr from a remote location. In terms of operation, the RoCr can be operated in two modes, Manual and Auto. An image of the modified RoCr, along with the integrated IR sensors and Arduino board, is shown in Figure 4.

In Manual mode, the user can direct the RoCr using the GUI, while in Auto mode, the RoCr can travel within a given space where infrared sensors (IR) are used for obstacle avoidance. In terms of modification, a number of IR sensors are attached to the RoCr and an Arduino mega microcontroller board is used to perform the Auto mode operations.

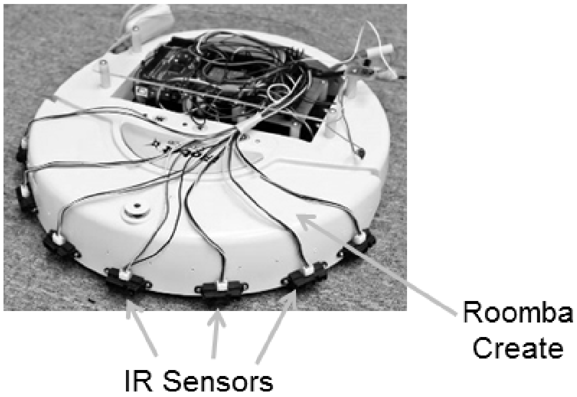


Figure 4. RoCr with integrated IR sensors.

LabVIEW was used to integrate the RoCr into the remote laboratory facility. It has a built-in remote operating feature, which makes remote accessing simple and efficient. Bluetooth was used as a wireless link between the RoCr and the host computer, which makes the RoCr a flexible mobile robot within a given area. A GUI was developed for the user interface. Additional hardware was designed to deal with the IR sensors as well as to implement the Auto mode operation of the RoCr.

For real-time video, an Internet protocol (IP) camera was used, which works independently from LabVIEW. It reduces the complexity of the processing system. The IP camera allows the users to view the travelling path of the RoCr. The complexity of the GUI was also reduced by separating the Auto and Manual modes within the GUI. Provisions were made for one or more individuals to view the operations of RoCr, but only one person at a time was allowed control.

System Block Diagram

The block diagram shows the details of the implementation of the modified RoCr system (Figure 5). There are three main components: host computer, RoCr, and Internet access. The host computer holds the LabVIEW GUI and communicates with the RoCr via Bluetooth. An IP camera was also used to provide real-time video of the travelling path of RoCr.

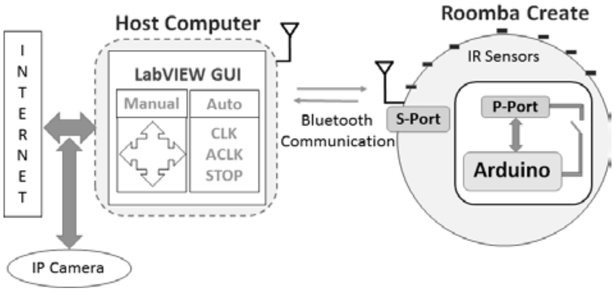


Figure 5. Block diagram of modified RoCr.

Figure 6 shows the physical layout for the modified RoCr. In addition to the in-built hardware, the modified RoCr consists of 7 IR sensors, a 12V battery, a Bluetooth receiver and an Arduino mega.

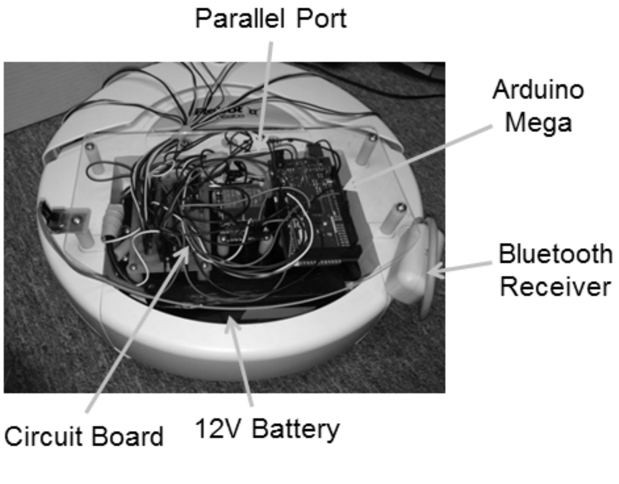


Figure 6. Physical layout of RoCr.

