Mobile Learning for Engineering Education in Jordan

A. Y. Al-Zoubi
Princess Sumaya University for Technology
Jordan

Abstract
A new architecture for the development of a wireless remotely controlled laboratory with focus on educational applications in electronics engineering is presented. The Internet is used as the communication infrastructure to enable remote students to access experimental equipment via mobile devices. The remote lab aims to support access of clients running on PCs or Personal Digital Assistant (PDA) devices while the server implementation was based on LabVIEW programming language. Experimental tools were created to allow users to collect data and information about the experiments, giving encouraging initial results where students are able to undertake simple engineering remote experimentation. Further experiments are envisaged and international collaboration is underway.

Introduction
The internet was first used in 1994 as a tele-control medium and subsequently applied in an educational context in 1996 when web-based laboratories were introduced by universities worldwide in undergraduate engineering courses [1-3]. The evidence that the field of remote engineering has matured are overwhelming, particularly as indicated by the number of remote laboratories in operation today. Furthermore, the range of disciplines being taught continue to grow and collaborations between universities all over the world are becoming increasingly common [4]. Related training courses have also been explored in chemistry, physics, and electrical engineering and other interesting setups include remote experiment and virtual labs for wind tunnels, a virtual laboratory for exploiting DSP algorithms and a learning tool for chip manufacturing [5]. Recently, a distributed laboratory was developed for the execution of virtual scientific experiments superimposed on a Grid infrastructure as part of a master degree in information networking [6]. The opportunity to provide students with remote access to experimental hardware and the ability to offer flexibility in time and place in which laboratories are conducted are also becoming powerful motivations for the field [7].
Additionally, the recent advances in mobile technologies have lead to an increase in the number of internet users accessing information via mobile devices and the number of applications designed for such devices is growing and becoming increasingly popular. With the release of Visual Studio .NET 2003, it was possible to create applications to run on resource-constrained devices, in almost the same way Windows application are created. These applications are built for the .NET Compact Framework that includes a large selection of framework classes and optimized for the small screen resolutions of handheld devices. From the pedagogical viewpoint, students' expectations on how and when they learn are also creating increasingly heavier demands upon all aspects of their learning as young people are making mobile devices an extension of their personal space and fundamental to their daily lives. In response, the world is moving very rapidly to engage with the opportunities and flexibility offered by mobile technologies [8].
Mobile remote solutions may therefore be attractive tools in enhancing access to practical experiments as they offer many different possibilities for applications in industry and

education because they are not subjected to limitations of location and time [9-10]. This, in combination with today's easy access to broadband internet, is transforming the way e-learning is carried out, allowing a higher level of interactivity and providing virtual environments closer to real ones. Hence, mobile remote systems can be very useful when applied to solutions involving high costs of people and equipment transportation. Different universities and institutions could share laboratory resources, expensive equipments and experiments by means of a cooperative network of remote systems.

In this paper, the possibility for students to access remote experiments via mobile devices is presented. The proposed approach is to implement a client Personal Digital Assistants (PDA) which may be programmed in LabVIEW, with a module that suits applications on handheld devices. Communication between the server and the mobile client was based on direct TCP/IP programming to facilitate data transfer through the internet. Experimental tools were created to allow the user to collect data and information about the experiments and a web-based feedback system was developed to provide anonymous feedback from students. Initial results are very encouraging, with students able to undertake simple engineering remote experimentation and further experiments and international collaborations are underway.

Architecture of the Mobile Remote Lab

An effective way of improving technology-enhanced engineering education is to combine theory and practice simultaneously in the same lesson. This may be achieved utilizing the graphical interface capability and web publishing facility of LabVIEW. In addition, a lesson may consist of text, figures and circuits and necessitates the use of multi-screens design. This requires the program to navigate from on page to another due to the limited capacity of the screen. Several methods have been implemented to manage this situation, including the use hyperlinks and buttons, as well as the attempt to interface LabView with other programming languages such as Java. The use of tab pages has however been found the most suitable convenient solution to navigate between pages in this particular application.

The application of interest here is to enable electrical engineering students to revise material anywhere at anytime. The material may be in the form of text, equations, figures, intimations, and circuits. However, it is also important for students to conduct experimental work virtually by simulation or remotely by accessing laboratory set-ups through the internet.

