Networked Instrumentation – a New Educational Tool for Students' Success

I. Târnovan[1], Rodica Holonec[2], C. Iakab[3]

[1]*Technical University of Cluj-Napoca, Str C-tin Daicoviciu Nr.15, +40264401518*
ioan.tarnovan@mas.utcluj.ro
[2]*Technical University of Cluj-Napoca, Str C-tin Daicoviciu Nr.15, +40264401520*
rodica.holonec@mas.utclu.ro
[3]*Technical University of Cluj-Napoca, Str C-tin Daicoviciu Nr.15, +40264401520*
iakab_ciprian@yahoo.com

Abstract – This paper presents the methods implemented in LabVIEW, a powerful graphical programming software that allows users to remotely monitor and control virtual instruments. Remote Panels and DataSocket can be successfully implemented in order to create virtual laboratories that would give students permanent access to useful educational resources. We created an application that remotely monitors and controls a FieldPoint system by using two virtual instruments that share data through a DataSocket server. Apart from the above-mentioned virtual instruments' functionality, the paper also presents the structure of our FieldPoint system and how students can use the VIs to interact with FieldPoint modules.

Keywords – Distance learning, networked instrumentation, virtual laboratory, remote monitoring and control, LabVIEW.

I. Introduction

Traditional development in measurement science mainly consists of increasing the accuracy and speed of measuring methods and instruments. The last decades have changed the traditional metrology by integrating it with information science; this is how virtual instrumentation appeared.
Virtual Instrumentation provides users with countless tools for processing and storing the acquired measurement data. It has also become an integrant part of automation and test systems leading to accurate and reliable control of the applications.
A new request emerged due to the World Wide Web: networked instrumentation, which allows people to interact with their instrumentation system from anywhere at any time. This technology has great impact on teaching methods, as you will see further on in this paper.

II. Networked Instrumentation and Distance Learning

Distance learning is a boosting field driven by the Internet facilities. However, a key component of current distance learning implementations in science and engineering is missing and that is distance laboratory experimentation. Moreover, teachers need to have at their disposal a distance learning opportunity for classroom teaching. In order to avoid the movement of experimental setup towards the classroom and to provide students with a distance learning mean, it is useful to offer interactive and remote access to laboratory facilities.
The combination between virtual instrumentation and internet results in a new tool – networked instrumentation or internet-enabled instrumentation [1]. The applications available when using this tool are the following:

- **Remote monitoring** – a process can be observed from another location on the network. Execution can be monitored through a client while the process is running on a server. The client cannot give feedback or provide inputs to the server process.
- **Remote control –** in addition to the capabilities of the remote monitoring system, in this case the remote user can send data, messages or inputs back to the server process that will affect the output.
- **Collaboration** – several clients communicate and share information with the server process and, at the same time, with each other as part of the communication.

Implementing the features of networked instrumentation in order to build a virtual laboratory would have a great impact on students' learning progress:

- unrestricted practice time with access from anywhere in the world;

- students do not need to purchase additional software (a web browser is needed only);
- laboratory instruments are used by several students at the same time.

The ideal candidate to provide the high level of interactivity and interoperability required for such a virtual laboratory is LabVIEW [2]. Beside its tools for virtual instrument design, which can interact directly with a physical process, it also features easy-to-use elements that allow remote monitoring and controlling: Remote Panels and DataSocket.

A. Remote Panels

LabVIEW was used in all previous successes of remote laboratories but extensive programming of Java, CGI or other third-party software tools was required to bring local laboratory functionality to a browser environment [3]. Now with LabVIEW Remote Panels, remote execution is just a couple of clicks away. Without any additional programming a LabVIEW program can be enabled for remote control through a common Web browser (see Figure 1). Having this new technology the user simply points the Web browser to the Web page associated with the application. Then the user interface for the application shows up in the Web browser and is fully accessible to the remote user.

