JPE 10-6-5

Real-Time Power Electronics Remote Wiring and Measurement Laboratory (PermLAB) Using 3-D Matrix Switching Algorithms

Johnson A. Asumadu[†], Ralph Tanner[*], and Hakeem Ogunleye[**]

[†*] Dept. of Electrical and Computer Engineering, Western Michigan University, Michigan, United States
[**] School of Electrical and Computer Engineering, Purdue University, Indiana, United States

Abstract

This paper presents a new architecture, called ”*Power Electronics Remote Wiring and Measurement Laboratory (PermLAB)*”, that translates a common gateway interface (CGI) string from a remote web user to a web server connected to a 3-dimension switching matrix board, can be used to switch on and off, and to control a cluster of instruments and components. *PermLAB* addresses real-time connection, switching, and data acquisition over the Internet instead of using simulated data. A software procedure uses a signature system to identify each instrument and component in a complex system. The Web-server application is developed in HTML, JavaScript and Java, and in C language for the CGI interface, which resides in a controller portion of LabVIEW. The LabVIEW software fully integrates the Web sever, LabVIEW data acquisition boards and controllers, and the 3-dimensional switching matrix board. The paper will analyze a half-wave rectifier (AC – DC converter) circuit connected over the Internet using the *PermLAB*. *PermLAB* allows students to obtain real data by real-time wiring of real circuits in the laboratory using a “virtual breadboard” on the Web. The software for the Web-based 3-dimensional system is flexible, portable, can be integrated into many laboratory applications or expanded, and easily accessible worldwide.

Key Words: Remote wiring, 3-D Switching matrix, Power Electronics, Laboratory, Measurement technologies.

I. INTRODUCTION

The Web and Internet technologies have been used for e-commerce, banking, video and data on-demand [1]-[3], to access, to retrieve, to control live, to achieve data acquisition system in distributed virtual instruments, etc. The Internet Protocol has also been used in meter applications [4]. It has been used to measure, process, and retrieve data with limited interaction, collecting off-line data, accessing real-time data, and for achieving data.

These Web-based architectures used expensive dedicated networks, several bus systems, and third-party protocols [5]-[7]. In [8] the software used is platform independent, and no third party protocol was required on both the software on the user side and on the instrumentation side; Java, JavaScript, and only hypertext transfer protocol (HTTP) protocol were required. Recently, real instruments and measurement data can be assessed remotely by using Ethernet [9]-[11]. Various data protocol systems have been used to connect sensors, control devices, and real instruments [12]-[15].

Even though [16]-[17] presented an e-learning system where the authors demonstrated the versatility of a didactic system, most of the instruments and measurement devices presented by the authors are pre-wired in the laboratory. One of the major drawbacks in most of the references cited is the dependency on specialized software and virtual instrumentation [18]-[19]. Some Web applications such as “computer based training”

(CBT) and “computer assisted learning” have been developed to take over part of the teaching-learning process [20]. Such Web-based approaches still depend on simulation software packages. The real challenge is to provide innovative new ways of using the Web for the training of engineering students. There have been calls for new remedies for revitalizing engineering in order to make engineering more attractive to students. For example, at the University of Minnesota, NSF has funded projects [21]-[22] aimed at developing new approaches of teaching power electronics.

A new Web-based hands-on real-time “*power electronics remote wiring and measurement laboratory (PermLAB)*” has been developed. Students are physically able to wire up electrical and electronics circuits at the host laboratory site using the Internet access and by means of a conventional “virtual breadboard” on the web. The data acquisition interface allows students to make real-time data measurements and the data collected are made available on web. The web-based architecture and protocol developed translates a CGI-query

Manuscript received May 3, 2010; revised Aug. 6, 2010

[†] Corresponding Author: johnson.asumadu@wmich.edu
Tel: +1 260 276-3147, Western Michigan University
[*] Dept. of Electrical and Computer Engineering, Western Michigan University, United States
[**] School of Electrical and Computer Engineering, Purdue University, United States

string from a user to a web server connected to a 3-dimension switching matrix board; hardware and software that implement remote wiring (includes components) and measurement architecture. The paper also presents a reconfigurable Internet-based instrumentation generation and data acquisition system through the use of transmission control protocol and internet protocol (TCP/IP) and HTTP. All instruments and components use GPIB interface to translate instrument commands and controls, and data transfer from Internet protocol to local bus control connected to real instruments and components. The components and instruments are physically defined on the 3-dimensional matrix switching board of relays. The developed software is independent of third part protocol (HTTP-based), and does not require expensive dedicated networks and different bus systems. The architecture and protocol are robust, uses encrypted data for transactions, and requires user identification for access.

II. SERVER SIDE HARDWARE ARCHITECTURE

A. Generalized n-Terminal Matrix Board Initialization

The matrix board is a standard matrix board with electronic relay latches. A pictorial representation of an example of an *n* x *m* matrix used is shown in Fig. 1. The matrix board is defined by *n* rows in the *x-y* plane and *m* columns on the *z*-axis. The exact physical location of each terminal of each component or instrument on the matrix board must be known. An applet initialization file is set up to contain the row numbers of all terminals of instruments/components on the matrix board. The following configuration statement is used to specify the exact location of a terminal of an instrument/component on the matrix board (uses electronic relays):

Instrument/Component Name |Description |rown | row(n-1) | . . . | row1 | row0

Optionally, a "!" before the "*Instrument/Component Name*" in the initialization file disables the *Instrument/Component* statement (i.e. a "!" represents a comment). The "*Instrument/Component Name*" is used to specify the name of an instrument or component. The "*Description*" is used to describe the instrument or component. The "*rown*" specifies the *row* number where the terminal of an instrument or component is connected to. For example, in Fig.1 the *k*th instrument/component such as a MOSFET has 3 terminals: terminal 1 (drain) is connected to row 2, terminal 2 (gate) connected to row 4, and terminal 3 (source) connected to row 5.

From the configuration, it is clear that the matrix board has a maximum of 2*n* terminals for all components/instruments. An instrument or component may have more than one terminal. Also, several terminals from different instruments may be connected to the same row. In Section 4.0 we will show how the terminals are set up and included in a binary file.

