

New Architecture of Remote Laboratories Multiuser based on Embedded Web Server

<http://dx.doi.org/10.3991/ijoe.v9i6.2886>

F. Yudi Limpraptono, Anak Agung Putri Ratna, Harry Sudibyo
University of Indonesia, Depok, Indonesia

Abstract—This paper presents a new architecture for multi-user remote laboratory based on an embedded web server used for experiment with the MCS-51 microcontroller system. The design for the remote lab uses a green computing approach, with the aim of reducing energy consumption, lowering the cost of procurement systems, improving performance and use, and also saving space. The remote lab system uses multi user and multi device architecture to support collaborative work and improve scalability. The design of the remote lab has n modules experiment, and each module can be accessed by a single user or a group which consists of several user. The prototype of a remote lab that has been realized consists of an embedded web server based on Raspberry Pi as the remote lab gateway and 2 experiment modules that are controlled by an embedded web server based on MCS-51 microcontroller. Development of the system is expected to contribute to creating an efficient remote lab that is able to facilitate collaborative work and support green computing.

Index Terms—Remote Laboratories, Embedded Web Server, Microcontroller, Multiuser, Collaborative Work, Green Computing

I. INTRODUCTION

A remote lab is a web-integrated system that can be accessed through a unified interface by a web browser[1]. By providing distance users a way to share resources and tools, it may cause cost reduction and access to some more specialized learning resources that are not available locally [2].

There are several advantages to the use of remote laboratories, as laboratory performance will be better and more efficient because students can use laboratory equipment for 24 hours. A remote laboratory creates autonomous learning [3], allows use by handicapped students [4], and supports resource sharing and collaboration between laboratories. In Indonesia, particularly, a remote lab system needs to be developed to share the laboratory (collaboration) between colleges. A remote lab can also be used to support the implementation of teaching and learning for the Open University in Indonesia [5].

One published strategy that is often used in building a remote laboratory, which has been published, is use of a computer-based server, but investment costs for hardware, software and maintenance are very expensive. Furthermore, the total electrical energy consumption by server, computer and cooling systems in the remote lab systems are very significant. Energy consumption for a computer system contributes to the increase of greenhouse gas

emissions. Each PC generates about a ton of carbon dioxide every year[6].

Web technology based on embedded system has grown rapidly in the post PC era [7]. Implementing a web enabled controller to the system replaces the computer as a web server, thus lowering the cost of system [8]. Applied embedded system supports the concept of green computing because the design has addressed several issues related to green computing such as reduced power consumption, reduced costs and space saving [6].

Various technologies in web programming have been applied to provide a comfortable remote lab environment, such as socket, applet, ajax, corba, labview, etc.[4]. The work in [4] describes current progress toward the micro-server AJAX web-based solution. This solution will be web-based, firewall-safe, more scalable, and support cooperative work among group members. N groups of user from any client platform will be able to access simultaneously any of the N networked programmable devices. The work in [2] describes an approach for the development of an embedded web server that facilitates a remote access for an electronic engineering laboratory experiment for distance learning. The system architecture developed for this project allowed use of low cost components.

An ideal remote lab must be able to meet the needs of users, for example, by providing experience similar to doing experiments in a conventional laboratory. Experiments are usually conducted in a conventional laboratory, usually consisting of a group of students who has to be able to experiment with the same module. This condition is what is known as the concept of collaboration or teamwork, where one student with other students can communicate and hold discussions. The concept of collaboration is recommended by ABET [9]. The work in [10] describes a client server architecture based on web services and .NET remoting services for an integrated learning environment with remote experimentation that allows students in disparate locations to simultaneously and collaboratively complete complex experimental exercises.

From the problem above, this research proposes a new architecture of remote labs that is efficient and has high scalability that can support collaborative work. Remote lab architecture is implemented with an embedded system technology to improve efficiency. The architecture of the remote lab uses multiuser-multidevice models to improve system scalability, where the remote lab has n modules and each module can be accessed by a single user or a group that consists of several users simultaneously.

This paper is organized as follows: section II contains the new architecture of remote lab; section III contains a

design hardware involving an embedded web server and lab module; section IV contains design software for the remote lab; section V provides results and discussion; and section VI concludes the paper.

II. ARCHITECTURE OF THE REMOTE LAB

Architecture of the remote lab is an extended and modified version of a paper presented at the International Conference on Remote Engineering & Virtual Instrumentation (REV2012), held in Bilbao, Spain, July 4-6, 2012. The paper published in [11] describes an architecture of the multiuser remote lab based on a hybrid server. The system has two kinds of web servers – that is a web server remote lab gateway based on a PC server and an embedded web server (EWS) based on an MCS-51 microcontroller for control experiment modules. Each experiment module in the remote lab that has been realized can only be accessed by one user at any time.

A. Design of Remote Lab System

A diagram of the remote lab architecture is shown in Figure 1. The system consists of a web server remote lab gateway based on an ARM 11 embedded System; two experimental modules for an MCS-51 microcontroller family, which is controlled by an embedded web server based on AT89S52; IP camera; and several computers on the client side. All the equipment is connected to a switch hub.

The web server for the remote lab gateway is implemented with an embedded system using the Raspberry Pi pico computer with an ARM1176JZF-S processor. The embedded system has 256 MB of RAM, 4GB of external memory, available various I/O ports and an Ethernet port 10/100. The embedded system using the Raspbian embedded Linux operating system. Raspbian is an operating system based on Debian Wheezy [12].