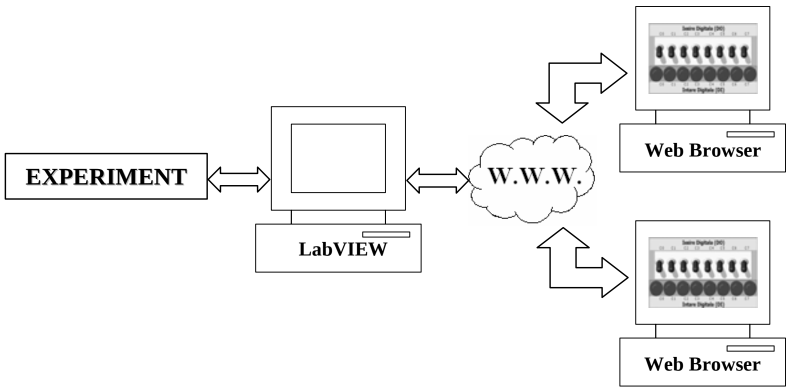


Figure 1. Internet Control of a Remote Laboratory

Acquisition is still occurring on the host computer but the remote user has total control and identical application functionality. In contrast with previous versions of LabVIEW which allow only one user to interact with an application, LabVIEW 8.20 introduces simultaneous control of remote applications. Consequently, using LabVIEW 8.20 more students can retain simultaneous control of the same application remotely. Several students can request control of the same VI from the host and LabVIEW will create an independent instance of the application for each student to control and monitor. This means that when a student operates a change on the Front Panel, no matter of the change, it will not affect the other students' instances of the application. This feature can be used when several students need access and control at the same time. This is of great importance for example when you have one computer of data files and a VI that runs analysis on the data files. If the student wishes to operate the VI while someone else is using it, there is no need to wait for the other to relinquish control before he can use the application to run his analysis.

B. DataSocket

DataSocket is a programming tool that enables you to read, write and share data between applications and/or different data sources and targets. DataSocket provides a unified API (application program interface) for low-level communication protocols like: HTTP, FTP and TCP/IP [1]. You can connect to different data sources without having to write different codes to support different data formats and protocols as DataSocket converts data for transfer and passes the actual values to your application. Use DataSocket when you want to share live data, regardless of how you display, store or obtain it.
DataSocket Server and Manager are included in the LabVIEW package so there is no need to purchase them separately, like in the case of Remote Panels. However, using DataSocket requires students to have on their stations the same LabVIEW version that is installed on the host machine on which runs the acquisition application. DataSocket can be used both for distance learning and for indoor laboratory experiments.

Imagine a lab with 30 student workstations, a lab server and a computer that acquires measurements and performs analysis of the measured data. The teacher needs to acquire and distribute data to networked student workstations. The teacher decides to connect the student workstations to the server so that the acquisition machine maintains only one connection to the server rather than to all 30 student workstations (see Figure 2). The students develop virtual instruments containing DataSocket reader components that connect to the server and read data.

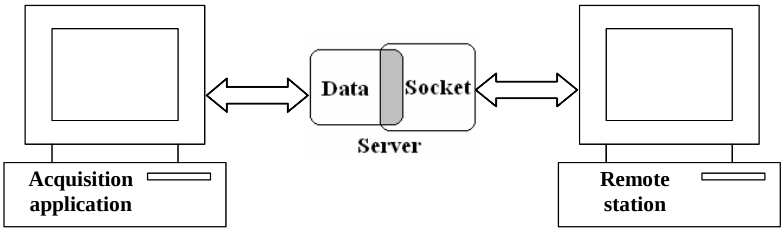


Figure 2. Using DataSocket for remotely access and control an application

The teacher can restart the acquisition or reboot the acquisition computer without having to reconnect all 30 student workstations. The student workstations stay connected to the server and receive new measurements as soon as the acquisition restarts and sends data to the server.

III. Remote controlling and monitoring of a FieldPoint system

The application that we developed aims at presenting students how DataSocket and LabVIEW can be used in order to remotely monitor and control a process. The application involves a DataSocket server, a publisher and a client. The roles of the publisher and the client are played by two virtual instruments (VIs) at the same time; both of them read/write data to the DataSocket server using the LabVIEW DataSocket Read/Write functions. The remote VI was built into an executable application so that there is no need for LabVIEW to be installed on students' machine.
Furthermore, the application utilizes a FieldPoint system so that students will be acquainted with its features. They will understand how they can use FieldPoint modules in order to interact with a real process: the modules' ranges, the functions implemented in LabVIEW that communicate with the FieldPoint modules. Using the executable VI the student will be able to control the FieldPoint system remotely.

