Implementation of a Mobile Accessible Remote Lab

Michael E. Auer1, A. Y. Al-Zoubi2 and Danilo Garbi Zutin 1

[1] Carinthia University of Applied Sciences/School of Systems Engineering, Villach, Austria

[2] Princess Sumaya University for Technology, Amman, Jordan

***Abstract*—The purpose of the proposed research is to design and implement a LabVIEW-based remote lab client to run on a TCP/IP enabled PDA (Personal Digital Assistant) device, thus teaching using this wireless m-learning system will not be limited by time and location. In addition, resources and equipments can be integrated and shared to the extent that critically events can be monitored and handled in time. An environment will be created to train students to handle factory automation, data acquisition, data management, and manufacturing processes using mobile devices. Furthermore, the integration and sharing of lab equipments via the Internet is a kind of teaching environment which promotes learning interests and efficiency using mobile devices.**

Index Terms—Mobile Laboratories, Analog ASIC, Data Acquisition, LabVIEW, Remote Engineering, Remote Laboratories, Tele-learning, Virtual Instruments.

I. INTRODUCTION

Mobility is one of the present trends in all fields of communication, especially for example in Internet 2.0 applications.

Remote solutions applied to engineering teaching laboratories have been shown to be attractive tools in enhancing access to practical experiments. This, in combination with today's easy access to broadband internet connection, is transforming the way e-learning is carried out, allowing a higher level of interactivity and providing virtual environments closer to the real environments.

We can differentiate between mobile labs and mobile experimenters and our focus in this work is a mobile experimenter.

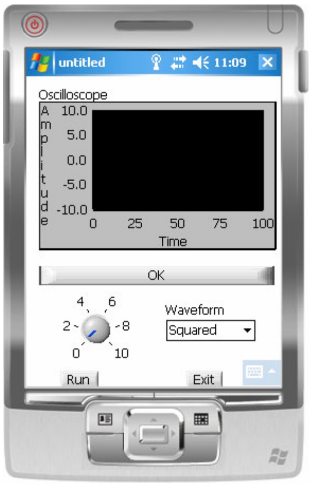
This work presents a possible implementation of a remote lab designed to support access of clients running on PCs or PDA (Personal Digital Assistants) devices. The server implementation does not differ and it treats requests form clients running on both platforms seamlessly. Its implementation is done fully on the transport layer (OSI Model) as it is based only on a simple TCP/IP communication and it doesn't use any other higher level protocols for data transference.

Due to recent advances in mobile devices like PDAs and Smartphones, the number of Internet users accessing information via mobile devices is increasing and this is pushing the sharing of information into another level. Along with these advances, the number of applications designed for such devices is growing. 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 you create a Windows application. These applications are built for the .NET Compact Framework that includes a large selection of Framework classes. The application is also optimized for the small screen resolutions of handheld devices.

Our approach to implement a client for PDA devices is based on LabVIEW. National Instruments offers a module within its LabVIEW suit to build applications for handheld devices.

A. *The LabVIEW PDA Module*

The LabVIEW PDA module allows the creation of custom, user-defined PDA applications for Palm, Windows Mobile for Pocket PC and Windows CE devices. This can be done by using the LabVIEW programming environment on the same way it is done for a PC application and deploying it to a PDA. It allows also the development of data acquisition applications with Compact Flash and PCMCIA DAQ cards.



Waveform Squared

Figure 1. Example of a VI deployed to a WM5 emulator

The LabVIEW PDA module also includes some libraries of sub VIs developed to take advantages of some resources available on PDAs and Smartphones, like short message services (SMS) and telephony. Besides that, it is also possible to use most of the known functions and APIs (Application Programming Interfaces) available when developing applications for PCs.

It is possible to deploy applications with the LabVIEW PDA module for the most common operating systems found for these devices, like Windows Mobile, Palm OS and furthermore it is also possible 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, the controls and indicators are sized and scaled and the functions pallete is reconfigured.

B. *The TCP/IP API of LabVIEW*

Internet Protocol (IP), User Datagram Protocol (UDP), and Transmission Control Protocol (TCP) are basic standards for the network communication.

