Real-Time Microwave Remote Laboratory Architecture

S. Farah, A. Benachenhou
LEOG Laboratoire d'Electromagnétisme et Optique Guidé
University of Mostaganem, Mostaganem, Algeria
Said.farah@etu.unilim.fr, benachenhou_a@yahoo

G. Neveux, D. Barataud, G. Andrieu, T. Fredon
XLIM CNRS UMR 7252, University of Limoges
123, avenue Albert Thomas, 87060 Limoges-France
denis.barataud@xlim.fr, guillaume.neveux@xlim.f

***Abstract*— An advanced software/hardware flexible and real-time microwave and optical REmote-LABoratory (LABoratoire d'ENseignement VIrtuel: LAB-EN-VI) architecture is presented in this paper. The software part is based on the use of a free license server Node.js written in JavaScript. It offers lightweight Hypertext Markup Language (HTML) and JavaScript clients. The integration of socket.io module enables a real-time operation mode of this Client/Server communication. Associated with hardware architecture, collaborative remote handling of resources (practical works and instruments) is enable in the same way as a chat communication used in Internet. Each remote action performed by one user is instantaneously visible in other users'web interfaces. The hardware part includes a mini-computer "PCduino" that executes node.js server and hosts Mysql databases under LINUX (Lubuntu) operating system. PCduino directly controls specifically developed relays circuit. These circuits offer the required flexibility and reusability for current remote laboratories. Practical Evaluation Board (PEB) corresponding to specific practical works are then designed and easily connected to instruments.**

Keywords—. Flexible, real time, collaborative, Node.js, remote lab.

I. INTRODUCTION

In the last years; many remote labs have been developed to control one experiment at a time [1]. It allows providing specific and unique solutions such as remote control of equipment's designed to be handled in face-to-face labs [2]. The current generation of remote laboratories tends to develop flexible and reusable architectures [3], [4], [5]. The next evolution consists in offering remote students the same capabilities as the one learning in a face-to-face laboratory with a strong interaction with teachers or tutors. Several examples of remote laboratories have been developed in this way [6], [7]. Previous work [8] uses Ajax technology and a Flexible Hardware Interface composed of an Arduino as Ethernet interface and relay Printed Circuit Board (PCB) named (FHI).

The virtual Microwave and Optical laboratory LAB-EN-VI has been developed at Limoges University. It is used by students in the undergraduate courses of the master ARTICC (Architecture des Réseaux et Technologies Induites des Circuits de Communication: Network Architectures and Associated Technologies of Communication Circuits). It is based on an accurate undergraduate teaching laboratory concerning optic and microwave fields and it addresses several learning issues as simulation and experiment. For this LAB-EN-VI, a specific work has been done to introduce multi-user real-time and collaborative practical works in a remote laboratory. This paper presents the software and hardware configurations and accurate description of these advanced real-time practical works, especially those developed for characterization of radio frequency (RF) devices. Firstly, the node.js server used as main software technology for the hereafter developed remote lab is described in the second paragraph of this paper. The software architecture of the web client interface is presented in the following paragraph. RF device characterization web client interface are also shown as a matter of example. Then, a description of the remote lab hardware architecture is presented. Next paragraph presents the operating mode of the developed real-time remote laboratory with examples of collaborative remote practical works.

II. DESCRIPTION OF THE LAB-EN-VI NETWORK ARCHITECTURE

A. LAB-EN-VI structure

Fig. 1 describes the LAB-EN-VI. network architecture.

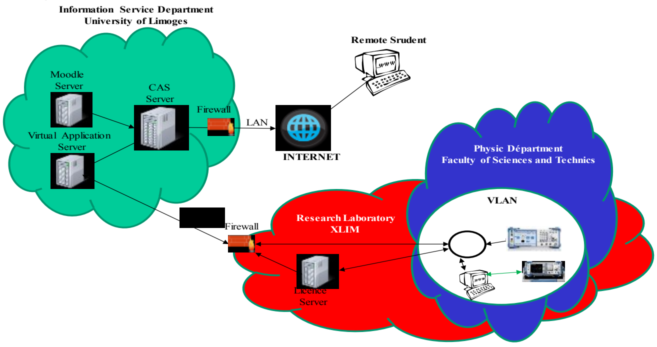

Fig. 1. Network Architecture of the LAB-EN-VI

Remote students have a unique login and password provided by the French CAS server in order to access the remote laboratory (software and hardware).