The web server remote lab gateway contains several software applications such as Apache web server, a MySQL database server and a remote lab user management application based on PHP. The user management application functions to handle user requests that will access the remote lab and manage all remote lab resources.

The experiment module is an evaluation board AT89SXX microcontroller and is equipped with some I/O modules such as an LED display, a LCD display, keypad and motor stepper. Users can interact with remote experiment modules such as uploading binary files to the target board, observing the changes via a web cam and interacting with the target board through a virtual I/O.

The main function of the embedded web server is to control the experiment module and become the user interface between the user and experiment modules. Furthermore, embedded web server can send log report to the database server and send a message to the chat application.

In this project the number of modules is limited to only 2 experiment modules, so every time the remote lab can serve two groups simultaneously. Each group can consist of more than 2 users. However the system can be expanded to handle a number of n modules.

B. Use Case Diagram

The design of the remote lab system can be described using use case diagram in Figure 2. There are two main components, namely the use of case diagram defining the actor and the use case. Actors are people, process or other systems that interact with information systems. Use case is the functionality provided as a system of units that exchanges information.

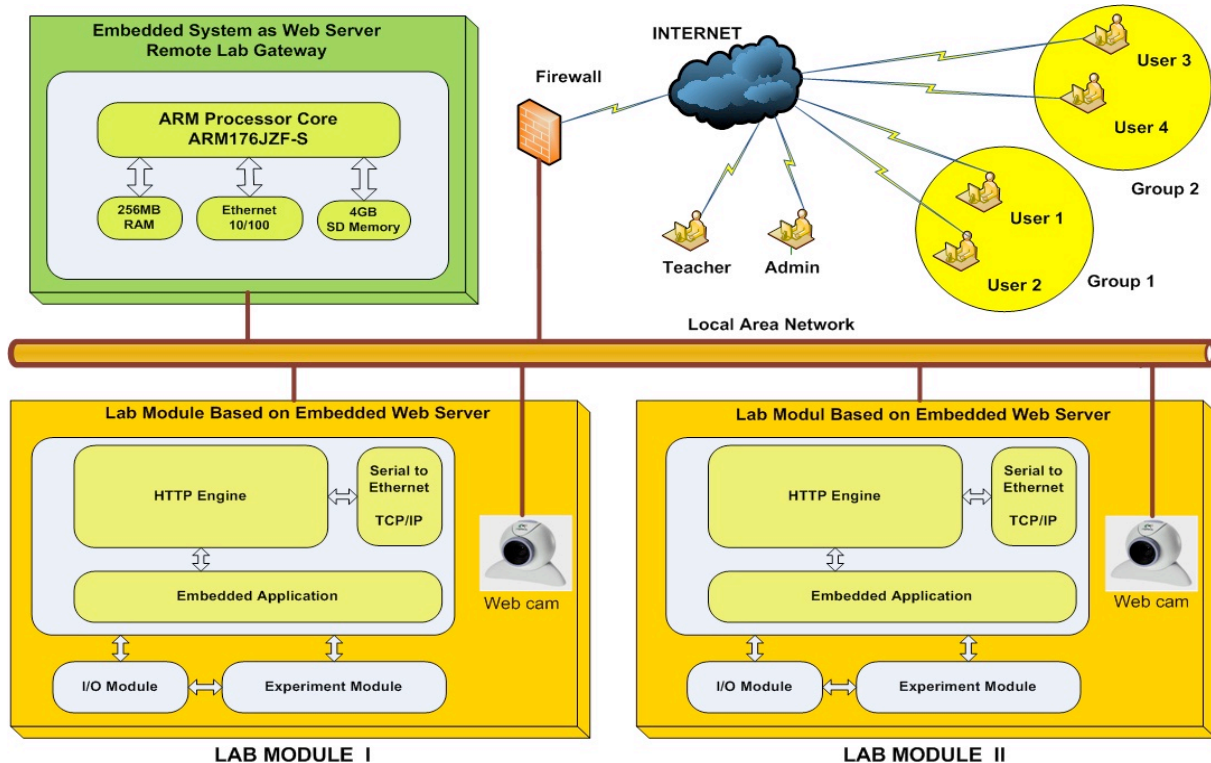


Figure 1. Remote Lab Architecture

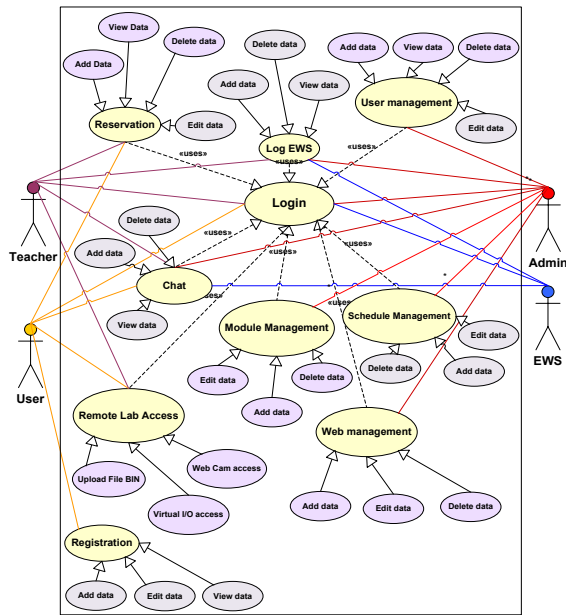


Figure 2. Use Case Diagram

Definition of an actor in a remote lab system can be described as follows:

1. Admin is an actor (person) who is in charge and has permission to perform operations on user data management, web content management, EWS log data management, and can communicate via a chat application with remote lab users.
2. Teacher is an actor (person) who has permission to perform operation to view the EWS log report and also to perform remote lab access operations for the purpose of learning. The teacher can also communicate with the user and admin via a chat application.
3. Users are actor (person) who have permission to access the remote lab. Users can access the experiment module and can communicate with admin, teacher or other users using chat application.
4. EWS is an actor (system) with permission to send an event log to the database server, and it is capable of sending log messages to the chat application.