TCP describes the communication between applications. When an application wants to communicate with another application via TCP, it sends a communication request. This request must be sent to an exact address. After a "handshake" between the two applications, TCP will setup a "full-duplex" communication between the two applications [8].

IP is responsible for routing a packet to its destination and it is a “connection-less” communication protocol.

LabVIEW offers an API (Application Programming Interface) for developing applications that includes TCP and UDP VIs and functions that can be used to create client or server VIs and it is possible to use TCP/IP protocols with LabVIEW on all platforms.

TCP is a connection-based protocol, which means that a connection must be established before transferring data. The data transmission occurs between a client and a server. TCP permits multiple simultaneous connections.

A connection is initiated by waiting for an incoming connection or by actively seeking a connection with a specified address. In establishing TCP connections, it is necessary to specify the address and a port at that address. Different ports at a given address identify different services at that address. TCP/IP communication provides a simple user interface that conceals the complexities of ensuring reliable network communications.

With the API for TCP/IP communication of LabVIEW it is possible to develop custom applications for TCP/IP communication. The programmer is responsible for developing both the client and the server.

II. SYSTEM DESCRIPTION

The CTI (Carinthia University of Applied Sciences) developed some solutions to deliver electronics related experiments for remote users. Among these experiments are the Remote Analog ASIC Design (READ) system and the Remote Electronic Laboratory (REL). Both are based in LabVIEW on the server and client sides. Both solutions are focused in delivering real experiments to users. The REL system is based on the control of real lab equipments (Oscilloscope, function generator, power supply and multimeter) controlled via GPIB (General Purpose Interface Bus). Therefore, a virtual instrument (VI) developed in LabVIEW communicates with these devices according to some client’s commands. A user interface designed for each lab equipment can be called by the user to set up signals, read responses and measurement results.

This work although is focused on our READ system. This system also provides remote users access to real experiments. But it relies on a data acquisition card to generate signals and read responses.

To control the data acquisition and signal generation on the server, the NI DAQmx driver software was used, which can be called from LabView to program NI measurement devices. The driver software has an application programming interface (API), which is a library of VIs, functions, classes, attributes, and properties for creating applications with data acquisition.

A. *The Circuit Under Test*

The ispPAC10 is member of a family of analog circuits capable of realizing a variety of popular analog functions including precision filtering, summing/differencing, gain/attenuation and integration.

The ispPAC10 operates within a voltage range that goes from 1 up to 4V. Because all performed experiments were carried out in single ended mode, all the inverter analog inputs were connected to VREFOUT (2.5V).

Its basic building block contains four integrated programmable analog modules known as PACblocks and a programmable analog routing pool. Each PACblock emulates a collection of op amps, resistors and capacitors. The ispPAC10’s flexibility permits the user to vary a filter’s gain, corner frequency, and other characteristics without the need for external components. The ability to program the internal capacitor values of the ispPAC10 allows the designer to realize thousands of distinct analog circuits and filter characteristics from a given circuit architecture.

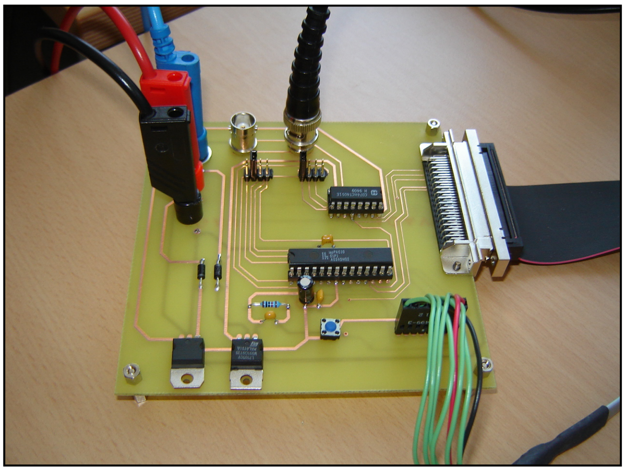


Figure 2. PCB with the ASIC

The assembled board also contains all necessary circuitry, like voltage regulators and an analog multiplexer to facilitate the selection of inputs to apply signals. It is also possible to make measurements with external devices once jumpers allow the connection of inputs and outputs to coaxial connectors.