To complete understanding of their theoretical course material, the LAB-EN-VI provides them different lab experience by controlling remote microwave and optical practical experiments. In such context, it is highly necessary to provide them the capability to work on different experiments based on an identical architecture. This is why a common interface "Ethernet Flexible Hardware Interface" (EFHI) has been developed capable of driving several “Practical Evaluation Boards” (PEB), as shown in the global description of a Remote practical work in Fig. 2.

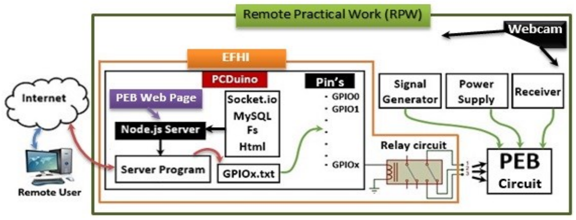

Fig. 2. Remote Practical Work architecture (Hardware/Software) of the LAB-EN-VI

B. *Detailed description of the Hardware architecture based on EFHI/PEB*

Hardware architecture of the remote laboratory is mainly composed of an advanced EFHI (Ethernet Flexible Hardware Interface) and a PEB (Practical Evaluation Board) provided for each practical work.

Fig. 3 is a photography of an example of a practical work with the EFHI, the PEB, the embedded function generator and receiver (©Keysight DSOX3014a [ref web]), the Ethernet webcam and the Scalable Vector Graphics/Java Script (SVG/JS) dynamic web interface.

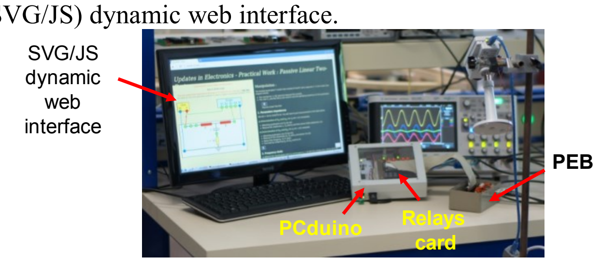

Fig. 3. Photography of the practical work

The advanced/upgraded version of the board EFHI consists of a PCDuino platform that controls a 13 relays PCB card through the 23 Digital Outputs called GPIO. Fig. 4 shows a photography of the advanced EFHI used in a real remote practical work.

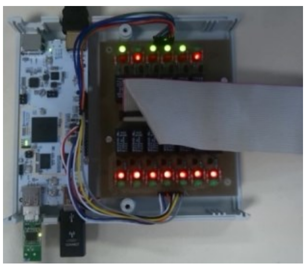

Fig. 4. Photography of working EFHI

PCduino platform is shown in the photography of figure 5.

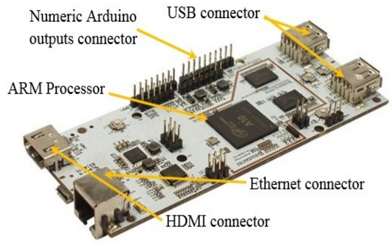

Fig. 5. First release of PCDuino platform

It is a mini PC platform able to work with different Operating System (OS) such as Ubuntu and Android ICS. The presented PCduino works with Linux OS integrating [12]. The main advantage of this advanced platform lies in the combination of computer hardware structure, computer functionalities and the Arduino’s functionalities used in the previous version of remote laboratory [8].