The description of use case that has been defined in the remote lab system can be described as follows:

1. Login is a process of identifying and authenticating the remote lab user.
2. User management is a process to manage user data. The process includes edit data, add data, view data and delete data.
3. Web content management is a process for managing web content of the remote lab. The process includes edit content, add content and delete content.
4. Registration is a process of handling the registration of new users of the remote lab. The process includes add data, edit data and view data.
5. Reservation is a process of booking for the lab activities. The process includes insert data, edit data, delete data and view data.
6. Schedule management is a process for managing the schedule of remote lab. The process includes edit schedule, add schedule and delete schedule.

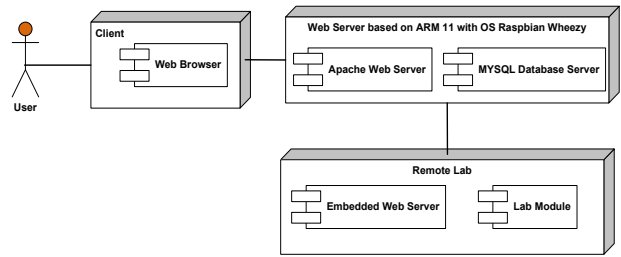


Figure 3. Deployment Diagram

7. Module management is a process for managing experiment modules, including edit module, add module and delete module.
8. Remote Lab access is a process to access the lab module, including a binary file upload, access virtual I/O and access the webcam.
9. Chat is a process for communication between actors. The chat application will be active if the user has successfully login.
10. EWS log report is a process of storing and displaying the log event data that sent by the EWS.

C. Deployment Diagram

The design of the deployment diagram is shown in Figure 3. The deployment diagram shows the configuration of components in the execution of the application. The diagram consists of three packages, namely client, server and remote lab. The client package contains one node, which is a web browser application; server package consists of two nodes, namely Apache web server and MySQL database server; remote lab package consists of two nodes, namely embedded web server and lab modules.

D. Activity Diagram

Remote lab system activity diagram is shown in Figure 4. The activity diagram is divided into 4 sections (swimlane) according to the number of actors involved in the system: admin, teachers, users and EWS.

1. Admin is the user with the highest privileges on the remote lab system. Admin must log in to enter the system. If the login is successful, then the admin can access the administrator menu, such as user management menu, schedule management menu, module management menu, remote lab access menu, log view menu and chat menu.
2. Teacher must login to enter the remote lab system. If the login is successful, then the teacher can access the teachers menu, which consists of the chat menu and log view menu. If the teacher wants to access the remote lab, they must make a reservation to set the date, time and experiment modules.
3. User must make the process of registration and reservation to be able to access the remote lab. If the login is successful, they can access the remote lab menu that consists of chat menu, upload BIN file menu and access I/O menu.
4. EWS (embedded web server) is a controller experimental module. EWS can send messages to the database server or chat application.