The most convenient and popular environment to implement virtual and remote experimentations is probably the Laboratory Virtual Instrumentation Engineering Workbench (LabVIEW) programming language, which has been extensively used for such purpose [1-2]. LabVIEW is high-level user-friendly graphical programming language developed by National Instrument (NI) for use by scientists and engineers, particularly for data processing and acquisitions applications. LabVIEW employs special controls to form instructions and icons to build subroutines, wired together in order to define the flow of data through the program. In this context, LabVIEW is a visual language and not text-based where the code is governed by a series of rules based syntax such as commas, periods, semicolons, square brackets, curly brackets, round brackets, and others. To the contrary, LabVIEW flowcharts the code as it is written to produce a program in a timely and efficient manner. This programming approach is based on building blocks called Virtual Instruments (VIs); each contains three main parts: front panel; block diagram; and icon/connector. The front panel is a means of interaction between the user and the block diagram (program) when the program is running. Users may utilize the front panel to control the program, alter inputs, and monitor changes, updates and other variations in real time. Indicators are program outputs used to show the status of program variables, parameters, flags or other types of data, states, and other information. Front panel objects appear as terminals on the block diagram where every control and indicator has a corresponding terminal which contains the graphical source code.

Additionally, certain functions and structures which reside in built-in LabVIEW VI libraries are used in the block diagram.

A special add-on LabVIEW module has also been devised to allow one to run LabVIEW virtual instruments (VIs) on PDA execution targets. The module is a good tool for creating data acquisition and remote system monitoring applications that are both portable and flexible. The LabVIEW PDA module is usually operated with an emulator that can be used to test the application inside a simulated environment to mimic the behavior of the actual PDA. It gives additional flexibility in the design and testing and establishes greater confidence that the applications will behave as intended, and hence the lesson is tested prior to loading into the actual PDA. It facilitates the development of graphical environment to create custom applications for a multitude of mobile devices and PDAs. The programmer is therefore given the choice to select of the appropriate operating system that will be implemented in the design before commencing with writing the program.

The system was modified further into the architecture shown in Fig. (3) to perform remote real electronic experiments via the internet. This architecture shows the methodology of how remote clients will connect to the remote lab. The user interface and the control of the lab hardware were developed with LabVIEW virtual instruments (VIs) to design front panels which resemble the front view of a real oscilloscope and a function generator, for example, and to have nearly the same functionalities as real devices. These VIs were then embedded into HTML and published in the Web server which also hosts the READ laboratory home page with information about the implemented exercises, as well as the links to system login. The front panels could then be remotely reached via a standard Web browser which does not require of the user any prior knowledge of LabVIEW to take advantage of the system facilities. In fact, the client simply needs a standard Web browser and a proper version of LabVIEW run-time engine that contains some libraries and web browses' plug-ins. This software is available free of costs at the National Instruments website and is automatically downloaded on the first attempt to access the system.

In addition, the driver software has an application programming interface (API), which is a library of VIs, functions, classes, attributes, and properties for creating applications. This library of polymorphic VIs replaces large parts of a complex DAQ application with a simple configuration dialog. The responses are read, for example, with the aid of the NI DAQmx driver software. Only one user can however access the remote lab at a time, therefore each user is allocated a thirty minutes time-slot to access the system by first inserting a username and a password. If both are valid, the system enables access to the lab.

Students may be able to enter READ and carry out analog electronics experiments from anywhere at anytime by simply modifying the system layout to include access via wireless devices as shown in the architecture of Fig. (4). The proposed approach is to implement a client PDA which may communicate with the Web and VI severs directly with transmission control protocol (TCP), internet protocol (IP), or user datagram protocol (UDP), being the basic standard protocols for network communications and data transfer through the internet. TCP/IP communication provides a simple user interface that conceals the complexities of ensuring reliable network communications. TCP describes communication between applications and permits multiple simultaneous connections which must be established prior to data transmission and thus necessitating the specification of an address and a port at that address when a client requires communicating with a server. A connection is initiated by waiting for an incoming connection or by actively seeking a connection with that specified address. After a "handshake" between the two applications, TCP sets up "full-duplex" communication between the client and a server. On the other hand, IP is a “connection-less” communication protocol responsible for routing a packet to its destination. LabVIEW offers an API for developing applications that includes TCP/IP and UDP functions used to create client and server VIs on all platforms.