- *EFHI Ethernet Flexible Hardware Interface:* It is a reusable hardware interface in all remote laboratories provided in EOLES bachelor. The relays PCB card designed specifically for the needs of remote practical works. The status of each relay is shown in real-time with two red and green LED. The flexibility provided with EFHI facilitates the design of PEB "Practical Evaluation Board" for each remote practical work. The connection between each EFHI and PEB is a classic 39 pin computer IDE (Integrated Device Electronics) cable. Fig. 6 shows the EFHI structure and the relays circuit base used in the 13 relays card.

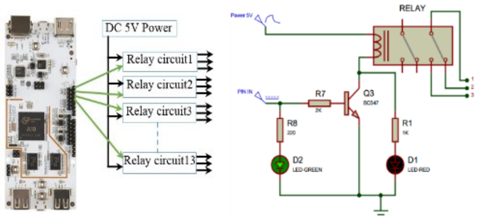

Fig. 6. EFHI structure and relay circuit

- *PEB Particle Evaluation Board:* PEB's are designed for each practical work. Each PEB represents the remote controlled interface of a practical work, it provide reusability and flexibility to the remote laboratory structure. The remote control of any PEB can be performed with the same EFHI. Fig. 7 shows the PEB designed to characterize RF active or passive devices. Instruments (RF generator, Signal analyzer or Network Vector Analyzer) are connected to the SMA RF Switch input; the characterized RF devices are connected to the SMA RF Switch outputs. For each practical work web interface, Node.js server program is unchanged. Only the code of the dynamic web interface needs to be modified.

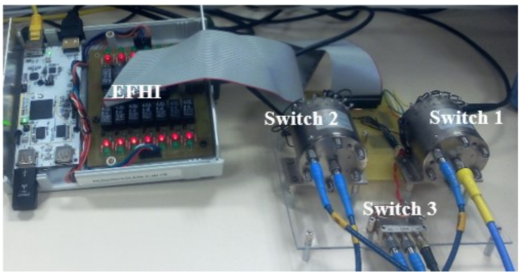

Fig. 7. RF active and passive PEB connected to the EFHI

III. NODE.JS SERVER DESCRIPTION

Node.js is a free platform built in JavaScript language, easy to install. To communicate with server, protocol like WebSocket, Adobe Flash sockets, JSONP polling, or AJAX long polling [10] can be used. Node.js use Socket.io module to communicate with their clients in direct conversation or broadcasting to multiple clients. Node.js is lightweight and efficient for data-intensive real-time applications that run across distributed devices [9].

A. *File System "fs". module*

File System module "*fs*" is an accessing File System module of Node.js software. This module simplifies the standard file operations (create, delete, write, read, open, close, …) [9]. In the context of the presented LAB-EN-VI, each practical work is associated with a specific html file.

B. *"Socket.io" module*

"*Socket.io*" is a mandatory module installed as node.js software in the "PCduino" mini-computer allowing the server-client input-output requests. Thanks to this fundamental added module, programs written in JavaScript language enable real-time bidirectional communications between web client and web server. Reliability, flexibility, rapidity over a large number of operating system platforms and web browsers are the main advantages of this technology [10] to obtain real-time applications. .

C. *Database management module MySQL*

MySQL node.js module provides the ability to communicate with mysql databases. The treatment of these popular databases is possible with node.js which facilitates the implementation of collaborative management of remote laboratory resources, access to web interfaces and registration of the various experiments performed by the remote user.

D. *Software architecture based on node.js and EFHI connections*

The hardware and software EFHI structures are illustrated in Fig. 8. The 13 PCduino pins are programmed in outputs digital mode. It controls the 13 relay states that are transmitted in real time to the PEB to applying configuration requested by the remote student.