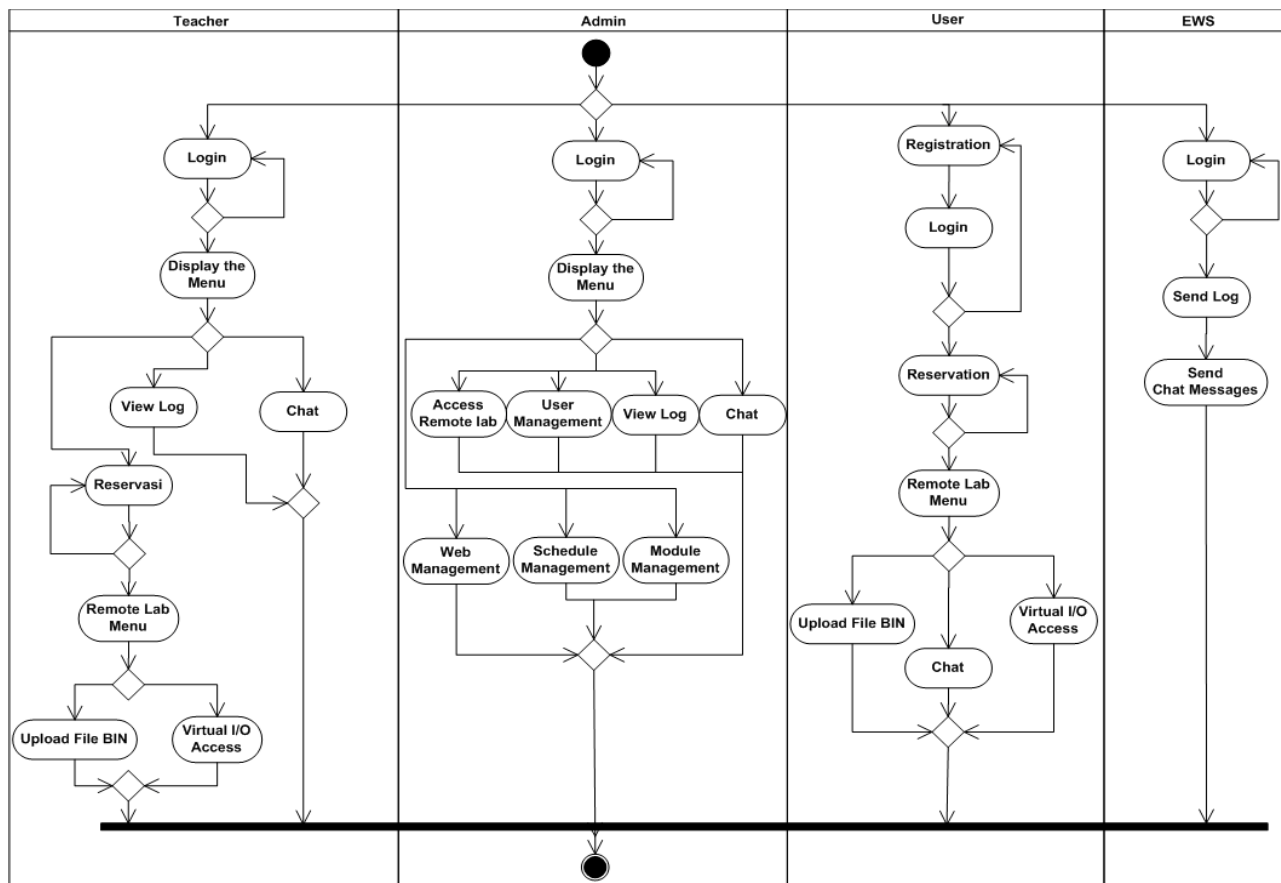


Figure 4. Activity Diagram

E. Sequence Diagram

A sequence diagram describes the behavior of objects in a use case by describing the object's life time and messages sent and received between objects. The number sequence diagram to be described is as many as the defining use case available within a system. This paper will only discuss the sequence diagram associated with remote laboratory access in collaboration.

Remote lab access sequence diagram is shown in in Figure 5. This sequence diagram describes the collaborative access scenarios performed by two users to access the same of an experiment module. The following is its brief explanation. Each user who will use remote labs has to register first. On the registration form, the user must fill in some required data such as user name and password. Once the user is registered, the user has the permission to log into a remote laboratory system. The next step that every user has to do is reservation or booking. To make a reservation, the user must first complete the login process. If the login is successful, then the user can log into the booking menu where the user must specify the date, time and experiment module that will be used to conduct the experiment.

Collaborative access is an activity to access the remote laboratory carried out by a group that consists of several

users. The maximum number of users in a group is determined by the administrator of the remote laboratory. To form a group each user must have a reservation on the same date, time and experiment module. If the reservation is successful, then each user in the group will have permission to access the same experiment module on the same schedule.

When a group is going to conduct an experiment, each user in the group should do the login process. If the login is successful, the user can access the remote laboratory menu. The system will check the reservation status of the user. If the user data are valid, the user will obtain access to the experiment module. Each user in the group is granted the same access: uploading a binary file to the target board's microcontroller, interacting with keypads, and observing the video of the experiment module sent by IP camera on the monitor screen of each user. For coordination and discussion in running the experiment module, all users in the group can use the chat application provided.

Experimental activities can be carried out during the period specified by the administrator of the remote laboratory. To track the time period, countdown of the time is provided on the upper-hand side of the web page. If the allotted time has expired, the entire session provided will be closed and a message stating "time is up" will be displayed.

