

Design and Test of Application-Specific Integrated Circuits by use of Mobile Clients

[doi:10.3991/ijoe.v5i1.779](https://doi.org/10.3991/ijoe.v5i1.779)

Michael E. Auer, Danilo G. Zutin

Carinthia University of Applied Sciences, Villach, Austria

Abstract—The aim of this work is to develop a simultaneous multi user access system – READ (Remote ASIC Design and Test) that allows users to perform test and measurements remotely via clients running on mobile devices as well as on standard PCs. The system also facilitates the remote design of circuits with the PAC-Designer. The system is controlled by LabVIEW and was implemented using a Data Acquisition Card from National Instruments. Such systems are specially suited for manufacturing process monitoring and control. The performance of the simultaneous access was tested under load with a variable number of users. The server implements a queue that processes user’s commands upon request.

Index Terms—Mobile Laboratories, Online Labs, Remote Control of Devices.

I. INTRODUCTION

We have developed a system to design and test Analog ASICs (Application-Specific Integrated Circuits) remotely over the internet. The system allows for users to create different circuits with an application software and upload them to a real device. The device under test used is an ispPAC10 from Lattice Semiconductor.

Remote solutions applied to engineering teaching laboratories have been shown to be attractive tools in enhancing access to practical experiments [1]. 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.

With the release of Visual Studio .NET 2003, the number of applications created for mobile devices grew as 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.

II. DEVELOPMENT OF THE SERVER IN LV

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 [6].

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.

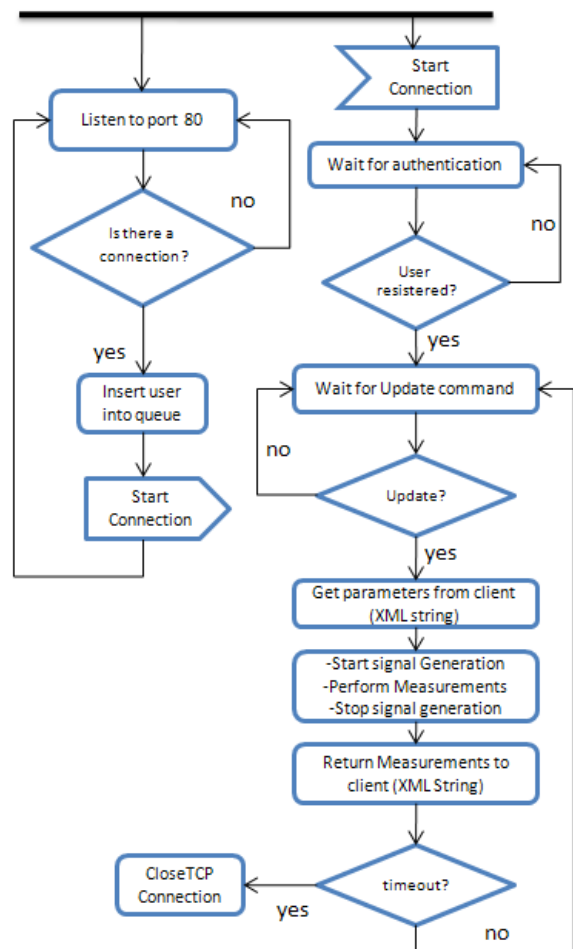


Figure 1. State Diagram of the Server

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.

Our server was developed with LabVIEW versions 8.2.1 and 8.5. LabVIEW offers many proprietary high level communication protocols, like data socket and shared variables, but our approach consisted in developing our own server with the TCP/IP API of LabVIEW, so that we are free to use any platform to develop our clients, as long as they implement a TCP/IP communication and interpret XML documents.

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 an XML string and transmits them via TCP/IP. On the server, this string is read and the parameters are extracted and converted back to their original type to be processed by the VIs controlling the data acquisition. The responses are then converted to an XML string again and sent back to the client. On the client these responses are converted back to their original types and displayed to the user. LabVIEW has its own XML Schema file that allows it to interpret the created XML file. Although we use XML as a vehicle for our data, this is not strictly necessary, but it makes the transferred data more portable and easier to be interpreted by clients developed in other platform.