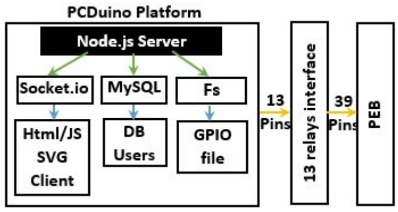

Fig. 8 . EFHI hardware and software structure

The remote student control electronic circuit through the dynamic web page written in SVG (Scalable Vector Graphics) language. Commands are sent in real-time to the server. According to these commands, the PCDuino pins are set to high or low voltage (0V or 5V). These statements are sent directly to enable or disable each one of the 13 relays. The associated PEB receives and applies the new configuration requested by the remote student using the software architecture and the command routing (from the web client interface to the concerned PEB) shown in Fig. 2

IV. WEB CLIENT INTERFACES STRUCTURE

A. *Web interface structure*

HTML client's remote laboratory communicates with Node.js server by JavaScript socket.io functions. The web client can be loaded on each request from a remote user by specifying the IP address and port number of the sever program hosted in PCDuino. JavaScript functions combined with SVG (Scalable Vector Graphics) scripts provide remote control of concerned by the remote student, so connections between instruments and electronic circuits as shown in Fig. 9 for RF devices characterization.

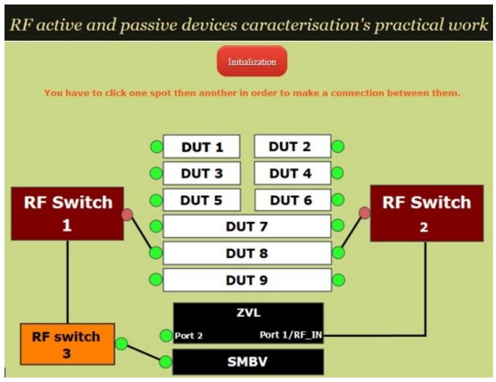

Fig. 9. RF device caracterisation SVG web interface

Even with slow internet connections, the association of HTML, JS, SVG enables a fast real-time control of the experiment.Remote student manipulates SVG circuit on the web interface by clicking on the green circle of the desired connection. EFHI actives relays corresponding to this connection. The practical work presented in Fig.9 allows the characterization of RF 2-port (Short, open, 50Ω circuits for calibration for instance) or 4-port Device Under Test (DUT) (attenuators, filters, biased transistors). Switches 3 and 4 are used to connect instruments to the reference DUT planes. Small signal as large CW or modulated signals can then drive the different DUT to measures S-parameters, power performances

(AM/AM, AM/PM) and linearity criteria (Intermodulation products, ACPR, EVM). The flexibility of this architecture allows the capability to use other instruments (RF oscilloscopes …). Remote control of the instruments used in this practical work was described in a previous work [8].

B. Remote Experiment Process

After authentication, the remote user connects as a client to the server running on PCDuino. As shown in the example of Fig. 10, students can perform the practical work by organising their computer screen. It load the PEB web interface (client), connect to instruments front panel, connect to web cam to have in real time the remote laboratory real vision.

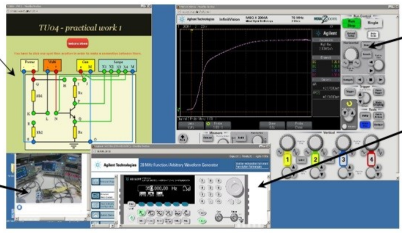

Fig. 10. Student’s screenshot of a real-time remote practical work to measure I/V charctareistics

Fig. 11 shows the real vision of the RF device caracterisation remote laboratory. A SMBV RF Generator is connected directly.

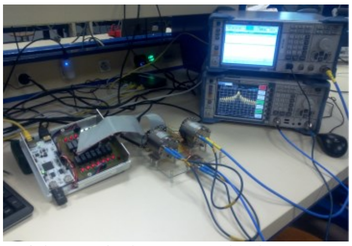

Fig. 11. Real vision of the RF device caracterisation remote laboratory

Front panels of the two instruments, RF generator and Signal Analyzer/Virtual Network Analyzer can be obtained in the screen of the remote student. A client/server software is used to facilitate the remote control of these two instruments. Fig.12. shows the front panel of the two instruments as seen in the remote student PC. In this example, a RF signal with frequency of 2GHz and a level of 10 dBm modulated by a Pulse signal with width of 2us and a period of 10 us is generated. It is caracterised by a Signal Analyzer.