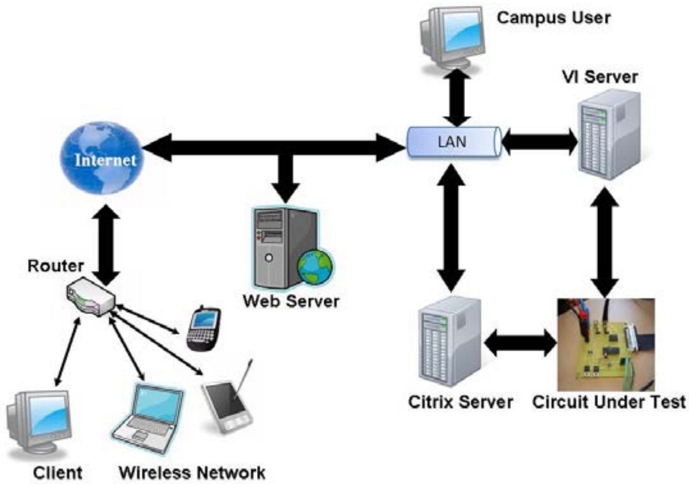


Fig. (4). Mobile Remote Laboratory Architecture.

Shared variables is another technique which may be implemented to transmit data between multiple computers using simple coding techniques as this technique manages network communication and bindings, which can be configured from within LabView. Shared variables are an easy way to transfer data between targets over a network due to its high level, easy configuration and setup interfaces. Shared variables use the Publish-Subscribe Protocol (PSP) which is a form of User Datagram Protocol (UDP). It is necessary to separately install a Shared Variable Engine (SVE) where a variable resides on a computer in order to share variables in standalone applications. For PDA devices, it is also necessary to install the support for shared variables. Shared variable servers use UDP port 2343 and a range of UDP ports from 6000 to 6010 for incoming and outgoing packets whereas clients use a range of UDP ports beginning with port 5000. The number of ports above 6000 that the network-published shared variable servers use depends on the number of servers running on the computer. The NI-PSP protocol also uses TCP ports and it begins looking for available TCP ports at port 59110 and increments upward until it finds an available port. The following table shows a comparison between TCP/IP and shared variables.

The design of the mobile remote lab in this work was focused on the use of TCT/IP communication technology rather than SV, despite its obvious advantages, in order to avoid making appropriate assumptions about communication policies between mobile devices with a server that may be assigned several port numbers by SV tools. TCP/IP communication was also preferred because portability and firewall rules are very important issues when developing such a system. Several problems have been experienced when attempting to access labs via SVE from different networks, especially networks from institutions that usually are under a strict firewall policy.

The process of client/server communication is described in the flowchart of Fig. (5), where it can be seen that the system was assigned standard port (80). In this basic sequence of operation, the server starts to listen to port 80. When it detects a connection from a client, it begins reading the parameters for signal generation from the socket. The data must however be converted to XML string type prior to transmission. The length of the string must first be sent followed by the data itself. Therefore, the client converts all parameters to string types and transmits them via TCP/IP. The server then calls a sub VI that handles generation and acquisition operations. A waveform signal is applied to the circuit and the responses are measured and stored to be sent back to the client. On the server, these parameters are read and converted back to their original type to be processed by the VIs controlling the data acquisition. The responses are then converted to string and sent back to the client. On the client side, these responses are converted back to their original types and displayed to the user. Each client connected has the requests processed during its time slice. While a client is connected, it is occupying a position in the queue and is consuming resources. Therefore, maximum connection duration of 30 minutes was allowed. The server processes client requests by applies the desired signal to the circuit under test and returns the measurements according to the simple process shows Fig. (6).

Clients were designed for PDAs as well as for Windows PCs and requests from both were treated seamlessly by the server. Due to resources constraints of PDA devices, not all the features designed for a client running on a PC are performed when accessing the system via PDAs, but this can be managed just by changing the client application. A simple experiment to generate a number of waveforms with varying amplitudes, frequencies and shapes was actually performed remotely on a PDA as shown in Fig. (7) to control and simulate an actual function generator. The most important consideration, however, was the limited resources of mobile devices compared to PCs which lead to a reduction of the features available for designing mobile remote laboratory. Nonetheless, the proposed solutions remain applicable to a number of experiments where initial results have been encouraging as students able to undertake simple engineering remote tasks. Further experiments are being prepared.