The fixed coordinates in the *x-y* plane will be called ***rows*** and the z-axis will be called ***cols*** (or sometimes ***nodes***). Note that the ***nodes*** are represented by the electronic relays. If an electronic relay is turned ON, then that ***node*** is connected. Each terminal of an instrument/component is connected to a row but all the *m* ***cols*** (***nodes***) in that row are available to that terminal of the instrument/component for connection. For example, the Fig. 1 shows that the terminal 1 of *k*th instrument/component is connected to row 2 and all the *m* columns (***cols*** or ***nodes***) of row 2 are available for connection.

TABLE I: Generalized Initialization File

Comment Name	Description	Rows							
		0	1	2	3	4	5	. .	n-1
!	Ground								
G	00	0	-1	-1	-1	-1	-1		-1
!	MOSFET								
MOSFET1	Mosfet	-1	-1	2	3	4	-1		-1
. .									
!	kth Load								
L	LDK	-1	1	-1					-1

The terminals of an instrument/component will are defined in the following format:

rown row(n-1)... row1 row0 colm col(m-1)... col1 col0

Normally *n* is usually greater than *m* so that more components/instruments can be accommodated. The binary representation is formed at the web site of the remote user interface and included in a CGI-string. The row and column numbers are translated into binary representation. For example, row 2 will be translated into binary value as "10".

B. 3-Terminal Switching Matrix Board Initialization File

The format for a configuration statement in an initialization file has the following format:

Instrument/Component Name | Description | Terminal_One |Terminal_Two | Terminal_Three |...| Terminal_TwoRaised_to_n

The "*Instrument/Component Name*" is used to specify instrument or component name and "*Description*" is used to describe an instrument or component. The "*Terminal_One*", "*Terminal_Two*", "*Terminal_Three*", etc, define the rows in the *x-y* plane and *z*-axis or column of an instrument or component on the matrix board.

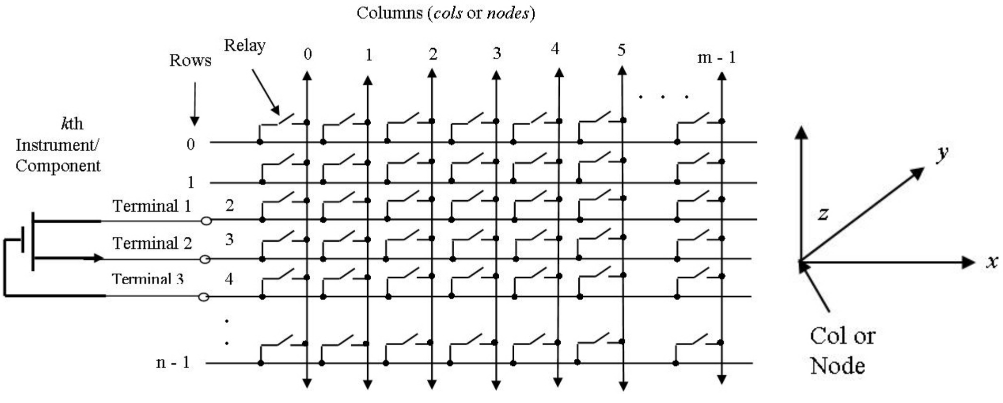
The generalized statements in a given initialization file is shown in Table 1. For example, the component MOSFET shown in Fig. 1 has three terminals and connected to rows 2, 3, and 4. This component will be initialized as follows:

! MOSFET transistor connected to terminals 2, 3, and 4 of
! switching matrix board
MOSFET1 Mosfet $-1234--1-1...-1$ (up to 2^n rows)

Statement 1: comment

Statement 2: the "*Instrument/Component Name*" is MOSFET1, "*Description*" as Mosfet, connected to rows 2, 3, and 4. The "-1" in the statement implies that these rows are not connected because the Mosfet has three terminals.

The rows of all instruments/components must be initialized and stored in a text file. The initialization file defines the exact



(a) (b)

Fig. 1: An n x m Switching Relay Matrix

physical location of the terminals of an instrument/component on the rows of the switching matrix board; how the terminals are connected to the rows only. The initialization text file is modified when an instrument/component is added or removed from the switching matrix board. The positions of the instruments/components are fixed on the matrix board at the host Web-server/microcontroller.

C. Remote Instruments on Switching Matrix Board

The proposed distributed instrument architecture on the switch matrix board is shown in Fig. 2 (see Appendix II) (similar to [23] and see Appendix I). The architecture consists of a switching matrix board, Web-based microcontroller server (front-end is a Web server and back-end is a microcontroller), host PC-based server, and multi-instruments/components. The instruments/components could be different or similar, active, analog, digital, or operating simultaneously. All the instruments and some components use GPIB interface to translate instruments commands and controls, and data transfer from Internet-protocol to local bus protocol (example from Ethernet to GPIB). The data and file transfer between the host PC and the remote PC is by means of common Internet data interchange techniques; file transfer protocol (ftp), hypertext transfer protocol (http), and any local Windows network protocols.

III. MEASUREMENT ALGORITHM AND SOFTWARE

A. Algorithm Implementation – Translation from Virtual Representation to Physical Representation

The Fig. 3 shows a virtual breadboard at the remote web user's PC (or laptop) with some instruments and components. The images of the instruments/components on switching matrix board at the web-server host are represented on the virtual breadboard using Java Applets. The images are produced using high graphic software. The image of the component BJT shown in Fig. 3 can be moved around on the virtual distribution breadboard by the remote web user interface on the PC. As the web user moves the instrument/component on the virtual bread board, the protocol (software embedded in Java file) keeps track of the "live" terminals on the virtual breadboard. Depending on the position of the terminals of BJT on the breadboard, the terminals will snap into the nearest columns (***cols*** or ***nodes***) on the breadboard. The column or node number encountered by a terminal is assigned to the terminal by JavaScript software. Terminals of several instruments/components may be connected to the same column (***node***). For example, the ***row*** 2 (terminal) of MOSFET1 image snaps to ***col*** 0, ***row*** 3 (terminal) of BJT image snaps to ***col*** 1, and ***row*** 4 (terminal) of BJT image snaps to ***col*** 2. Furthermore, a wire tool is provided to combine multiple columns into a single column. A summary of the final translation is shown in Table 2. And the terminals of the BJT are addressed as follows:

*row*2 *row*3 *row*4 *row*5 *col*0 *col*1 *col*2 *col*3

B. Software Implementation

The software consists of three main components. First, a web-based graphical user interface (GUI) that simulates and feels like a wireless distribution breadboard and instruments/components. It allows the remote web user to wire remotely instruments and components, translates commands and controls, and transfer data from instruments/components into web pages. Second, the main web-server components are relay switching matrix board, web-based microcontroller, and control systems that provide physical connections between instruments and connections. Third, a feedback system that provides the remote web user with circuit analysis tools to enable the remote web user to evaluate the connected circuit or network on the GUI interface. The software was implemented on hardware platform made of a PC-based web server, a dedicated web-based microcontroller server, National Instrument (NI) LabVIEW software, and NI instruments (switching relay matrix board and data acquisition cards), and Java applet. There are four main components of the software – (i) the GUI interface, (ii) CGI, and C and assembly languages for the dedicated Web-based microcontroller, (iii) software for analog-to-digital converter (ADC), GBIP, input/output (I/O) cards to translate commands and controls, and data transfer from Internet protocol to local bus control, and (iv) protection

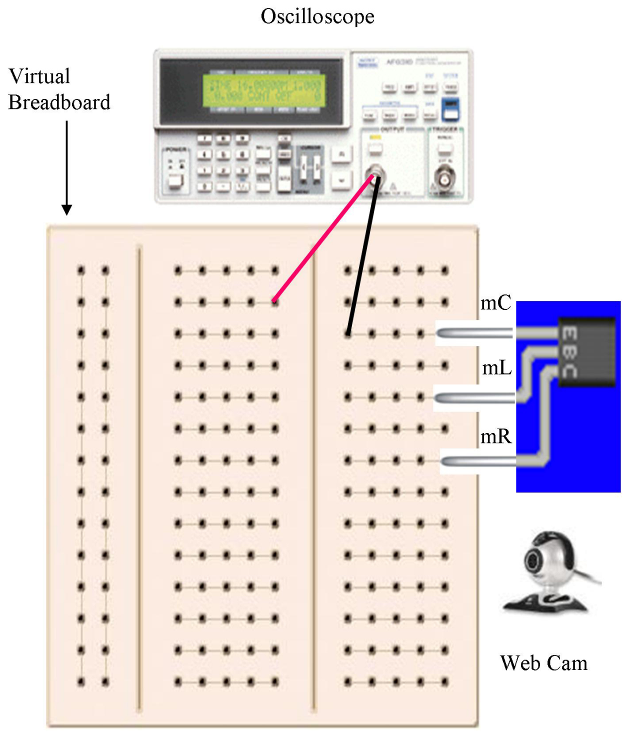


Fig. 3: Virtual Breadboard with Some Instruments And Components

TABLE II: Translation Table

Component	Name	Row	Column
INSTR2	Left_terminal	.	.
	Right_terminal	.	.
BJT	mL	2	0
	mC	3	1
	mCR	4	2
.	.	.	.
.	.	.	.

of network (servers and network devices) and preventing unauthorized access.

1) Java Applet and GUI: There are three main Java classes. First, a main Java source draws the applet and generates the column or node numbers of the terminals of the instruments/components on the virtual distribution board. The second Java “instrument and component“ class allows the remote user to control the instruments and components on the virtual distribution board. Other Java classes, derived from the “instrumentation and component” class, contain functions to draw, rotate, resize, and move terminals of the instruments and components. In addition to the NI LabVIEW software, the NI hardware also came with Java applet. Thirdly, the Java applet loads a dedicated initialization file that is used to define the row number locations of all terminals of the instruments and components, and drivers physically on the relay matrix.

2) Web-based Microcontroller and Applet/CGI Interaction: The front-end of the Web-based microcontroller is a web server and the back-end is a microcontroller. The web-based microcontroller server is used for CGI execution. The source code for the controller portion was written in C and assembly languages. The Java applet at the remote user sets up a CGI string depending on the connections on the html page by the remote web user. The CGI string is passed onto the web-based microcontroller by the remote user. The web-based microcontroller parses and sends the commands, through the controller portion, to the relay matrix board to control switching and for information for data (voltage, current, connection profile, etc.) requests to and from other instruments for display on new html page at the remote web user.

3) Control and Data Acquisition: All instruments and components use a GPIB interface card to translate instrument commands and controls, and data transfer from Internet to local bus control. Most electrical and electronics instruments and components use GPIB protocols. In addition to the GPIB card, input-output (I/O) cards (serial, parallel, and USB), and analog-to-digital converters (ADC) were also provided.

4) Protection of Network and Preventing Unauthorized Access: The network (servers, network devices, and instruments and components) was protected against intrusion and constant service entry device hackings. The system was also protected against internal abuse of servers, network devices, virus, and short circuit of instruments and components. Every interaction and activities on the network was recorded, extensively documented, and date stamped. Reports were generated from time to time and on request to identify any breach of protocol and anomalies of the network. Alarms were also installed to help alert any immediate failures in the network especially network devices, and instruments and components.

IV. SYSTEM IN ACTION: *PermLAB* EXPERIMENT IMPLEMENTATION

A. Hardware

The Fig. 4 (see Appendix III) shows an example of a portion of the switching matrix board relay configuration with an ac source, a diode, and a 1-kΩ resistance. The Fig. 5 shows the physical connection of a power electronics half-wave rectifier circuit in the laboratory; the ac source, the diode, and the 1-kΩ resistor. The web-sever at the host houses the 3-dimensional switching matrix board and web-based microcontroller, instruments, and components. The exact location of the terminal of each instrument or component on the matrix board must be known and a Java applet initialization file was set up to contain the row numbers of all the terminals. The electronic components displayed on the virtual distribution breadboard are driven by the initialization file. The file details what components are physically available and the exact location of their terminals on the matrix board, defined by rows. Consider a 9 x 4 matrix portion of the switching matrix board; 9 rows and 8 columns. In the experiment only three columns are used. The terminals of the ac source, the diode, and the 1-kΩ resistor are physically connected to the following rows on the switching matrix board:

ac source: ***row*** “4” and ***row*** “5”
Diode: ***row*** “8” and ***row*** “9”
1-kΩ resistor: ***row*** “6” and ***row*** “7”

The Table 3 (shown in Appendix I) shows the contents of the initialization file defining the physical terminal row number positions of the instruments/components on the matrix board.