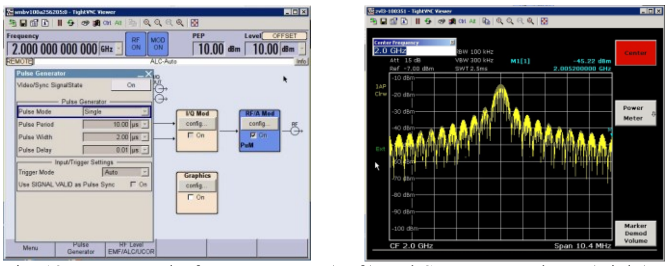

Fig. 12. Front panel of RF generator (Left) and Spectrum analyser (Right) as shown by a remote student

Students can also work collaboratively on the same Practical work using virtual classroom as Adobe Connect or BigBlueButton. When a group of users connects to a remote practical work, each user's action is transmitted and applied to the web user's interface simultaneously and applied to the real remote practical work.

V. CONCLUSION

This paper presents an advanced software/hardware real-time and flexible microwave and optical REmote-LABoratory. It describes both software architecture based on the use of Node.js written in JavaScript and hardware architecture of the different microwave practical works.

Each practical work is named Practical Evaluation Board. PEB are easily connected to instruments and are remotely controlled by a flexible motherboard named EFHI Ethernet Flexible Hardware Interface.

EFHI consists of minicomputer PCDuino (connected to the network) and a relays card controlled by the PCDuino digital outputs. This solution replaces heavy server by low coast minicomputers with the same functionalities.

Many microwave practical works have been realized to help the students to understand how the microwave equipment work. They were also developed to remotely make them measuring the characteristics of filters, attenuators and FET biased transistors in terms of DC, S-parameters, power and linearity performances.

REFERENCES

[1] A. K.M. Azad, and P. Kaushik, “RoombaCreate® for Remote Laboratories,” iJOE, Volume 10, Issue 4, 2014.

[2] A. Kalantzopoulos, and E. Zigouris, “Online Laboratory Sessions in System Design with DSPs using the R-DSP Lab,” iJOE – Volume 10, Issue 4, 2014.

[3] O. Salaheddin, “A Web-Based Remote Lab Platform with Reusability for Electronic Experiments in Engineering Education,” iJOE – Volume 10, Issue 4, 2014.

[4] N. Sousa, G.R. Alves, and M.G. Gericota, “An Integrated Reusable Remote Laboratory to Complement Electronics Teaching,” Learning Technologies, IEEE Transactions on ducation, Volume: 3 , Issue: 3, Page(s): 265 – 271, 2010.

[5] U. Hernandez-Jayo, J. Garcia-Zubia, “Low cost remote and reconfigurable analog electronics laboratory,” Remote Engineering and Virtual Instrumentation (REV), 2014 11th International Conference, date of Conference Porto 26-28 Feb, Pages 457 – 460, 2014.

[6] A. Kalantzopoulos and E. Zigouris “Online Laboratory Sessions in System Design,” iJOE – Volume 10, Issue 4, 2014.

[7] S. Odeh, M. Anabtawi, A.Ayesh, “Applying reusabililty in remote labs ,” Electrotechnical Conference (MELECON), 2012 16th IEEE Mediterranean, Learning Technologies, date of Conference 25-28 March 2012, Pages 265 – 271, 2010.

[8] S.Farah, A. Benachenhou, G.Neveux, D.Barataud, “Design of a flexible architecture for characterization of active and passive microwave devices (patch antenna, filters, couplers and amplifier stage) in a remote laboratory,” Global Engineering Education Conference (EDUCON), 2013 IEEE, Pages 847 – 853, 2013.

[9] Online “Node.js,” http://nodejs.org

[10] Online “socket.io 1.0 is here, featuring the fastest and most reliable real-time engine,” http://socket.io/

[11] Online “Accessing the File System in Node.js,” http://www.sitepoint.com/accessing-the-file-system-in-node-js/

[12] Online “Mini PC + Arduino (TM), pcDuino,” http://www.pcduino.com