The LabVIEW PDA Module

A special add-on LabVIEW module has also been devised by National Instruments to allow one to run LabVIEW virtual instruments (VIs) on PDA execution targets. The module is a good tool for creating data acquisition and remote system monitoring applications that are both portable and flexible. The LabVIEW PDA module is usually operated with an emulator that can be used to test the application inside a simulated environment to mimic the behavior of the actual PDA. It gives additional flexibility in the design and testing and establishes greater confidence that the applications will behave as intended. It facilitates the development of graphical environments to create custom applications for a multitude of mobile devices and PDAs. The programmer is thus given the choice to select the appropriate operating system that will be implemented in the design before commencing with writing the program. In addition, the LabVIEW PDA module allows the creation of custom, user-defined applications for Palm, Windows Mobile for Pocket PC and Windows CE devices. This can be achieved with a LabVIEW programming environment on the same way it is made for a PC application but deployed into a PDA. It also allows the development of data acquisition applications with Compact Flash and PCMCIA DAQ cards. Furthermore, the PDA module includes some libraries of sub VIs developed to take advantages of additional resources available on PDAs and Smartphone, like short message services (SMS) and telephony. Besides, it is also possible to use most of the known functions and APIs available when developing applications for PCs with the LabVIEW PDA module for the most common operating systems found on these devices, like Windows Mobile, Palm OS and to deploy it for emulated devices or real ones via the proper synchronization tool like Microsoft Activesync for Windows Mobile. Due to the limited graphical capabilities of these devices, however, the controls and indicators are sized and scaled and the functions palettes are reconfigured.

Programme Design

A new PDA application is built on the LabVIEW when the programmer selects a PDA project from the targets list on the “getting started” screen as shown in Fig. (1a). The next screen is a “create new PDA project”, which allows the selection of a project type from project type list. A new VI can be created by selecting ”blank VI”, whereas import VI is used to open an existing VI template, determine the project location and VI path. The project files are then to created on the selected path as shown in Fig. (1b). The device type is subsequently chosen from a list which conforms with actual the PDA operating system Fig. (1c). Another list also appears to select VI template as a pocket PC landscape screen or pocket PC portrait screen Fig. (1d).

The application of interest here is to enable electrical engineering students to access the content of the “Analog Electronic” course at Princess Sumaya University for Technology. The main part of the course focuses on the design, analysis and applications operational amplifiers in electrical systems. The material usually takes the form of text, equations, figures, animations, circuits’ diagrams, displays and graphs. This requires the program to navigate from on page to another due to the limited capacity of the screen. Several methods have been implemented to manage this situation, including the use hyperlinks and buttons, as well as the attempt to interface LabView with other programming languages such as Java. The use of tab pages has however been found the most suitable convenient solution to navigate between pages in this particular application.

The first tab page contains an introduction about the subject of the lesson. The student can revise the lesson before conducting experiment. The next tab is op amp circuit figure with variable controller and scope. The user can see the circuit and set the variables; the frequency and amplitude for the input signal, the value of resistances. The program then generates the signal from there set values of frequency and amplitude, and feeds this signal to input of the op amp to produced the output, when the ‘Show’ button clicked then the op amp generate the output signal and draw it on the scope as shown in Fig. (2). In the result tab page the user can read a conclusion for the experiment and make an offline quiz in the last tab page. This tab page is designed to ensure students’ understanding and comprehension of the experiment. The student can navigate tab pages forward and backward and content of the lesson at any time.

In the result tab page the user can read a conclusion for the experiment and make an offline quiz in the last tab page. This tab page is designed to ensure students’ understanding and comprehension of the lesson and experiment. The student can navigate tab pages forward and backward and content of the lesson at any time.

Results and Discussion


back
play

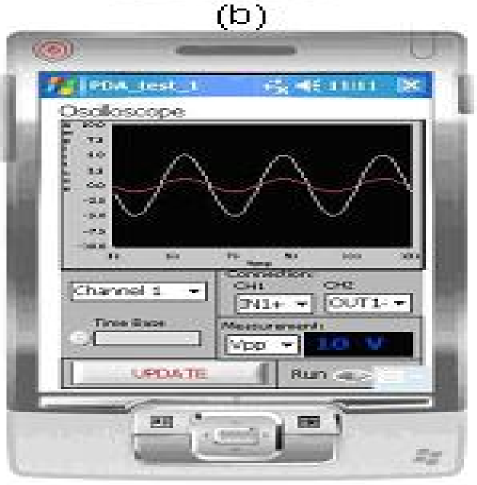
(a)


Questions
Q5.Advantages Of DBMS:
A1. Atomicity.
A2. Security.
A3.A1 AND A2
A4. None Of the above.
Previous
Next
Finish