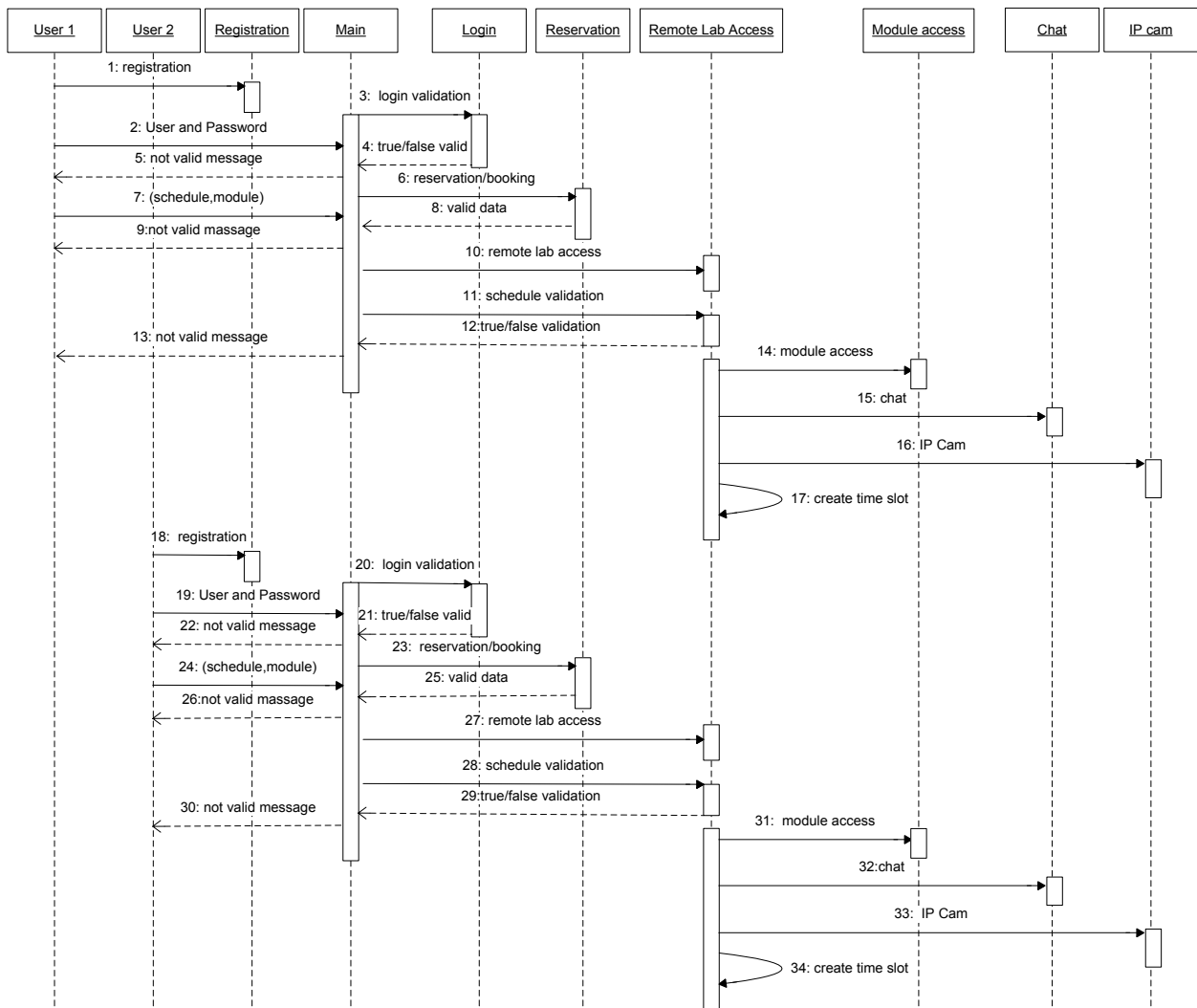


Figure 5. Sequence Diagram Access Remote Lab Collaborative

III. DESIGN OF HARDWARE

A. Embedded Web Server Remote Lab Gateway

Remote lab design implements the web server remote lab gateway based on Raspberry Pi embedded systems. The Raspberry Pi (or RasPi) is an ultra-low-cost credit-card sized Linux computer. It was developed by the Raspberry Pi Foundation, which is a UK registered charity [13]. This contains an ARM1176JZFS, with floating point, running at 700Mhz, and a Videocore 4 GPU. The GPU is capable of BluRay quality playback, using H.264 at 40MBits/s. This computer has 256MB of RAM, 2 USB Port and a 10/100 Ethernet port. Moreover, the computer also has an SD card with a maximum capacity of 8Mb. SD cards are used to store the operating system and application programs. This computer requires a power supply with a voltage of 5 volt and current (minimum) 700mA [12].

The web server remote lab gateway uses Raspbian Wheezy operating system that comes with the Apache web server and Mysql database server. Figure 6 shows the hardware of the Raspberry Pi embedded system.

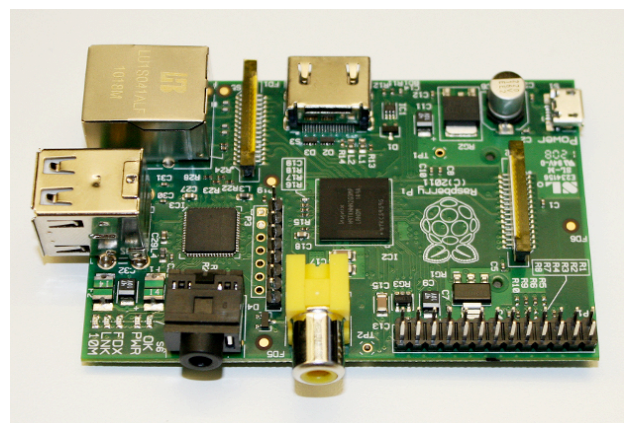


Figure 6. Raspberry Pi

B. Embedded Web Server (EWS) and Experiment Module

Embedded Web Server for control experiment module is built using 8-bit microcontroller MCS-51 family (AT89S52). AT89S52 facility consists of 8 Kbyte internal

code memories, 256 Byte internal RAM, a 32-bit I/O, a serial port and a few interruption hardwares. To meet the needs of the EWS software, the microcontroller system is equipped with an external supporting components that uses external data memory (RAM) for 8Kbyte (6264) and RS232 chip. In addition, the experiment module requires a serial to ethernet converter WIZ110SR, so that the experiment modules can be accessed through the computer network (LAN).

The experiment module consists of a microcontroller AT89S52 and comes with some input output (I/O) equipment for the learning and programming a microcontroller, such as LCD displays, stepper motors, and LED display. Input port facilities are provided in the form of the virtual keypad, generated by the embedded web server. Four virtual keypad buttons are displayed on EWS web pages. The block diagram of the embedded web server and experiment module are shown in Figure 7.

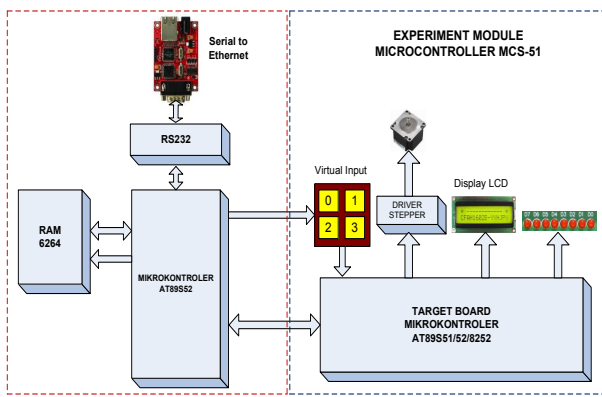


Figure 7. EWS Block Diagram and Experiment Module

IV. DESIGN SOFTWARE OF THE EWS

Embedded web server based on AT89S52 is used to control the experiment module. The flow chart in Figure 8 gives an overview of the workings of the EWS and can be described as follows:

- In active condition, EWS always waits for requests from client. If there is a request from a client, then EWS translates the request type. There are two types of requests: GET and POST instructions.
- If the instruction is POST, the next step is the EWS takes the data from memory, where the data are a binary file that is uploaded by the client. Next EWS running write binary data procedures and download the data to the flash memory of the microcontroller.
- If the instruction is GET, the EWS will check if there are parameters passed. Parameters are associated with the command to enable the port I/O and are used as a virtual button generated by EWS. For example, if the parameter is T0, then the EWS will run the procedure to activate BIT P1.4 of the microcontroller.
- If the GET instruction without parameters, it means the client's request to open the EWS web page and the EWS will respond by sending the HTML code to the client.