Manual Control

To control the RoCr manually, the user must use the arrows within the middle part of the GUI (Figure 7). By pressing each arrow, a set of predefined commands passes serially in an array format to the RoCr. These commands were made to perform a particular task defined in the RoCr manual. The combination of commands performs various tasks such as forward, reverse, left, right, speed adjustments, radius adjustments, etc. Apart from that, users can visualize the sensor status like Right bump, Left bump, Left wheel drop, Right wheel drop, Virtual wall, Right wheel over current, etc. Figure 7 depicts the GUI with all the control buttons/arrows and sensor status indicators mentioned above.

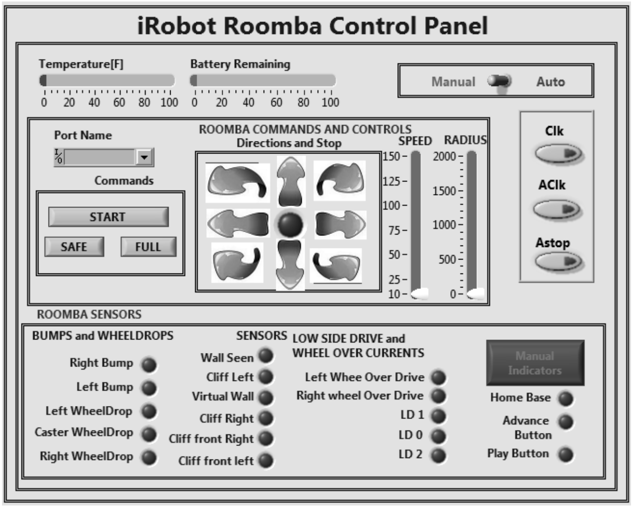


Figure 7. GUI for RoCr control.

Auto Control

The user can switch the RoCr to Auto control by pressing the Auto tab shown in Figure 7. By doing that, the RoCr moves with a program called obstacle avoidance. Obstacle avoidance is developed in the Arduino board to avoid the obstacles. The block diagram in Figure 8 shows the hardware implementation of the Auto mode.

A number of hardware modifications were used to implement the Auto mode. When the user chooses the Auto button, it switches the Auto enable in the RoCr parallel port to high. The RoCr parallel port, which is connected to the driver circuit, triggers the Arduino power so the output data transfers from the microcontroller TX0 to the RoCr RXT. Similarly by enabling the mode selection in the parallel port, the user can change the control strategy of the Auto mode.

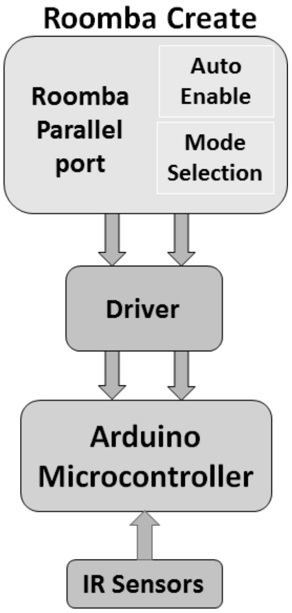


Figure 8. Block diagram Auto mode implementation.

Control Strategies in Auto mode

Clockwise avoidance: This implements motion of the RoCr in a clockwise direction as a diversion when the IR senses an obstacle.

Counter-clockwise avoidance: This is similar to the previous one, but in this case, the RoCr moves in a counter-clockwise direction when IR sensors detect an obstacle.

Switching between Auto and Manual modes

Figure 9 illustrates the flowchart for the process of controlling the RoCr. By default, the RoCr is in the Manual mode, so the user can control buttons like forward, left, right, back, speed, radius, etc. Also, the user can observe the sensor status, as mentioned, within the ‘Manual Control’ section. To switch to the Auto mode, the user needs to send the command 899307 via the serial port. Once the command reaches the RoCr, it switches the RoCr to the Auto mode by enabling the VIN pin in the Arduino. Then the Arduino will take control over the RoCr and start communication with it.