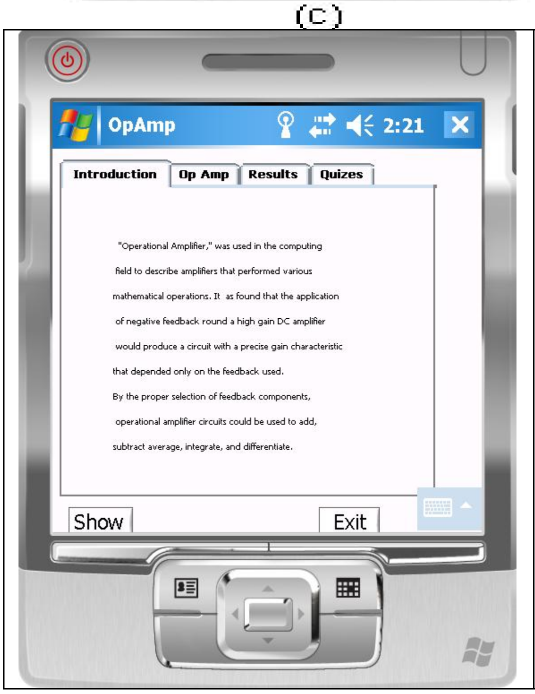
(b)


Start
PSUT Mobile Virtual Lab
Exit
Start
Phone
Contacts

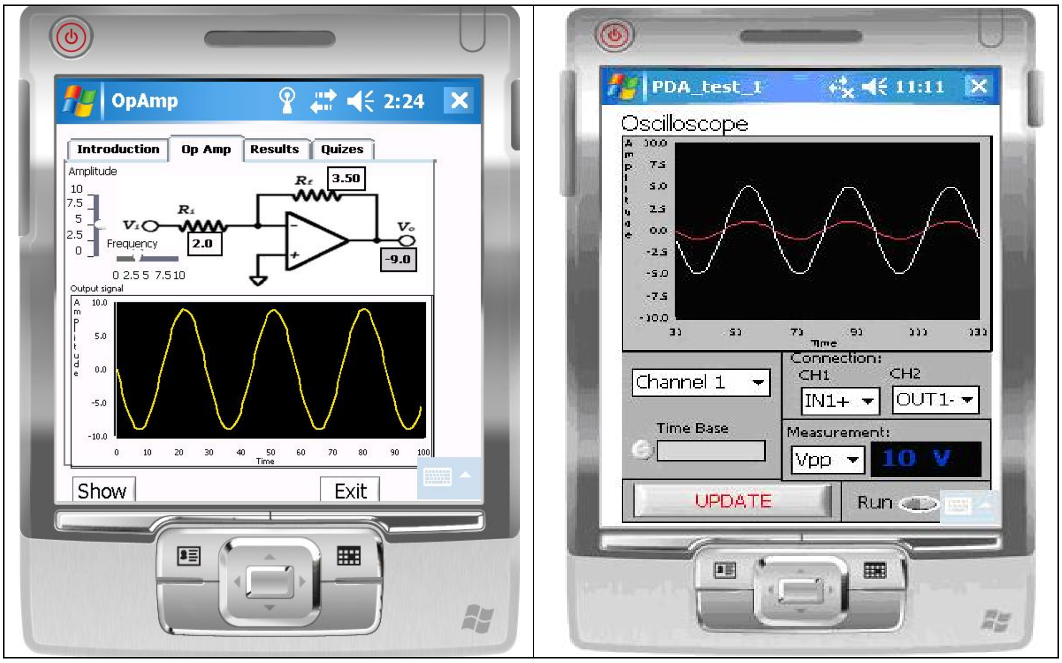
(c)


PDA_test_1
Oscilloscope
Channel 1
Connection:
CH1
CH2
IN1+
OUT1-
Time Base
Measurement:
Vpp
1.0 V
UPDATE
Run

(d)


OpAmp
2:21
Introduction
Op Amp
Results
Quizes
"Operational Amplifier," was used in the computing
field to describe amplifiers that performed various
mathematical operations. It as found that the application
of negative feedback round a high gain DC amplifier
would produce a circuit with a precise gain characteristic
that depended only on the feedback used.
By the proper selection of feedback components,
operational amplifier circuits could be used to add,
subtract average, integrate, and differentiate.
Show
Exit