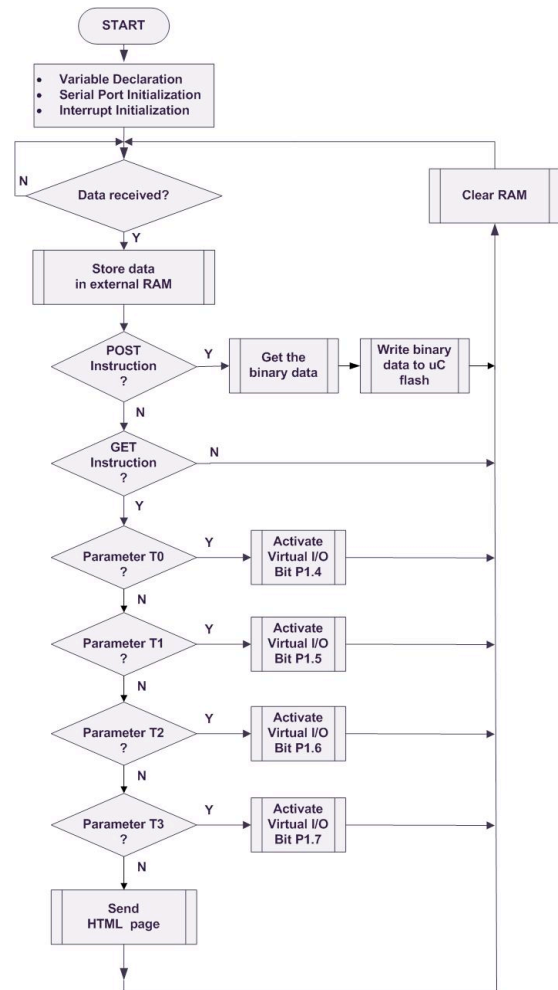


Figure 8. Embedded Web Server Flow Chart

V. RESULT AND DISCUSSION

This research activity has resulted in a prototype of the remote laboratory multiuser based on embedded web server used for microcontroller system experiment. Prototype of a remote lab that has been realized consists of an embedded web server based on Raspberry Pi as remote lab gateway and two experiment modules that is controlled by an embedded web server based on an AT89S52 microcontroller. Each lab module is equipped with an IP camera to observe the experiment module remotely. The IP camera type is a Planet ICA-type HM 100 Series. All the equipment is connected to the network via a switch 10/100 Mbps. Figure 9 shows the remote lab has been realized.

The following will discussed some tests performed on the remote lab system that has been realized. The testing system included EWS experiment module controller performance tests, web server as remote lab gateway performance tests and green computing improvement tests. Remote lab system testing was performed on the local area network (LAN) with maximum bandwidth 100Mbps.

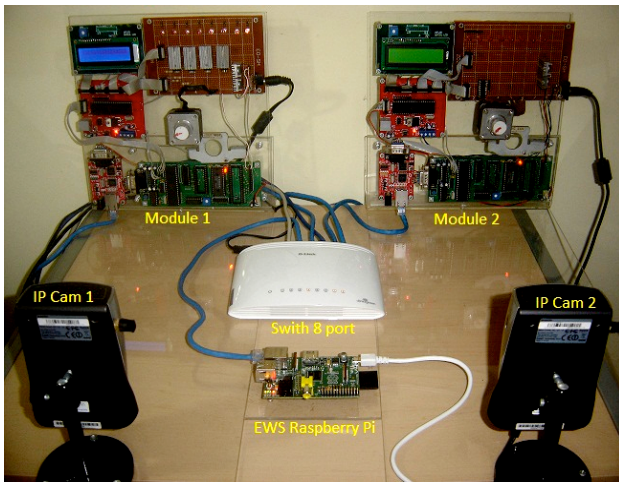


Figure 9. The Remote Lab has been realized

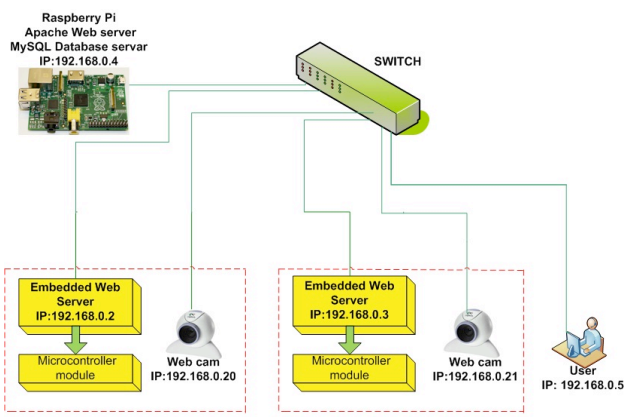


Figure 10. EWS Testing Diagram

The first test was a test of the performance of EWS module experiments. Testing was done by programming the flash memory of the target board 100 times. The results showed a success rate of 98%. Figure 10 shows testing of embedded web server diagram.

