Remote Monitoring and Control of Irradiation Experiments

O. Toader, V.H. Rotberg and G.S. Was

Michigan Ion Beam Laboratory
Department of Nuclear Engineering and Radiological Sciences
The University of Michigan, Ann Arbor, MI 48109, USA

Abstract. As computer technology plays an increasing important role in particle accelerator facilities, instrumentation systems can be expected to include web connections and other remote capability features. The Michigan Ion Beam Laboratory at the University of Michigan in Ann Arbor has developed remote monitor and control capability by using a combination of commercial software packages and in-house software development. Irradiation parameters such as ion current on the samples and apertures, sample temperature read from an optical pyrometer, and chamber pressure can all be accessed and monitored remotely through a web site, as can ion source parameters such as power supply currents and voltages or feed gas pressure. With administrator permission, the control parameters of the ion source or the readouts from the irradiation stage can be modified in real time during an experiment. A description will be given of the various ways in which this type of remote monitoring and control has been accomplished at the Michigan Ion Beam Laboratory.

INTRODUCTION

The Michigan Ion Beam Laboratory (MIBL) is located in Ann Arbor and is a part of the Department of Nuclear Engineering and Radiological Sciences at the University of Michigan. Much of the recent work done in the laboratory is related to the radiation damage experiments in metallic alloys, which can require significant radiation time (from a few hours to several days). Having the capability to control and observe the experiment from a remote location has several benefits, including the ability to track an experiment and to assist in solving a problem that might arise. Alerts to a pager or E-mail messages to laboratory staff when a problem occurs would minimize troubleshooting time and reduce down time. A significant fraction of the work is done in collaboration with researchers from different organizations or institutions necessitating a means to track the progress of the experiments remotely. The best way to remotely monitor an experiment is via the World Wide Web, with a vehicle that would be platform independent and use the software already installed on the computers in MIBL. The different approaches that were tried and implemented will be presented in this paper.

LABORATORY DESCRIPTION

The laboratory is equipped with a 1.7 MV Tandetron accelerator, a 200 KV ion implanter and an ion beam assisted deposition system (IBAD). Remote monitoring could be implemented only on the Tandetron and the IBAD system as the implanter does not have advanced computer control.

The accelerator is a solid-state gas insulated, high frequency device, capable of operation between 0.2 and 1.7 MV. Only the Torvis-type source is computer controlled, but the accelerator can be fed also by a duoplasmatron and a sputtering source. A beam of protons of up to 300 μA can be produced with the Torvis source that can be used for radiation damage experiments. The Tandetron has two beamlines, a 15^0 beamline for ion beam modification (implantation, mixing) and radiation damage, and a 30^0 beamline for ion beam analysis, each terminated with a target chamber. Both beamlines contain a quadrupole magnet for focusing, an analyzing magnet, a raster-scanner and a steerer. The 15^0 beamline and the chamber are equipped with cryopumps that can routinely achieve pressures in the 10^{-9} Torr range. Beyond the main chamber on the 15^0 beamline is an electrically isolated irradiation sub-chamber to which is attached a temperature controlled and heated sample stage for radiation damage experiments between 50^0 and 600^0 C. The 30^0 beamline contains an aperture system, a Faraday cup for charge collection, a beam viewer, a translation two-axis goniometer and detectors for backscattering and glancing angle measurements. It is turbo-pumped and equipped for rapid sample turn-around. Rutherford backscattering spectroscopy (RBS), nuclear reaction analysis (NRA), elastic recoil detection (ERD), and ion channeling are conducted in this chamber. The components that are interfaced and controlled/monitored by computer are the Torvis source and the end-station data-collection system. The control/monitor software programs are written in Labview from National Instruments (NI) and were developed at MIBL

The IBAD system consists of two 6 kV electron beam guns and a low energy Kaufman ion gun in a large cryopumped vacuum chamber with a base pressure in the order of 10^{-10} Torr range. The Kaufman ion gun can produce inert gas beam currents up to 100 mA at energies between 10 and 1200 eV. The controller of the ion gun is a power supply model MPS 3000, with a GPIB and serial ports interfaced with a computer and with a system of two thickness monitors. In the IBAD there are also a flow controller GFC 1000, a Varian pressure monitor and a residual gas analyzer, all with a serial interface. The software running on the computer to monitor and control the deposition experiments is written in Labview with the code developed at MIBL. Remote monitoring for this system is done in a manner that is similar to that for the Tandetron and will not be presented here.