The application software used to program and therefore create the desired circuit within the ispPAC10 is the PAC-Designer software, also from Lattice Semiconductor. For PC clients, this application software is available via our Citrix Presentation server; therefore users can change circuit schematics as wished. For mobile clients, although, this feature obviously is not available and we assume that predefined analog functions should be already uploaded to the device.

B. *The Server and Client*

As mentioned previously, the communication between server and client is based on direct TCP/IP programming with the API for TCP/IP communication from LabVIEW.

As the server is started, it keeps listening to a specified port (TCP port 80). When it detects a connection from a client, it first reads from the socket the parameters for a signal generation and acquisition. The server calls then a sub VI that handles the signal 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. As the response is sent back, the client converts the data and displays the data to the user. This process is described by the following simple diagram.

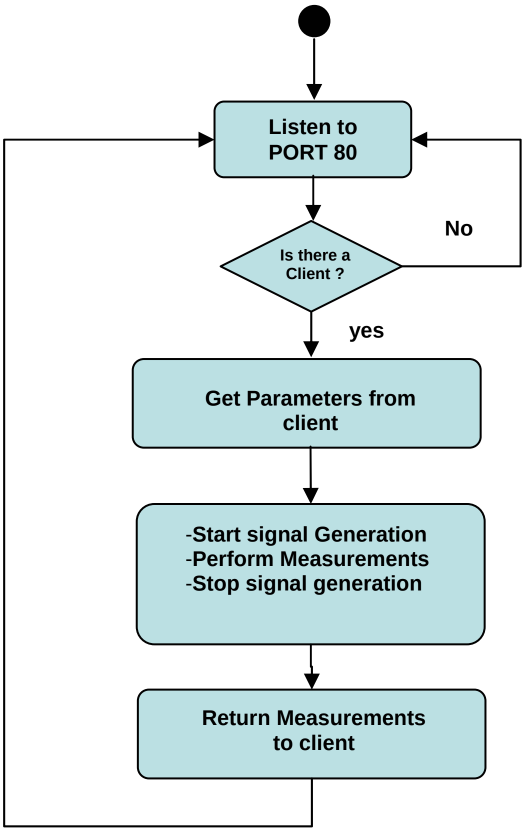


Figure 3. Basic sequence diagram of the server

The server processes client requests by applying the desired signal to the circuit under test and returning the measurements as shows Fig. 3. Clients were designed for PDAs as well as for Windows PCs. Requests from both should be 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.

When transferring data directly via TCP/IP, the data must be previously converted to string types. The length of the string must first be sent (also as a string) and then the data itself. Therefore, the client converts all parameters (frequency, offset, waveform, etc) to string types and transmits them via TCP/IP. On the server, these parameters are read and converted back to their original type to be processed by the VIs controlling the data acquisition.

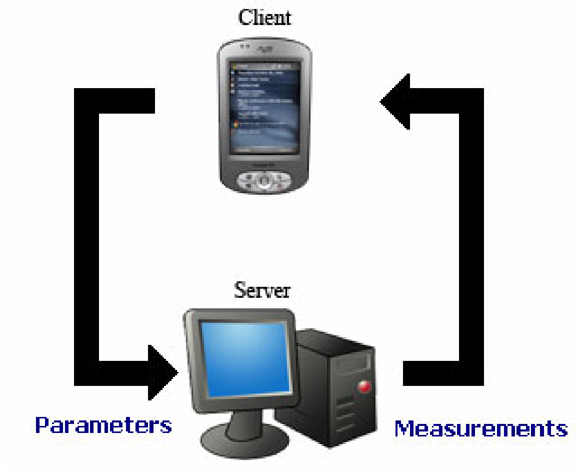


Figure 4. Client/Server Communication

The responses are then converted to string and sent back to the client. On the client these responses are converted back to their original types and displayed to the user.

III. CONCLUSIONS

The most important consideration when designing a remote laboratory client to be accessed by such devices like PDAs and Smartphones is the resource limitation compared to PCs. Therefore it is necessary to optimize the clients for these limitations which lead to a reduction of the features available for such labs.