The accessibility test of the EWS experiments module on embedded the web server can run well on some types of web browsers such as Mozilla Firefox, Internet Explor-

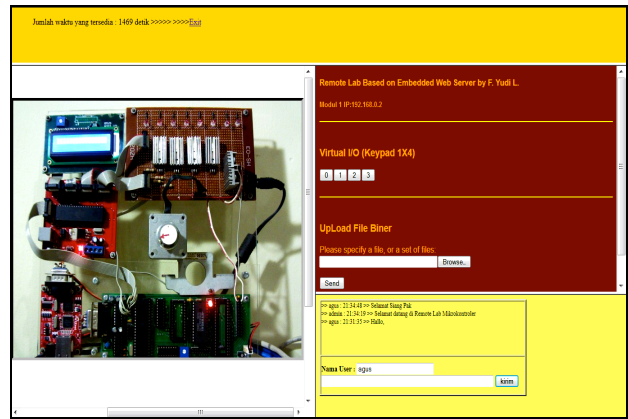


Figure 11. Accessibility Test of the Remote Lab

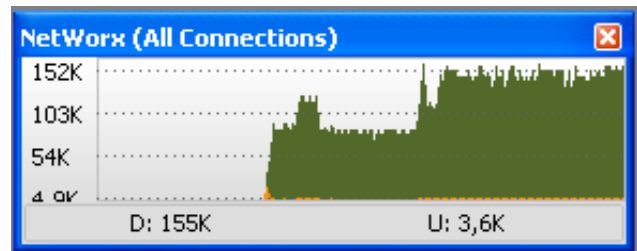


Figure 12. Bandwidth Test

er, Google Chrome and Opera. Figure 11 shows the accessibility test of the remote lab.

Bandwidth testing was needed to access an experimental module, which was performed using Networx applications. Testing was done with a video streaming IP cam resolution of 320X240, and the test results provided a bandwidth of 150kbps. Bandwidth test results are shown in Figure 12.

Performance test of the Raspberry Pi embedded web server was conducted using the software Webservers Stress Tool 7. The test type is RAMP and run test for 5 minutes. User Simulation: ramp test with up to 10 simultaneous users, 10 seconds between clicks (Random). The result of the test average click time for all URLs is 121ms. Figure 13 shows the performance test of web server remote lab gateway.

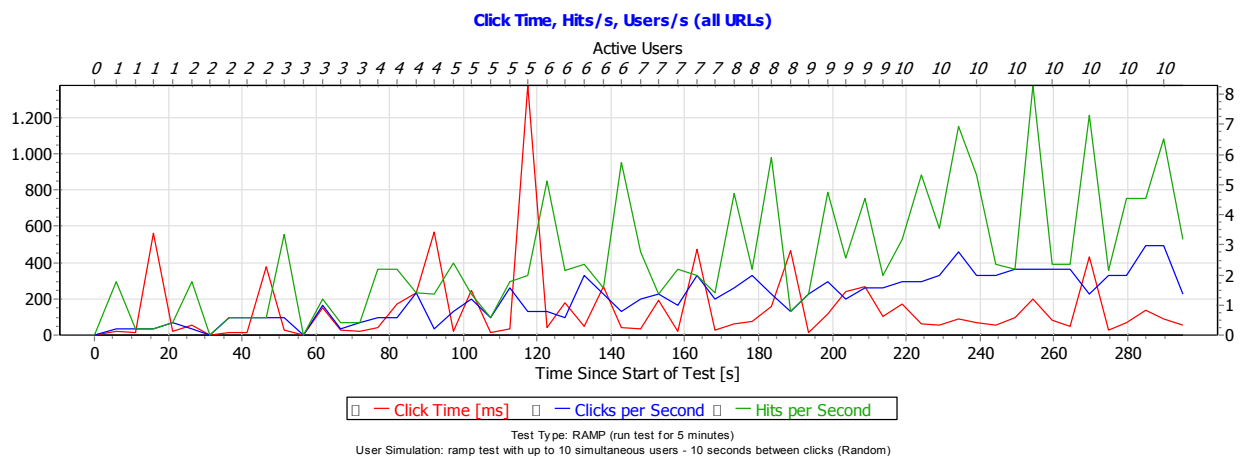


Figure 13. Test Result Click Time

Testing of several parameters related to green computing in web server remote lab gateway includes

1. Raspberry Pi webserver power consumption is 3.5 watts, which can provide power savings of 96% when compared with the use of computer server.
2. The size of Raspberry Pi is 78cm³, when compared to the size of the computer; it saves 99% of space.
3. The price of Raspberry Pi is about \$35, so when compared to the average price of a computer, it saves 88% of overall cost.
4. Remote labs can be used for 24 hours, so the system performance can be improved. It provided 67% performance improvement when compared with conventional lab that is only used for 8 hours a day.
5. Collaborative work in the remote laboratory increases the scalability more than 200% because the experiment module can be accessed by groups that consist of several users simultaneously. The number of group members is set by the administrator.

VI. CONCLUSION

A new architecture remote lab multiuser is presented in this paper. The architecture of the remote lab system is designed using a multiuser and multi device model to support collaborative work to improve system scalability. The remote lab is implemented with an embedded system technology, with the aim to improve system efficiency and support green computing program.