COMPUTER HARDWARE AND SOFTWARE.

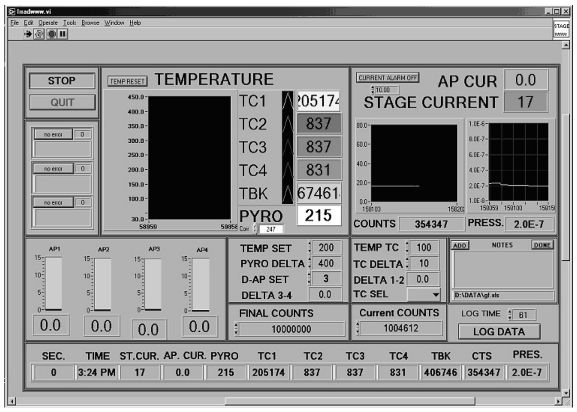
MIBL is equipped with Pentium 4, 500 MHz computers. The computer controlling the Torvis source is interfaced with the source through a serial link and a set of analog and digital modules. The computer monitoring the stage parameters has a PCI temperature board, a PCI analog/digital board and a scope card. The computer controlling the IBAD instruments has an analog/digital board and a GPIB card. Anyone interested in developing Web monitoring would pursue one of the many available avenues according to available software, budget and preferences. A few options to connect remotely were considered and some weak and some strong points of each were identified. Labview is a powerful software package that can be used to design user interfaces to interactively control the systems in a graphical environment taking advantage of many programming tools by creating virtual instruments that mirror real ones and allowing remote users to collaborate in real time. With Labview one can connect to other applications and share data through ActiveX, the Web, DLLs, shared libraries, TCP/IP etc. The second package we are using is AppletView (Java based), available from Nacimiento Inc. AppletView assists in publishing the LabView instrumentation on the web and makes the web browsers capable of connecting to the applications running on a local computer. The built-in server mediates the communication between the local and the remote computer, as the instruments' outputs are made available to a remote client. This is accomplished by streaming the data from the local instruments to a Java applet running in a Web browser. The software enables the programmer to configure and shape the user interface according to the needs.

IMPLEMENTATION

1. Labview with Data Sockets

To remotely monitor experiments, a program was written to make the connection between the Labview data acquisition (DAQ) program and the Web interface using data sockets. The data socket (DS) is a unified end-user application programming interface (API) based on URLs for connecting to DAQ systems located anywhere, and is language independent and operating system (OS) independent. Using this protocol data can be sent via the Web between different machines running Labview. The DS approach consists of a data socket server (DSS) and an API. The advantages of this route are the reliability of the program, the nice Web interface (all the Labview graphical user interfaces (GUI) configuration tools available) and the speed of the data update. The disadvantage is that in order to be able to view the experiment the web user has to download and install the Run-Time Engine version of Labview (freeware from NI) plus a small Labview executable program written at MIBL, or download the entire application as a self-installing program from the MIBL site.

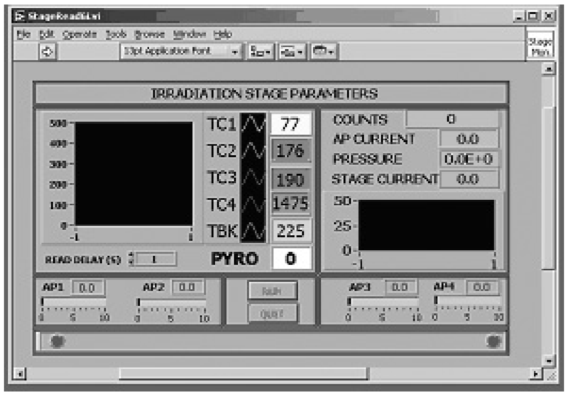
The size of the download of the whole application is rather large, making the process a little difficult for slow computer users. In addition the program needs to be recompiled on a Macintosh machine to have it available for Macintosh users. This code was implemented for the Tandetron but no programs were developed to remotely control and monitor the source and the IBAD system this way. This option would be ideal for a lab with very fast computers and experiments requiring the processing of a very large amount of data. It would not be a good option for a network of slower computers or with no access to Labview software running on a Macintosh computer. The MIBL computer terminal interface is presented in fig. 1, and the remote panel available to any web user is presented in fig. 2.