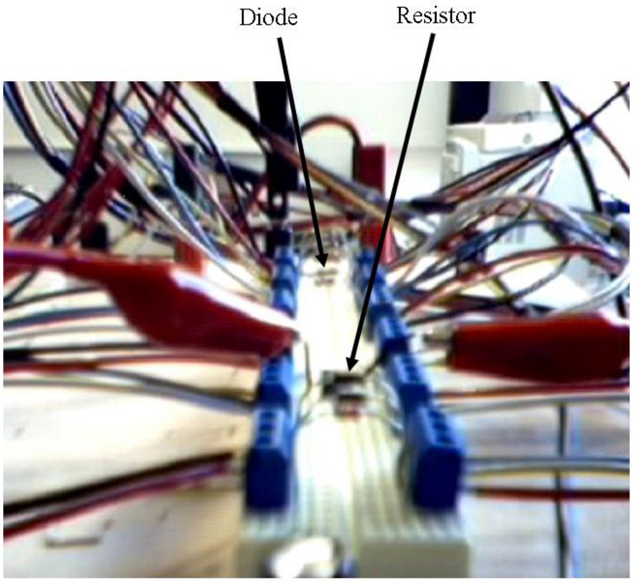


Fig. 5: Physical Connection of Instruments and Components

B. *Software*

1) Translation from Virtual Representation to Physical Representation: As the remote web user builds the circuit on the web page, the Java applet software keeps track of the "live" components on the virtual breadboard. The translation algorithm assigns column numbers to the terminals of the components. For flexibility purposes, the algorithm automatically assigns ***col "0"*** to the column when it encounters the first component's terminal. The following columns are assigned to the terminals by the algorithm:

ac source: ***col*** "0" and ***col*** "2"
diode: ***col*** "0" and ***col*** "1"
1-kΩ resistor: ***col*** "1" and ***col*** "2"

Therefore, the terminal representation of the components is represented by the following format (explained in Section 2.0):

ac source: ***row*** "4" and ***row*** "5" ***col*** "0" and ***col*** "2"
diode: ***row*** "8" and ***row*** "9" ***col*** "0" and ***col*** "1"
1-kΩ resistor: ***row*** "6" and ***row*** "7" ***col*** "1" and ***col*** "2"

The row numbers are assigned and represent the physical positions of the terminals of instruments and components on the switching matrix board at the host. The column numbers are assigned by Java software as a component is moved to a position on the virtual breadboard of the remote web user. Table 4 (Appendix II) shows the virtual to physical translation from the breadboard to the switching matrix board. The component terminals are also indentified by names (optional) as shown in Table 4 (see Appendix II). For example, consider the *ac* source: The left terminal is called *acL*, physically connected to ***row*** 4 (assigned at the host) and assigned to ***col*** 0 (assigned by Java software at web user). The left terminal is called *acR* connected to ***row*** 5 and assigned to ***col*** 2.

2) CGI Binary String : The Table 4 (shown in Appendix II) shows the translation table. In the translation process the Java applet transforms the information it has built up from the completed circuit into a Binary Representation. The binary string has values for CLK (to re-start clock of hardware) and !RESET to control the reset function of the hardware. A reset is always applied to the hardware to clear the previous state of the physical hardware before a new connection is made. The translation uses the information from the configuration file to select the address of a component and then it appends to that information, which node the component is connected (if that component is used). For example, the binary string "0100100000" (integer value is 288) turns on the left terminal *acL* of ac source (***row* 4** and ***col* 0**) and the binary 0100101010 (298) turns on the right terminal *acR* of ac source (***row* 5** and ***col* 2**). The Table 4 shows the final translation and binary representation. Once the circuit is completed, a compaction takes place and the translation algorithm builds up a CGI string with the following format:

http://141.218.115.11/ref.cgi?C0=7&&C1=288&&C2=298
&&C3 =320&&C4=329&&C5=305&&C6=314 (1)

Where

http://141.218.115.11/: This is the IP address of the Web-based microcontroller server.

C0: The value of C0 tells the CGI program how many parameters will follow. In this example parameter 1 through 6 were used, so C0 = 7 (C0 included in the count).

C1 – C(n): These variables are sent as an integer between 256 and 511. This integer contains the row and column values to switch on an individual relay; the virtual representation of the physical relay ***columns*** or ***nodes*** (***col***) on the relay matrix board.

The algorithm checks the circuit to make sure that the remote web user has not created short circuits. The CGI string is sent to the Web-based microcontroller server when the remote web user presses the complete circuit button on the Web page.

Once this validation takes place, the microcontroller resets the matrix board in preparation for the connection. The microcontroller places the binary string on the controller data bus of the switching matrix board in order to assert the appropriate relay signals and to complete the switching. The Web-based microcontroller server receives the end of measurement from the I/O of the switching matrix board, pulses for 10 ms, and takes any measurement specified by the remote user and generates an HTML page with the results from the operation.

3) Internet-based Data Transfer: All the instruments and some components use GPIB interface to translate instruments commands and controls, and data transfer from Internet-protocol to local bus protocol (example from Ethernet to GPIB). The Fig. 6 (Appendix IV) shows a typical connection configuration [19] used to link the matrix controller board, instrument(s), the host PC, and the remote PC. Webcams, also considered as components, can be used for visual control and image recordings of the physical readings and state of instruments and components in the system. The number of instruments and components that can be added is only limited by the data transfer protocol of the GPIB software. For

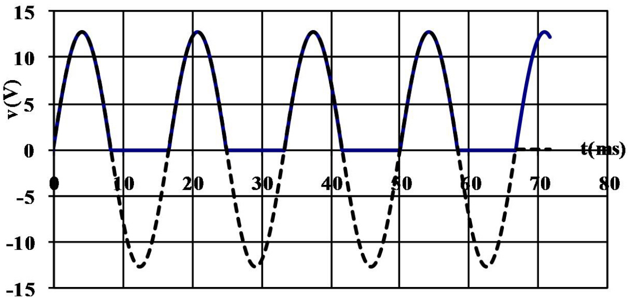


Fig. 8: Output and Diode Voltage

example, readings from analog-to-digital converters were also displayed in an html page. The Java applet sent the following CGI-string to the Web-based microcontroller:

http://141.218.115.11/ref.cgi?C0=7&&C1=288&&C2=298&
&C3 =320&&C4 =329&&C5=305&&C6
=314&AD1=1&AD2=0

where AD1 and AD2 represent two nodes in the network. These are integer values between 0 and 7 that tells the microcontroller to read the analog-to-digital converter lines AD1 and AD2

The data and file transfer between the host PC and the remote PC is by means of common Internet data interchange techniques; file transfer protocol (ftp), hyper-text transfer protocol (http), and any local Windows network protocols. However, it may be necessary to provide login procedure and protection for the switching system.