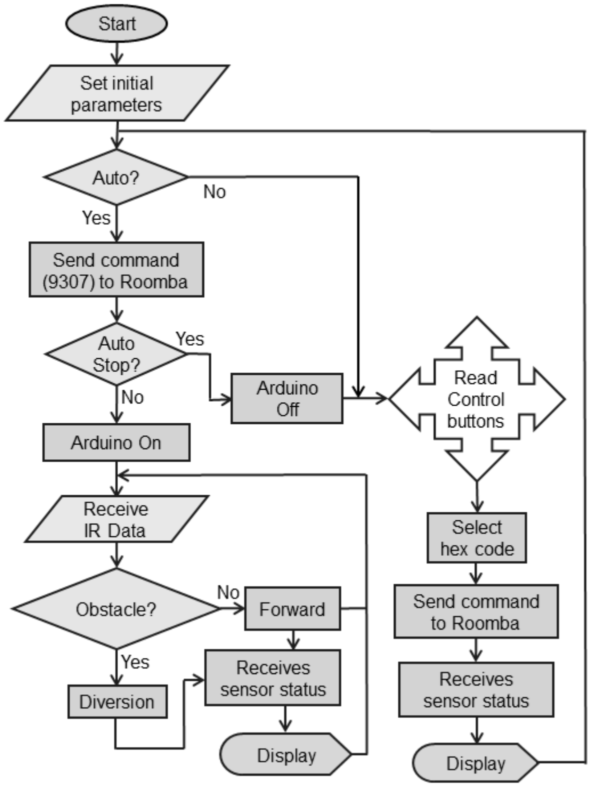


Figure 9. Flowchart for control process of RoCr.

It is difficult to switch it back to the Manual mode from the Auto mode without losing the serial connection, as the RoCr is already continuously communicating with the host computer. To break the existing communication between the RoCr and the Arduino without losing the serial connection, we needed to send sequential data with a high data rate in a repeated fashion, where the data rate was much higher than the Manual mode. In this way, it was possible to discontinue the communication between the RoCr and the Arduino without losing the Bluetooth serial connection between the host computer and the RoCr. To implement this, a stop button is provided within the GUI to send a set of repeated data. When it reaches the RoCr via the serial port and disables the Arduino, it sets the RoCr back to the Manual mode.

Integration of Small Scale Model House

The small scale model house is the second experimental system developed and integrated within the remote laboratory facility (Figure 10). The control system for the house was designed to maintain a desired indoor temperature. The house has an active heater and a passive cooling system. Passive cooling uses a fan and opens the door. With this arrangement the inside temperature of the house can be maintained at the ambient temperature or higher. There is an indoor temperature sensor and an external light level sensor to control a light inside the house. A GUI was developed to allow remote users to manipulate the temperature control and view the indoor temperature (Figure 11).

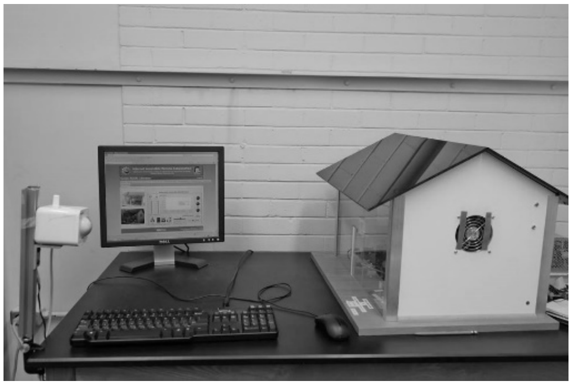


Figure 10. Small scale model house.

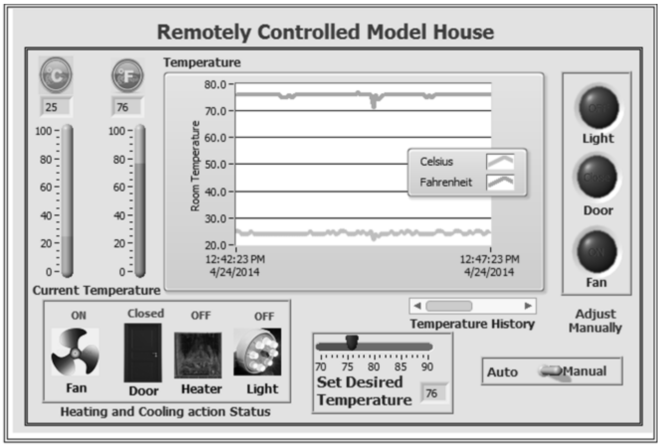


Figure 11. Image of the GUI.