A. The FieldPoint system

A FieldPoint system is a modular device with I/O functions that can be used for measurement and automation applications that require a large number of sensors [4]. We used four FieldPoint modules (Figure 3) having eight channels each:

- the analog input module FP-AI-110 can measure voltages and currents and the range for each channel can be set using the FieldPoint Explorer software utility;
- the analog output module FP-AO-200 can generate current within the ranges 0-20 mA and 4-20 mA;
- the digital input module FP-DI-330 is compatible with voltages up to 120/240 VAC/VDC;
- the digital output module FP-DO-400 whose channels can output a voltage equal to the module voltage supply;

Figure 3. FieldPoint system: network module FP-1000; analog modules: FP- AI-110, FP-AO-200; digital modules: FP-DO-400, FP-DI-330

These four modules connect to the host computer through a FP-1000 network module, using an RS-232 cable. The circuit we made in order to point out the modules functions is presented in Figure 4.

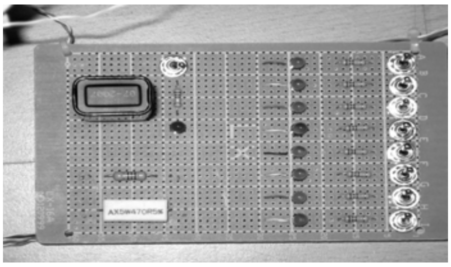


Figure 4. The circuit which interacts with the FieldPoint system

The eight channels of the digital output module will turn on/off the row of LEDs; if the switches on the right side on Figure 4 are on, then the voltage applied to the LEDs will also be applied to the channels of the digital input module. Further on, the current generated by channel 0 of the analog output module is fed to channel 0 of the digital input module, while the channel 1 of the analog input module measures the voltage of a battery. Channels 2 and 3 of the analog input module measure the voltages on two resistors (left-down side of Figure 4) at the variation of the currents generated by channels 1 and 2 of the analog output module.

B. The DataSocket server

The DataSocket server runs on the same machine that communicates with the FieldPoint system. We used the DataSocket Read and Write functions and the DataSocket Transport Protocol (dstp) when developing the VIs. The general format for this protocol is: *dstp://ServerName//Item*, where *ServerName* is the machine's IP where the DataSocket server is running, and *Item* is a tag for the data published on the server.

C. The Local Virtual Instrument

This VI interacts with the FieldPoint system by using the FP Write and FP Read functions and publishes the measured information to the server. The values it reads from the server, which are dictated by the remote VI, are written to the FieldPoint modules output.
In the first subdiagram of the Flat Sequence Structure used in the Block Diagram of the VI we create the references to the FieldPoint's channels we use – see Figure 5. Then, in the second subdiagram, we read/write data from/to the FieldPoint and to the DataSocket server. The VI does not allow local control as this is done only remotely.