The diagram shown in Fig. 01 describes the functioning of the server. We decided to use TCP port 80 for the client/server communication. The server listens to TCP port 80 and when there is a connection, it assigns to this connection the last position in the queue. 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 we allow maximum connection duration of 30 min.

The server processes client requests by applying the desired signal to the circuit under test and returning the measurements back to the client. 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.

The user can reconnect afterwards if needed.

As described previously, this system relies on lower level TCP/IP communication 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 [8].

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 [8].

	Shared Variables	TCP/IP
Implementation	Fast	Takes more time to develop applications
Implementation complexity	Easy implementation	More complex
Portability to other client Platforms	Low. Relies on LV on the Client	High, client can be developed in different platforms
Problems with Firewall rules	Several TCP and UDP ports must be opened for incoming and outgoing packets	It is possible to choose which port to use

Figure 2. Comparison of Technologies

Nevertheless, we justified our approach to implement our application in the TCP/IP transport layer 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. Figure 2 describes the main differences and considerations taken into account in order to choose the communication technology.

III. 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.

The assembled board shown on Figure 3 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 because jumpers allow the connection of inputs and outputs to coaxial connectors.

IV. 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.

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). Figure 4 shows a VI running on an emulated device.

Due to the limited graphical capabilities of these devices, the controls and indicators are sized and scaled and the functions palette is reconfigured.

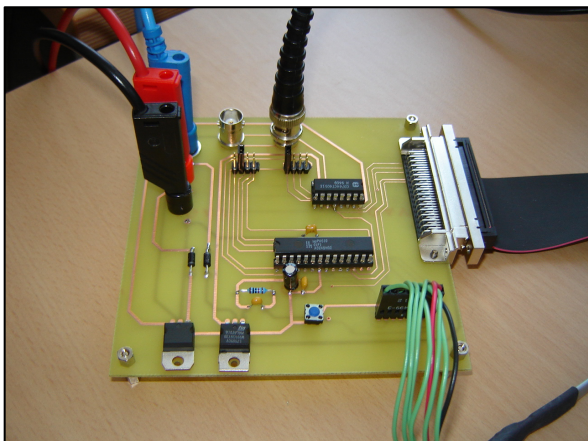


Figure 3. PCB with the ASIC



Figure 4. Client running on a WM5 emulator

V. CONCLUSION

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 the implementation.

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.

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, W. Gallent, The Remote Electronic Lab as a Part of the Telelearning Concept at the Carinthia Tech Institute (Published work style). Villach, Austria, 2000.
- [2] G. Perry, "Visual Basic 6.0 in 21 days (Book style)" Ed. Indianapolis: Sams Publishing, 1998.
- [3] Unknown: "LabView Basics I Course Manual", Austin, TX: National Instruments Corporation, 1998.
- [4] Unknown: "Getting Started with Measurement Studio for Visual Basic", Austin, TX: National Instruments Corporation, 2001.
- [5] W. Gallent, "Remote Electronic Lab (Thesis or Dissertation style)," Ms. dissertation, Dept. of Electronics., Carinthia University of Applied Sciences., Villach, Carinthia, 2000.
- [6] <http://www.w3schools.com>

- [7] 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.
- [8] URL: <http://www.ni.com>
- [9] Revans R (1982) Origins and Growth of Action Learning. Bratt-Institute fur Neues Lernen, Sweden.
- [10] Schmid C (2007) Grid Technologies for Virtual Laboratories in Engineering Education. REV Conference 2007, Porto.
- [11] Tuttas J, Wagner B (2001) Distributed Online Laboratories. Proceedings of the International Conference on Engineering Education. Norway.
- [12] Wankat PC (2004) Analysis of the First Ten Years of the Journal of Engineering Education. Journal of Engineering Education, Vol. 93, No. 1, pp. 13–21.

AUTHORS

Michael E. Auer and **Danilo Garbi Zutin** are with the School of Systems Engineering, Carinthia University of Applied Sciences, Villach, Austria (e-mail: { m.auer | dgzutin }@ieee.org).

This article was modified from a presentation at the Workshop on Remote control of Devices (WoRD 2008) in London, United Kingdom, November 13-16, 2008. Manuscript received December 22, 2008. Published as submitted by the authors.