Hardware Description:

The block diagram for controlling the small scale model house is shown in Figure 12.

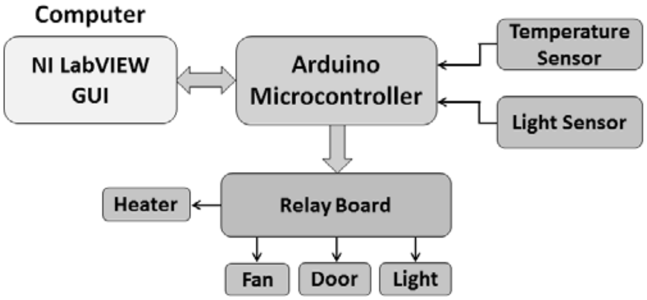


Figure 12. Hardware block diagram.

There are two input sensors for the measurement of the house’s indoor temperature and the outside light level. The analog input pins of the Arduino board are used to collect the sensors’ data. Sensor signals are then passed to LabVIEW for control decisions. LabVIEW is also used to design a GUI for a user interface. The decisions are then passed to the Arduino for controlling the fan, light, door, and heater. Since the Arduino cannot drive these actuators directly, a set of relays is used for driving the actuators.

System Control

The purpose of this project is to monitor and control the model house from a remote location. The model house has two modes of operation. A flow chart for the model house control is shown in Figure 13.

In the Auto mode, once the user sets the desired temperature, the cooling and heating mechanism automatically works to make the current temperature equal to the desired temperature. The user is able to view which mechanism is taking control. When the desired temperature is less than the actual temperature, the fan will turn on and the door will open to cool the model house. Similarly, when the desired temperature is greater than the actual temperature, the heater will turn on to increase the current temperature. Thus, the balance can be achieved in one stage.

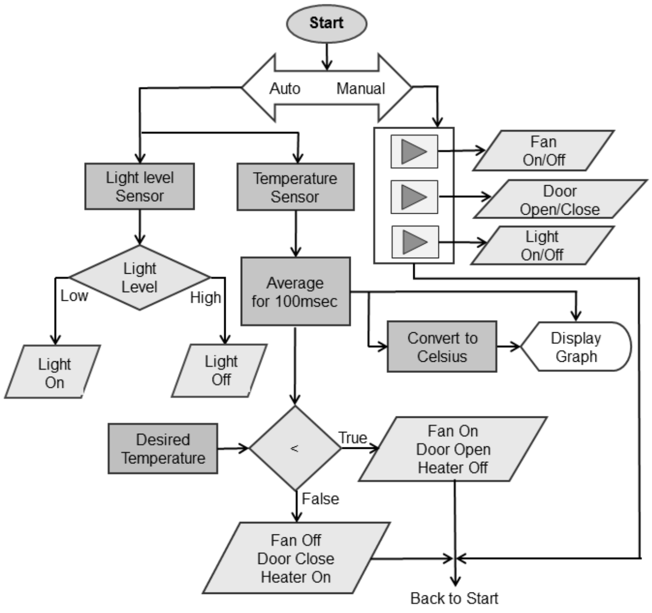


Figure 13. Flow chart for control mechanism.

There will be an offset around two degrees when the actual temperature is equal to the desired temperature. This is used to avoid fluctuations in the heating and cooling mechanism. In terms of light control, the light will be 'Off' once the light sensor detects higher light levels. The light will be 'On' when the light level is detected as below a threshold. In the Manual mode, the user can control the fan, light, and door manually using the control buttons on the GUI (Figure 11). There is a toggle button on the GUI to switch between the Auto and Manual modes.