4) Interactivity: The Fig. 7 (shown in Appendix IV) shows how a pop up menu on the web allows a remote user to control the ac waveform generator. The controls at the remote user do not depend on any third party protocol (only on HTTP) or special network architecture. Instruments and some components have their own IP addresses and with web update cycles for all instruments/components (including the Webcams) in sequence. In this way there is no interference of the updates of instruments/components if they are requested at the same time. An instrument/component reading and its webcam are synchronously updated within two seconds. When an instrument is requested, a window pops up showing the GPIB controls for the instrument and the reading of the instrument.

In Fig. 8 a webcam is used to show the live ac source face readings as the remote web user interacts with the controls on the web.

The reading of any instrument can be changed by the controls on the web. The webcam, when activated, will simultaneously show live the changes in the reading (or waveform) of the instrument. An instrument/component or a webcam must be requested by clicking on it in the Web window. The output waveforms obtained from the data of the Web-based laboratory are shown in Fig. 8: the output voltage and voltage across one of the diode. These waveforms are exactly the same waveforms obtained from a “normal” laboratory session.

V. CONCLUSIONS

The above-mentioned process of connecting instruments and components on a 3-dimensional matrix board with Web access has a lot of advantages. It is simple, does not depend on any third party or proprietary protocol, and it does not depend on any special network. The algorithm is very innovative in facilitating the remote commands to connect real-world instruments and components. This system will work for any instruments or components with any number of terminals. This system will work for remote instrumentation or applications where it is necessary to translate instrument commands from Internet-protocol to local bus protocol such as from Ethernet to GPIB. The students performing real-time remote lab will it convenient, save time, and will not be under tremendous pressure. The remote students will make the same mistakes as the student in real lab environment. This will also be very convenient for the non-traditional students.

APPENDIX I

TABLE III: Initialization File for an 10 x 8 Relay Switching Matrix Board Example

Comment	Description	Rows									
Name	Instrument/Component	0	1	2	3	4	5	6	7	8	9
!	rwm_init file										
!	Ground										
G	00	0	-1	-1	-1	-1	-1	-1	-1	-1	-1
!	AC Power Supply										
S	AC	-1	-1	-1	-1	4	5	-1	-1	-1	-1
!	Diode										
DIODE	diode	-1	-1	-1	-1	-1	-1	6	7	-1	-1
!	Resistor										
R	1K	-1	-1	-1	-1	-1	-1	-1	-1	8	9

APPENDIX II

TABLE IV: Virtual to Physical Translation and Binary Representation of CGI String

Component			Translation										Binary Representation	Value
Name	Row	Column	CLK	RST	bit4	bit 3	bit 2	bit 1	bit 0	Col2	Col1	Col0		
acL	4	0	0	1	0	0	1	0	0	0	0	0	0100100000	288
acR	5	2	0	1	0	0	1	0	1	0	1	0	0100101010	298
diodeL	8	0	0	1	0	1	0	0	0	0	0	0	0101000000	320
diodeR	9	1	0	1	0	1	0	0	1	0	0	1	0101001001	329
1kΩL	6	1	0	1	0	0	1	1	0	0	0	1	0100110001	305
1kΩR	7	2	0	1	0	0	1	1	1	0	1	0	0100111010	314