Remote mobile solutions can be an attractive tool as they offer many different possibilities for applications in industries and e-learning. Wireless mobile experimenters have a great advantage because they are not subjected to limitations of locomotion, but require also a good network (wireless) infrastructure to perform experiments. For that reason, we understand that each situation must be studied within a scenario prior to implement such a system.

Moreover, remote systems can be very useful when applied to solutions involving high costs of people and equipment transportation. Different institutes and schools could share experiments and therefore knowledge in an additional manner. Also by using remote solutions it would be possible for universities with less financial resources to take advantages of expensive equipments from other institutions by means of a cooperative network of remote systems.

The work presented with this paper shows one possibility for delivering remote control over one specific device (ispPAC10) to clients running on mobile devices.

As described previously, this system makes use of pure TCP/IP messages to send data through the internet. Another way to implement this communication with LabVIEW is using Shared Variables. Shared Variables consist of a way to transmit data between multiple computers using simple coding techniques. They are faster and simpler to implement than similar communication methods.

Shared variables use the Publish-Subscribe Protocol (PSP) which is a form of User Datagram Protocol (UDP).

Sharing these variables is possible due to the Shared Variable Engine (SVE). For stand alone applications based on shared variable, it is necessary to install the SVE separately. For PDA devices it is also necessary to install the support for shared variables [10].

The SVE is where a variable resides on a computer. It manages the network communication and bindings, all of 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 set up interfaces.

Shared variable servers use UDP port 2343 and a range of UDP ports from 6000 to 6010 for incoming and outgoing packets and the 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 [10].

Nevertheless, we justified our approach to use TCP/IP communication because portability and firewall rules are very important issues when developing such a system. We have experienced several problems when attempting to access labs via SVE from different networks, especially networks from institutions that usually are under a strict firewall policy. We cannot make any assumptions about these policies and try therefore to base our system in standard ports (80) for the communication with the server rather than on several different ports used by the shared variable tools.

Finally we would like to point out, that the solutions shown are applicable also to other labs or experiments in terms of an easy to adapt standard solution.

REFERENCES

[1] M. E. Auer, *Virtual Lab versus Remote Lab*. Villach, Austria.

[2] M. E. Auer, W. Gallent, The Remote Electronic Lab as a Part of the Telelearning Concept as the Carinthia Tech Institute (Published work style). Villach, Austria, 2000.

[3] H. Edwards, *Building an Interactive Web Page with DataSocket* (Application note 127). Austin, TX: National Instruments, 1999.

[4] G. Perry, “Visual Basic 6.0 in 21 days (Book style)” Ed. Indianapolis: Sams Publishing, 1998.

[5] Unknown: “LabView Basics I Course Manual”, Austin, TX: National Instruments Corporation, 1998.

[6] Unknown: “Getting Started with Measurement Studio for Visual Basic”, Austin, TX: National Instruments Corporation, 2001.

[7] W. Gallent, “Remote Electronic Lab (Thesis or Dissertation style),” Ms. dissertation, Dept. of Electronics., Carinthia University of Applied Sciences., Villach, Carinthia, 2000.

[8] http://www.w3schools.com

[9] C. Mergl, “Comparison of Remote Labs in Different Technologies (Thesis or Dissertation style),” Ms. dissertation, Dept. of Electronics., Carinthia University of Applied Sciences., Villach, Carinthia, 2006.

[10] URL: http://www.ni.com

AUTHORS

M. E. Auer, is with the Carinthia University of Applied Sciences and Head of the Center of Competence “Online Labs and E-Leaning, Villach, A-9500 Austria (e-mail: m.auer@ieee.org).

D. G. Zutin is with the Carinthia University of Applied Sciences, Villach, Kaernten A-9500 Austria (e-mail: D.GarbiZutin@ fh-kaernten.at).

A. Y. Al-Zoubi is with the Princess Sumaya University for Technology and Dean of Graduate School and Scientific Research , Amman, Jordan (e-mail: zoubi@psut.edu.jo)

This article was modified from a presentation at the IMCL2008 conference in Amman, Jordan, April 2008.

Manuscript received 01 June 2008. Published as submitted by the authors.