Performance characteristics of the House model

The performance of the house model was determined by analysing the following issues:

Safety*:* This is an important factor in any remote laboratory application. In this project, a safety alert system was developed to protect the house from overheating. A simple algorithm was developed in LabVIEW especially for safety purposes. This algorithm links the model house with a Gmail server so whenever the temperature is above the overheat setting, it will send an alert email to a concerned authority. Overheating may happen due to hardware malfunctions, wiring problems, and sometimes even an operational problem.

Figure 14 shows the LabVIEW wiring diagram for implementing the alert system. It automatically sends an email with a body and subject alert. The alert mechanism works in parallel with the model house operation so it will not affect the performance of the model house. Figure 15 shows a sample email from the automated alert mechanism.

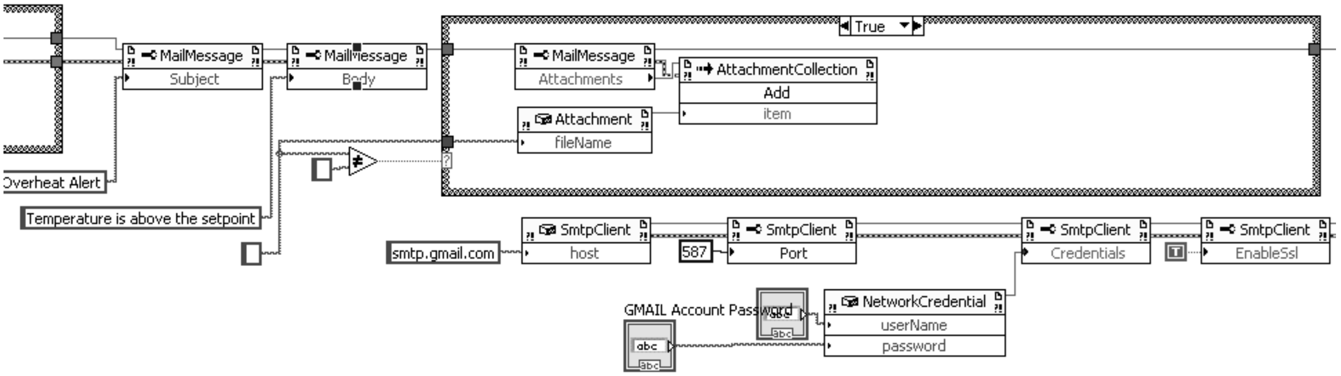
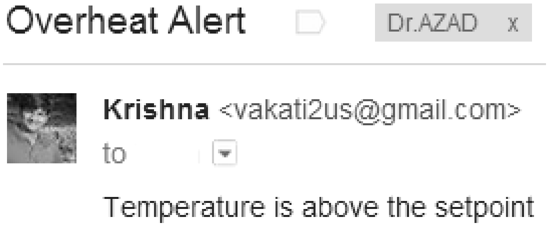


Figure 14. Safety alert system implementation in LabVIEW.



Overheat Alert

Dr.AZAD x

Krishna <vakati2us@gmail.com>
to

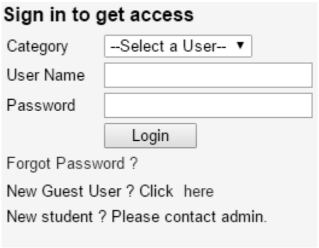
Temperature is above the setpoint

Figure 15. Mail alert for overheating.

Reliability: Reliability, or consistency of a system, is characterized in different ways. Hardware reliability depends on the selection of the components, amplification, number of parts used, etc. Programming reliability was based on the number of loops, processing time, complexity of the code, etc. The systems were designed using as few components as possible. Most of the I/O operations as well as the control implementation were performed within the Arduino microcontroller. This processor was much more reliable in terms of operation. Programming of LabVIEW was completed at various levels to improve the consistency of the overall system. Even though the Auto and Manual modes of the model house are integrated within one GUI, the coding is separated for each mode and for each sensing mechanism.

Maintenance: The systems were stable in performance and operated for months without any difficulty. Most problems happened during power fluctuations.

Web Security: The remote laboratory facility used a web portal that is password controlled. Each user needed to be registered by the administrator. In addition to this, the portal collected all data regarding user access. An image of the application page for the login process is shown in Figure 16.



Sign in to get access

Category --Select a User-- ▾
User Name
Password

Login

Forgot Password ?
New Guest User ? Click here
New student ? Please contact admin.