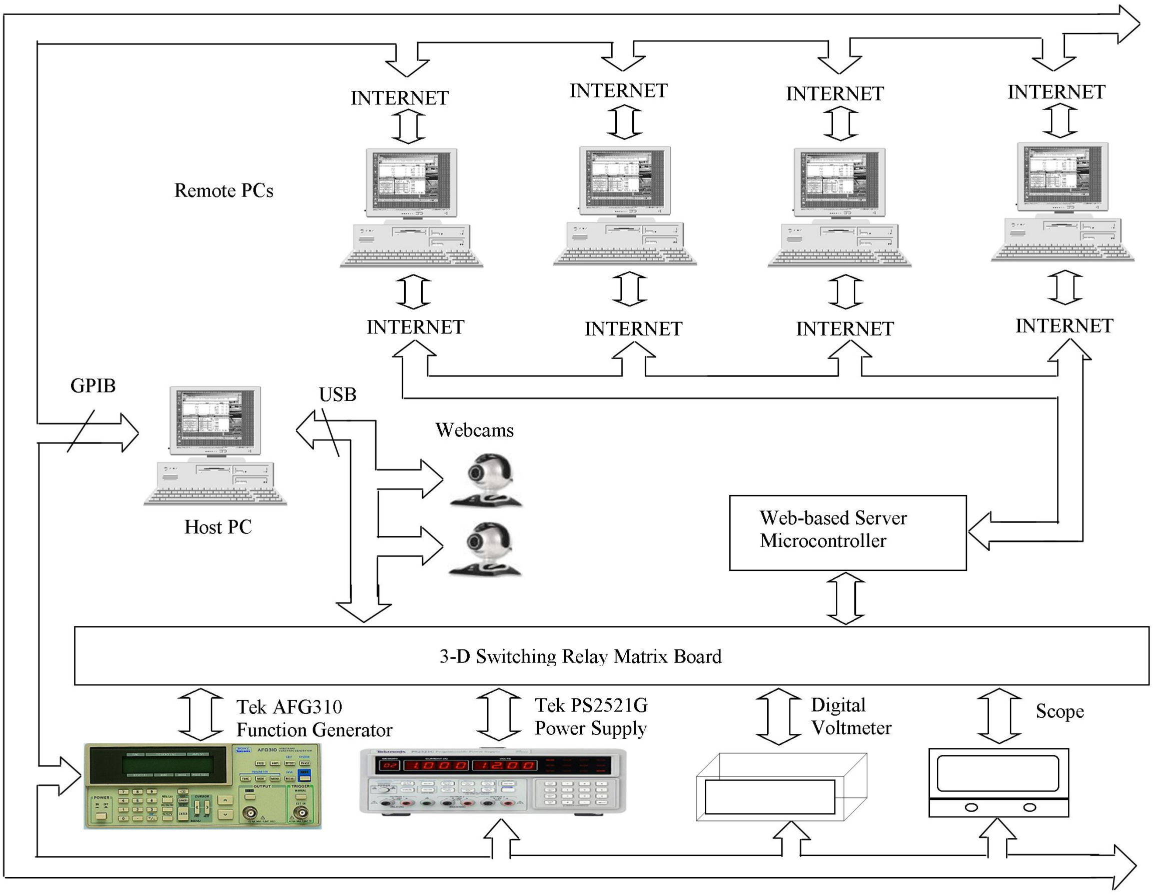


Fig. 2: Distributed Instrument Architecture with Matrix Board, Controller, Instrument(s), Host PC, Webcams, and Remote PC