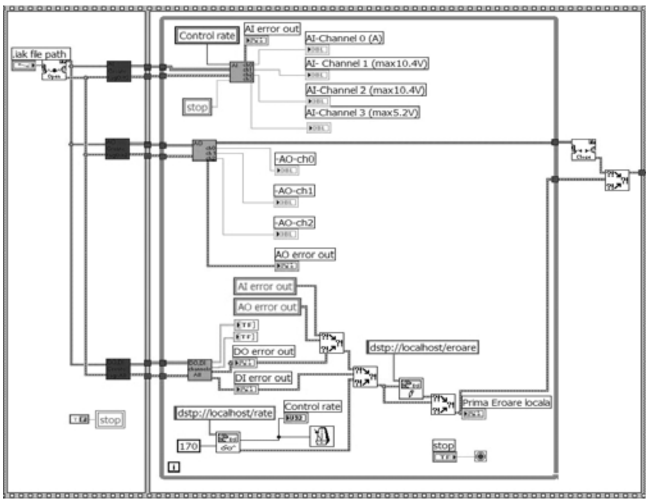
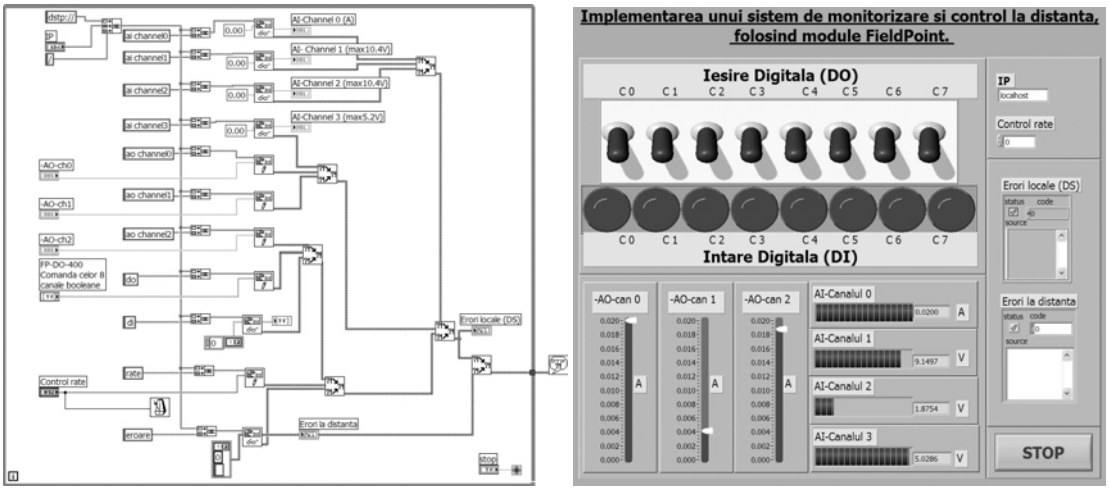


Figure 5. Block Diagram of the Local Virtual Instrument

D. The Remote Virtual Instrument

This VI monitors the inputs and controls the outputs of the FieldPoint modules through the DataSocket server and the local VI. In the IP string control we specify the IP address of the local machine where the server is located, which shares the data between the two VIs. Beside the outputs of the FieldPoint modules, the rate at which the acquisition is done on the local machine can also be controlled. The Block Diagram (a) and Front Panel (b) of the VI are as shown in Figure 6.



(a) (b)

Figure 6. Remote Virtual Instrument. (a) Block Diagram. (b) Front Panel.

This VI was built into an executable application so that there is no need for LabVIEW to be installed on the remote machine.

We must mention the fact that students can write their own virtual instrument that will interact with the DataSocket server. They obviously need to know the items' names where data is published on the server.

IV. Conclusions

The world of networked instrumentation is beginning to boost, especially in applications for distance-based education and remote labs. Using LabVIEW as platform of choice is one of the easiest, fastest ways to get your instrumentation system on the Internet.

Generally students only need a web browser in order to interact with an application. But they can also develop their own virtual instrument that will interact through a DataSocket server with the application. In this case they will be able to analyze data according to their own needs.

The example we carried out using the DataSocket server and the functions provided by LabVIEW aim to show how easy it is to monitor and control a process remotely. We consider this example as a starting point for the implementation of a virtual laboratory that would give our students the opportunity to have unlimited access to laboratory resources.

References

[1]Travis J., Internet Applications in LabVIEW, Prentice-Hall, 2000.

[2]Gustavsson I., "Laboratory Experiments in Distance Learning", International Conference on Engineering Education, Oslo, 2001.

[3]Chiculita C., Frangu L., "A Web Based Remote Control Laboratory", SCI Conference, 2002, Orlando, Florida, USA.

[4]Hedeşiu H., "Familia FieldPoint", Măsurări şi automatizări, nr.1, 2002.

[5]Gillet D., Salzmann Christophe, Gorrochategui E., "Remote Manipulation with LabVIEW for Educational Purposes", Computer Based Measurement and Automation in Education, National Instruments Conference Proceedings 1998, Austin, Texas, USA.

[6]Luncă, E., Brânzilă, M.: Aplicații server-client în LabVIEW. DataSocket şi VI Server, Revista de Instrumentație Virtuală, nr.2, 2005.

[7]Bitter, R.: LabVIEW advanced programming techniques, 2000.