Figure 16. Access control to remote lab.

Hosting GUIs on the Web

As an inherent feature of LabVIEW, the developed GUI can be launched as a web page using the web publishing tool. This web page can then be accessed using any web browser that already has the LabVIEW runtime engine installed in it. All the control and monitoring facilities of the GUI are available to the remote computer over the Internet. Figure 17 shows a webpage displaying the small scale house along with its GUI. This page was available online to control the model house from any place at any time.

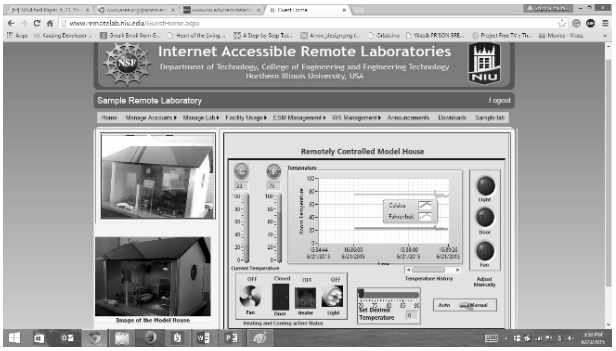


Figure 17. Webpage on remote computer.

Conclusion

This paper provides an account of modifying the RoCr and development of a small scale model house so that they could be integrated within a remote laboratory facility for remote access. The process involved understanding the basic hardware functionalities and added new hardware and software features. It also discusses various technical advancements, and performance characteristics involved when developing remote laboratories. Finally, these developed remote laboratory experiments were integrated within an existing learning management system that was developed within NIU.

Acknowledgement

The authors would like to thank the National Science Foundation for its support for the reported work. This paper is based on an NSF TUES (Transforming Undergraduate Education in

Science) project, award number DUE-1140502. Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. E. Cindy and Hmelo-Silver, "Problem-Based Learning: What and How Do Students Learn?" *Educational Psychology Review, 16*(3), Sept 2003.

2. A.P. Boyle, D.N. Bryon, and C.R.C. Paul, "Computer-based learning and assessment: A paleontological case study with outcomes and implications," *Computers and Geosciences, 23*(5), 573-580, 1997.

3. G. Bengu and W. Swart, "A computer-aided total quality approach to manufacturing education in engineering," *IEEE Transactions on Education, 39*(3), 415-422, 1996.

4. T.K. Grose,"Can distance education be unlocked," *PRISM*, 19-23, 2003.

5. P. Orduna,"Shared the remote laboratories among different institutions: A practical case," *Remote Engineering and Virtual Instrumentation (REV), 9th International Conference*, Bilbao, 4-6 July 2012.

6. K. Jona, R. Roque, J. Skolnik, D. Uttal, and D. Rapp, "Are Remote labs worth the cost?," *International Journal of Online Engineering*, 48-53,2011.

7. N. Zorica, M. Jan, and N. Andrew, "Remote Laboratories versus Virtual and Real Laboratories," *33rd ASEE/IEEE Frontiers in Education Conference,* Boulder, CO November 5-8, 2003, Session T3E.

8. V.J. Harward, J.A. Del Alamo , S.R. Lerman, P.H. Bailey, J. Carpenter, K. DeLong, and D. Zych, "The iLab shared architecture a web services infrastructure to build communities of Internet accessible laboratories," *Proceedings of the IEEE*, 931-950,2008.

9. L Gomes, and J Garcia-Zubia, *Advances on remote laboratories and e-learning experiences* ed., Bilbao, Spain: Univ.Deusto 2007.

10. J.L. Hardison, K. DeLong, V.J. Harward, J. Del Alamo, R. Shroff, and O. Oyabode, "Enabling Remote Design and Troubleshooting Experiments Using the iLab Shared Architecture," *ASCE Earth and Space Conference 2010.* Honolulu, HI, 2010.

11. A.K.M. Azad, "Internet Accessible Remote Expermentation with Integrated Learning Management System" *ASEE Annual Conference.* Altanta, USA, 2013.

12. S.K. Esche and C. Chassapis, *Internet Accessible Remote Laboratories: Scalable E-Learning Tools for Engineering and Science Disciplines*, Eds. Azad, Auer, and Harward, IGI Global 2012.