APPENDIX III

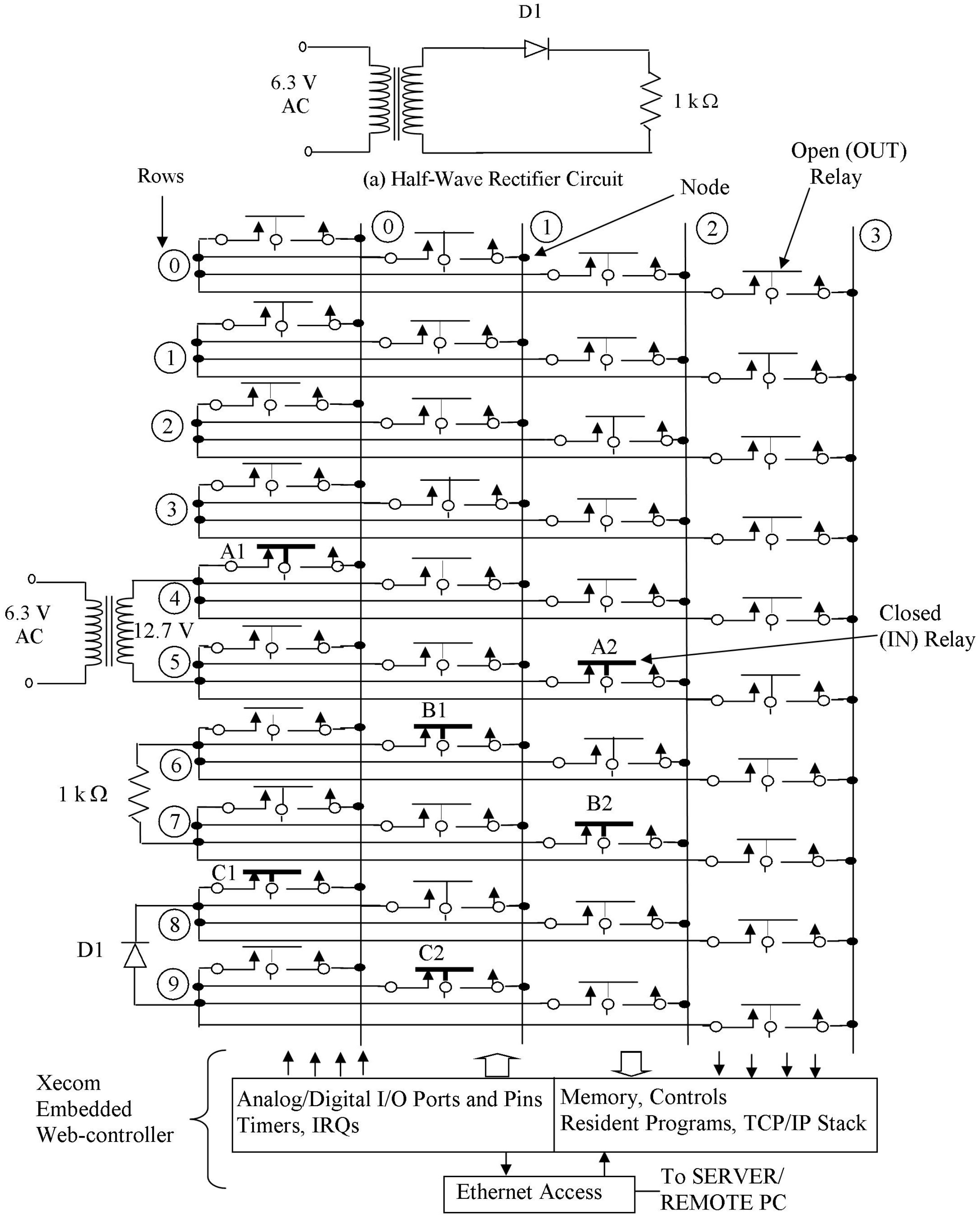


Fig. 4: Features of the PermLAB

APPENDIX IV

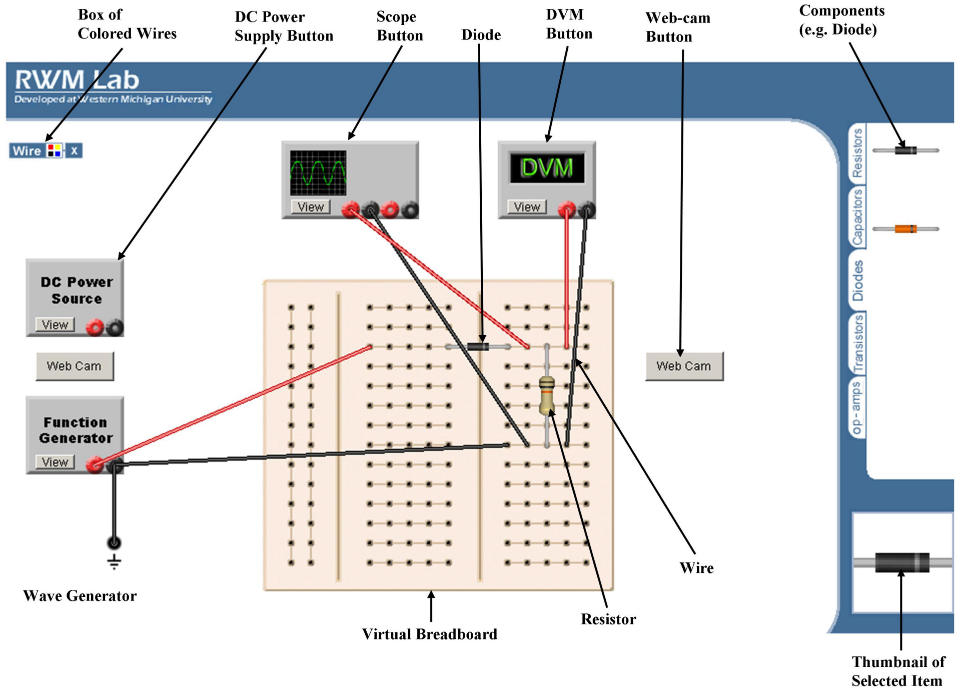
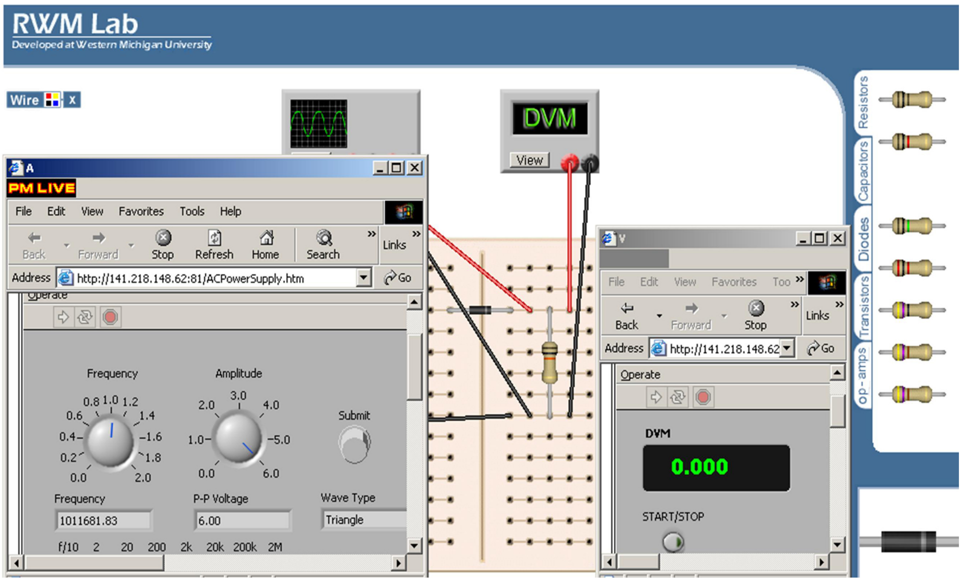


Fig. 6: Web Capture of Wired Half-wave Rectifier.


RWM Lab
Developed at Western Michigan University
Wire
x
DVM
View
PM LIVE
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search
Links
Go
Operate
Submit
START/STOP
Resistors
Capacitors
Diodes
Transistors
op-amps

Address http://141.218.148.62:81/ACPowerSupply.htm

Frequency 0.0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0

Amplitude 0.0 1.0 2.0 3.0 4.0 5.0 6.0

Frequency 1011681.83

P-P Voltage 6.00

Wave Type Triangle

f/10 2 20 200 2k 20k 200k 2M

Address http://141.218.148.62

DVM 0.000

Fig. 7: GPIB Controls and Display, and Live Web-cam Capture of Waveform Generator and DVM

References

[1] G. Fortino and L. Nigro, "Development of virtual data acquisition systems based on multimedia internetworking," *Computer Standards and Interfaces*, Elseviar, Vol. 21, pp. 429 – 440, 1999.

[2] W. Holfelder, "Interactive remote recording and playback of multicast videoconferences," *Proc. of 4th International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS'97)*, Darmstadt, Germany, pp. 450 – 463, Sep. 1997.

[3] G. Fortino and L. Nigro, "A measurement on-demand service for access and delivery process acquisition data," *Proceedings of the 17th IEEE Instrumentation and Measurement Conference*, Vol. 2, pp. 661 – 666, May 2000.

[4] J. Delsing, N. K. Hyyppa, and T. Isaksson, "The IP-meter, design concept and example implementation of an internet enabled power line quality meter," *Proceedings of the 17th IEEE Instrumentation and Measurement Conference*, Vol. 2, pp. 657 – 660, May 2000.

[5] M. Bertocco, F. Farraris, C. Offelli, and M. Parvis, "A client-server architecture for distributed measurements systems," *IEEE Tr. on Instrumentation and Measurement*, Vol. 47, No. 4, pp. 1143 – 1148, Aug. 1998.

[6] VMEbus Extensions for Instrumentation. "TCP/IP instrument protocol specifications VXI-11," Revision 1, http://www.vxibus.org/?q=node/206, pp. 1 – 66, Jul. 1995.

[7] A. Ferrero, V. Piuri, "A simulation tool for virtual laboratory experiments in WWW Environment," *IEEE Transactions on Instrumentation and Measurement,* Vol. 48, Issue 3, pp. 741 - 746, Jun. 1999.

[8] M. Bertocco, "Platform independent architecture for distributed measurement systems," *Proceedings of the 17th IEEE Instrumentation and Measurement Conference*, Vol. 2, pp. 648 – 651, May 2000.

[9] J. A. Asumadu, R. Tanner, J. Fitzmaurice, M. Kelly, H. Ogunleye, J. Belter, and S. C. Koh, "A web-based hands-on real-time electrical and electronics remote wiring and measurement laboratory (*RwmLAB*) instrument," *Proceedings of the 20th IEEE Instrumentation and Measurement Conference*, Vol. 2, pp. 1032 – 1036, May 2003.