OpAmp
2:23
Introduction
Op Amp
Results
Quizes
Amplitude
10
7.5
5
2.5
0
Frequency
0 2.5 5 7.5 10
Output signal
0.00
0.0
Time
Show
Exit



Conclusions

A new wireless remote lab system was developed to enable students to access experiments via the internet from their mobile devices. The types of experiments under consideration were mainly electronic circuits at the junior undergraduate level for electrical engineering students at the Carinthia University of Applied Sciences and Princess Sumaya University for Technology. Student clients were able to access the remote lab using a PDA which communicates with the server controlling the equipment through the simple TCP/IP protocol because of its portability and the simple firewall rules governing its operation. This has been an important consideration compared to the use of shared variable techniques which experience several problems when attempting to access labs from different networks, especially from institutions that usually place strict firewall policy.

The most important consideration, however, was the limited resources of mobile devices compared to PCs which lead to a reduction of the features available for designing mobile remote laboratory. The proposed solutions however remain applicable to other labs and experiments because they are easy to adopt standard solution. The initial results are very encouraging, with students able to undertake simple engineering remote experimentation. It can be still argued that learning environments will remain a combination of wired and wireless for the foreseeable future as not all affordances offered by wired environments are transferable to small mobile devices. Further experiments and international collaborations are underway.

References

[1] BurGin Aktan, Carisa A. Bohus, Lawrence A. Crowl, and Molly H. Shor, "Distance Learning Applied to Control Engineering laboratories", IEEE Transaction on Education, Vol. 39, No. 3, AUGUST 1996, pp 320-326.

[2] C. Lon Enloe, William A. Pakula, Greg A. Finney, and Ryan K. Haaland, "Teleoperation in the Undergraduate Physics Laboratory-Teaching an Old Dog New Tricks", IEEE Transaction on Education, Vol. 42, No. 3, August 1999, pp 174- 179.

[3] Hong Shen, Zheng Xu, B. Dalager, V. Kristiansen, Ø. Strøm, Michael S. Shur, Tor A. Fjeldly, Jian-Qiang L¨u and T. Ytterdal, "Conducting Laboratory Experiments over the Internet", IEEE Transaction on Education, Vol. 42, N0. 3, August 1999, pp 180-185.
[4] Euan D. Lindsay and Malcolm C. Good, "Effects of Laboratory Access Modes Upon Learning Outcomes", IEEE Transaction on Education, Vol. 48, No. 4, November 2005, pp 619-631.
[5] Sabina Jeschke, A. Y. Al-Zoubi, Olivier Pfeiffer, Nicole Natho, and Jarir Nsour, "Classroom-Laboratory Interaction in an Electronic Engineering Course, 5th International IEEE Conference on Innovations and Information Technology, 16-18 December 2008, Al-Ain, United Arab Emirates.
[6] Ioannos Christou, Sofoklis Efremidis, Thanassis Tiropanis and Antonis Kalis, "Grid-Based Virtual Laboratory Experiments for a Graduate Course on Sensor networks", IEEE Transaction on Education, Vol. 50, N0. 1, February 2007, pp 17-26.
[7] A. Y. Al-Zoubi, Ashraf Tahat and Omar Hassan "Mobile Virtual Experimentation Utilizing SMS", Proceedings of the 4th IASTED International Conference on Communications, Internet, and Information Technology, pp 131-134, Cambridge, MA, USA, 31 October- 2 November 2005.
[8] David M. Kennedy and Doug Vogel, "Designing for Flexible Learning: Developing Wired and Wireless Applications", 3rd International Conference on Web Information Systems and Technologies (WEBIST), 3-6 March, 2007, Barcelona, Spain. www.webist.org/index.htm.
[9] Zoe Doulgeri and Tilemachos Matiakis, "A Web Telerobotic System to Teach Industrial Robot Path Planning and Control, IEEE Transaction on Education, Vol. 49, No. 2, May 2006, pp 263-270.
[10] Costas S. Tzafestas, Nektaria Palaiologou and Manthos Alifragis, "Virtual and Remote Robotic Laboratory: Comparative Experimental Evaluation", IEEE Transaction on Education, Vol. 49, No. 3, August 2006, pp 360-369.
[11] Michael Auer, A. Y. Al-Zoubi, Danilo Garbi Zutin and Hatem Bakhiet, "Design of Application-Specific Integrated Circuits for Implementation in a Network of Remote Labs", ASEE2008 Annual Conference and Exposition, Pittsburgh, Pennsylvania, USA, 22-25 June 2008.