13. A.K.M. Azad, M.E. Auer, and J. Harward, (Editors). *Internet Accessible Remote Laboratories: Scalable E-Learning Tools for Engineering and Science Disciplines,* IGI Global, Pennsylvania, 2011.

14. T.A. Fjeldly and M.S. Shur, *Lab on the web—Running real electronics experiments via the Internet*, Hoboken, NJ, 2003.

15. N. Aliane and R. Pastor,"Limitations of remote laboratories in control engineering education," *International Journal of Online Engineering*, 31-43, 2008.

16. R. Hashemian, "A Method to Design, Construct and Test Digital Hardware all in Classroom Environment." *Proceedings of*

the FIE2007 Conference, 11-13. Milwaukee, WI. 2007.

17. R. Hashemian, “Creating an Internet-Based Remote Access Technique to Construct Digital Devices in Classroom Environment*.”* Northern Illinois University, Electrical Engineering. *CIUE*, 2006.

18. R. Hashemian,” A Low-Cost Server-Client Methodology for Remote Laboratory Access for Hardware Design.” *39th ASEE/IEEE Frontiers in Education Conference.* San Antonio, TX: IEEE 2009.

19. G. Heimbach, (n.d.). *Creating Quality UIs with LabVIEW.* Retrieved from http://www.ieee.li/pdf/viewgraphs/creating_quality_uis_with_labview.pdf.

20. P. Kaushik. “Developing Control Experiments as a part of Remote Laboratory Facility,” *121st ASEE Conference & Exposition.* Indianapollis, IN, 2014.

21. A.K.M. Azad, A. Otieno, O. Ghrayeb, and N. Anand, “Internet Based Experiments for Physical Laboratory Set-up,” *Proceedings of the 2003 American Society for Engineering Education Annual Conference.* American Society for Engineering Education, 2003.

22. Irobot, ”Irobot create open interface,” Retrieved http://www.irobot.com/file library/pdfs/hrd/create/Create%20Open%20Interface_v2.pdf,2006, (accessed August 2013).

Biographical Information

Krishna C. Vakati is a graduate student in Electrical Engineering at Northern Illinois University and has worked as a teaching/research assistant for the last two years. He has considerable experience conducting laboratory classes as well as research work on remote laboratories. Since 2009, he has been an active member of the IEEE. He has organized, volunteered at and participated in several IEEE events while at NIU. During his stay at NIU, he has made contributions to a number of externally funded projects.

Abul K. M. Azad is a professor with the Technology Department at Northern Illinois University. He has a Ph.D. in Control and Systems Engineering and an M.Sc. and B.Sc. in Electronics Engineering. He has been in academics for 15+ years, and his research interests include remote laboratories, mechatronic systems, mobile robotics, and e-Learning. In these areas, he has over 100 refereed journal and conference papers, edited books, and book chapters. So far, he has attracted approximately $1.7M in research and development grants from various national and international funding agencies. He is a member of the editorial board for a number of professional journals as well as an Editor-in-Chief of the International Journal of Online Engineering. He is active with various professional organizations (IEEE, IET, ASEE, and ISA) as well as a member of the Board of Trustees of the CLAWAR Association. He has served as chair and co-chair of numerous conferences and workshops, in addition to serving on the program committees of approximately 30 international conferences. He is a project proposal reviewer with various national and international funding agencies in the US, Europe, and Australia.

Reza Hashemian is a professor in Northern Illinois University's Department of Electrical Engineering, where he teaches and does research on analog circuit design, digital designs with FPGAs, VLSI designs, image compressions and processing, and computer arithmetic. In earlier positions, he served as the director of research and industrial contact at Sharif University of Technology and also as the associate director and then director of Materials and Energy Research Center (MERC), Tehran, Iran. He has worked in industry for six years, including Signetics (now Phillips), Inc. and Texas Instruments, Inc. In the past, he held a professional engineer license in Illinois. He was a member of the first executive committee of the IEEE, Iran Section, 1970-1972, and now is a lifetime member of the IEEE. He received his B.S. from Tehran University, Iran, and his M.S. and Ph.D. from the University of Wisconsin, Madison.