[10] A. Ferrero, S. Salicone, C. Bobora, and M. Parmigiani, "REMLab: Java-based remote, didactic measurement laboratory," *IEEE Trans. on Inst. Meas.*, Vol. 52, No. 3, pp. 710 -715, Jun. 2003.

[11] L. Benetazzo, M. Bertocco, F. Ferraris, C. Offelli, M. Parvis, and V. Piuri, "A web-based distributed virtual laboratory," *IEEE Trans. on Inst. Meas.*, Vol. 49, pp. 349 - 356, Apr. 2000.

[12] C. C. Ko, B. M. Chen, S. H. Chen, V. Ramakrishnan, R. Chen, S. Y. Hu, and Y. Zhuang, "A large-scale web-based virtual oscilloscope laboratory experiment," *Eng. Sci. Education J.*, Vol. 9, No. 2 pp. 69 – 76, 2000.

[13] M. Kenney, P. De Bievre, and A. Picard, "A web-based database for international programme to improve the avogadro constant along the silicone route," *IEEE Trans. on Inst. Meas.*, Vol. 52, No. 2, pp. 641 - 645, Apr. 2003.

[14] G. Bucci and C. Landi, "Distributed measurement and architecture for industrial applications," *IEEE Trans. on Inst. Meas.*, Vol. 52, No. 1 pp. 165 - 174, Feb. 2003.

[15] C. Quitans, M. J. Moure, M. D. Pena, and E. Mandabo, "A virtual instrumentation laboratory based on a reconfigurable coprocessor," *IEEE Trans. on Inst. Meas.*, Vol. 55, No. 2 pp. 165 - 174, Apr. 2006.

[16] G. Andria, A. Baccigallupi, et la, "Remote didactic laboratory "G. Savastano," the Italian experience for e-learning at the technical universities in the field of electrical and electronic measurement: architecture and optimization of the communication performance based on thin client technology," *IEEE Trans. on Inst. Meas.*, Vol. 56, No. 4 pp. 1124 - 1134, Aug. 2007.

[17] G. Andria, A. Baccigallupi, et la, "Remote didactic laboratory "G. Savastano," the Italian experience for e-learning at the technical universities in the field of electrical and electronic measurements: overview on didactic experiments," *IEEE Trans. on Inst. Meas.*, Vol. 56, No. 4 pp. 1135 - 1147, Aug. 2007.

[18] F. M. Schaf and C. E. Pereira, "Integrating mixed-reality remote experiments into virtual learning environments using interchangeable components," *IEEE Trans. On Indust. Electr,* Vol. 56, No. 12, pp. 4776 – 4783, Dec. 2009.

[19] L. Lo Bello, O. Mirabella, A. Raucea, and L. Capetta, "ENELPILOT: From a research testbed to virtual educational laboratory," *IEEE Trans. On Indust. Electr,* Vol. 56, No. 12, pp. 4844 – 4853, Dec. 2009

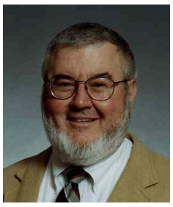
[20] P. Andre and V. Rajagopalan, "Web-based learning for power electronics: A case study", *Proceedings of National Science Foundation Workshop on Multimedia Delivery of Power Electronics Education*, Nov. 2000.

[21] Proceedings of NSF-sponsored faculty workshop on teaching of courses in power electronics and electric drives, tempe, http://www.ece.umn.edu/groups/workshop2003/ Jan. 2003.

[22] T. Undeland and N. Mohan - Meeting the challenge: revitalizing courses in power electronics and electric drives - http://www.ets.kth.se/eme/norpie/program_detail.pdf

[23] C. Cassaigo, L. Callegaro, and G. La Paglia, "Internet calibration of electrical metrology: first application at IEN," *Proceedings of the 21st IEEE Instrumentation and Technology Conference,* Vol. 1, pp. 102 – 105, May 2004.

Johnson A. Asumadu (S'82, M'94, SM'00) received the B.Sc. from the University of Science Technology, Ghana in 1975, M.Sc. from Aston University, England in 1978, MEE from the Rensselaer Polytechnic Institute, USA in 1983, and Ph.D. from the University of Missouri-Columbia, USA in 1987. He was a Commonwealth Scholar in England in 1976 and a Fulbright Scholar in USA in 1982. He is now a faculty at the Western Michigan University, Michigan. He has published and co-authored several technical papers and books, has patents, and has been involved in numerous government and industrial-sponsored projects in areas of power systems, power electronics, and instrumentation. His research interests include power electronics, control engineering including fuzzy logic applications, and microprocessor/microcontroller embedded applications. Dr. Asumadu is a member of IEEE Power Electronics Society, IEEE Instrumentation and Measurements, IEEE Power Systems Society, IEEE Industrial Electronics Society, and IEEE Control Systems Society, and was IEEE Region 4 Communications Director (1998 – 2002).



Dr. Ralph Tanner has taught at Western Michigan University since 1986. Before joining the academic ranks, Dr. Tanner worked as an engineer and manager in the automotive, electronics, and aerospace industries for over twenty-years. At Western Michigan University, he has taught nearly twenty courses at all levels from freshman courses to graduate courses. He has acted as an academic advisor at the graduate and undergraduate level. Dr. Tanner's research interests are primarily in the area of robotics, spanning from industrial applications to walking devices. Dr. Tanner has served the university in numerous roles including as the President of the Faculty Senate. He was a Research Ethics Fellow with the Center for the Study of Ethics in Society. In this role, he teamed with a graduate student Research Ethics Fellow to study the ethical dimensions of research as it relates to mentoring, intellectual property rights, the use of human subjects in research, and conflicts of interest. Dr. Tanner is a licensed Professional Engineer, a Certified Manufacturing Engineer, and a Professional Registered Parliamentarian.

Hakeem Ogunleye obtained a Bachelors of Science degree in Computer Engineering from Western Michigan University in 2002. He obtained a Masters degree in Electrical and Computer Engineering from Purdue University, West Lafayette in 2003. He also holds a graduate certificate in Software Engineering from University of Illinois, Urbana-Champaign. He worked at Cummins Inc. as a Controls Engineer from 2004 to 2009. He currently consults on using Information and Communication Technologies to automate, streamline and optimize business processes in Nigeria. His interests are in computer architecture, automation, controls, network security and software engineering.