After testing the remote lab a few conclusions can be obtained

1. Remote labs can work well and can be accessed by a variety of web browsers.
2. Scalability and performances testing of the system were done using Webserver Stress Tool 7 with results of the average click time of 121ms for 10 simultaneous users.
3. Tests on experimental modules provided bandwidth requirements on average to access a module for 150kbps.
4. The new architecture of the remote lab has contributed to improved efficiency with a percentage above 60%, and it also supports green computing.

ACKNOWLEDGMENT

The authors would like to thank the University of Indonesia, which provided the funds for the remote laboratory project, and thank the University de Haute Alsace, France, which has given us the opportunity to do a research visit for 3 month. The authors also thank the National Institute of Technology Malang, which funded the publication of the paper.

REFERENCES

- [1] G. Tokdemir and S. Bilgen, "Remote Lab Effectiveness Assessment Model," in *11th International Conference on Optimization of Electrical and Electronic Equipment, 2008. OPTIM 2008, 2008*, pp. 234–239. <http://dx.doi.org/10.1109/OPTIM.2008.4639517>
- [2] E. Otoakhia, T. Jenmanachaiyakun, A. Afaneh, S. Alzebda, M. Mani, O. Sonbul, and A. N. Kalashnikov, "Embedded web server for remote laboratory access for undergraduate students studying electronic engineering," in *2011 IEEE International Symposium on Circuits and Systems (ISCAS), 2011*, pp. 337–340. <http://dx.doi.org/10.1109/ISCAS.2011.5937571>

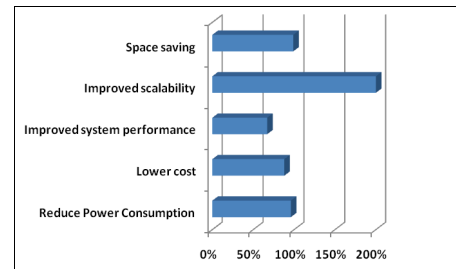


Figure 14. Testing of Green Computing Improvement

- [3] L. Gomes and S. Bogosyan, "Current Trends in Remote Laboratories," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 12, pp. 4744–4756, Dec. 2009. <http://dx.doi.org/10.1109/TIE.2009.2033293>
- [4] J. Garcia-Zubia, D. Lopez-de-Ipiña, and P. Orduña, "Evolving towards better architectures for remote laboratories: a practical case," *International Journal of Online Engineering (iJOE)*, vol. 1, no. 2, Aug. 2005.
- [5] F. Y. Limpraptono, H. Sudibyo, A. A. P. Ratna, and A. S. Arifin, "The design of embedded web server for remote laboratories microcontroller system experiment," in *TENCON 2011 - 2011 IEEE Region 10 Conference, 2011*, pp. 1198–1202.
- [6] S. Murugesan, "Harnessing Green IT: Principles and Practices," *IT Professional*, vol. 10, no. 1, pp. 24–33, 2008. <http://dx.doi.org/10.1109/MITP.2008.10>
- [7] L. Yang, L. Jiang, K. Yue, and H. Pang, "Design and Implementation of the Lab Remote Monitoring System Based on Embedded Web Technology," in *2010 International Forum on Information Technology and Applications (IFITA), 2010*, vol. 2, pp. 172–175.
- [8] I. Gustavsson, K. Nilsson, J. Zackrisson, J. Garcia-Zubia, U. Hernandez-Jayo, A. Nafalski, Z. Nedic, O. Gol, J. Machotka, M. I. Pettersson, T. Lago, and L. Hkansson, "On Objectives of Instructional Laboratories, Individual Assessment, and Use of Collaborative Remote Laboratories," *IEEE Transactions on Learning Technologies*, vol. 2, no. 4, pp. 263–274, Dec. 2009. <http://dx.doi.org/10.1109/TLT.2009.42>
- [9] J. E. Ashby, "The effectiveness of collaborative technologies in remote lab delivery systems," in *Frontiers in Education Conference, 2008. FIE 2008. 38th Annual, 2008*, pp. F4E–7–F4E–12.
- [10] M. J. Callaghan, J. Harkin, E. McColgan, T. M. McGinnity, and L. P. Maguire, "Client-server architecture for collaborative remote experimentation," *J. Netw. Comput. Appl.*, vol. 30, no. 4, pp. 1295–1308, Nov. 2007. <http://dx.doi.org/10.1016/j.jnca.2006.09.006>
- [11] F. Yudi Limpraptono, A. A. Putri Ratna, and H. Sudibyo, "Remote laboratories multiuser based on embedded web server," in *2012 9th International Conference on Remote Engineering and Virtual Instrumentation (REV), 2012*, pp. 1–7.
- [12] Raspberry, "Raspberry Pi Quick Start." www.raspberrypi.org, 2013.
- [13] Raspberry, "Datasheet Raspberry Pi Computer." www.raspberrypi.org, 2013.

AUTHORS

F. Yudi Limpraptono is doctoral student in the Department of Electrical Engineering, University of Indonesia, Kampus Baru UI Depok, West Java, Indonesia and as full time lecturer in Department Electrical Engineering, National Institute of Technology Malang, East Java, Indonesia (e-mail: fyudil@yahoo.com)

A.A. Putri Ratna is lecturer in the Department of Electrical Engineering, University of Indonesia, Kampus Baru UI Depok, West Java, Indonesia (e-mail: ratna@eng.ui.ac.id)

Harry Sudibyo, is a Professor in the Department of Electrical Engineering, University of Indonesia, Kampus Baru UI Depok, West Java, Indonesia (e-mail: hari-sudi@eng.ui.ac.id)

Submitted 07 June 2013. Published as re-submitted by the authors 05 November 2013.