TC1 205174 · TC2 837 · TC3 837 · TC4 831 · TBK 67461 · PYRO 215
AP CUR 0.0 · STAGE CURRENT 17 · COUNTS 354347 · PRESS. 2.0E-7
TEMP SET 200 · PYRO DELTA 400 · D-AP SET 3 · DELTA 3-4 0.0
TEMP TC 100 · TC DELTA 10 · DELTA 1-2 0.0 · TC SEL
FINAL COUNTS 10000000 · Current COUNTS 1004612 · LOG TIME 61
D:\DATA\gl.xls

SEC.	TIME	ST.CUR.	AP. CUR.	PYRO	TC1	TC2	TC3	TC4	TBK	CTS	PRES.
0	3:24 PM	17	0.0	215	205174	837	837	831	406746	354347	2.0E-7

FIGURE 1. GUI in MIBL



TC1 77 · TC2 176 · TC3 190 · TC4 1475 · TBK 225 · PYRO 0
COUNTS 0 · AP CURRENT 0.0 · PRESSURE 0.0E+0 · STAGE CURRENT 0.0
READ DELAY (S) 1
AP1 0.0 · AP2 0.0 · AP3 0.0 · AP4 0.0

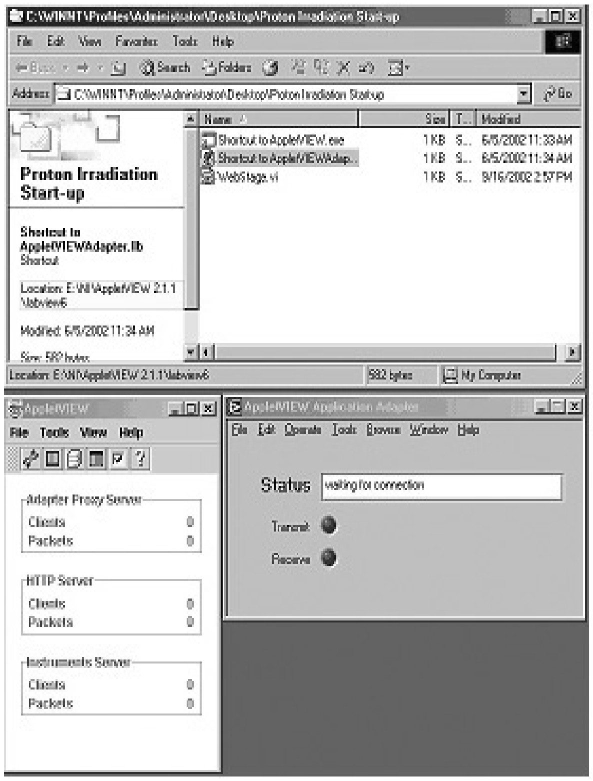
FIGURE 2. GUI for data sockets remote users

2. Labview with the Built-in Web-Server

Web monitoring with the Labview Web Server was implemented after Labview 6.0 was released. This version allows any application to be published as a web file with a click of a button. The next step required to have remote monitoring would be to start the Labview Web Server and that is simply done with a menu selection making the application available on-line. The Labview Web Server approach has very nice security and user monitoring features, but also some disadvantages. Like the previous method, it would require the user to download and install the Labview Run-Time Engine from the National Instruments site or form the MIBL site before the first run. The Labview Web Server comes with a built-in one-client capability, limiting the number of users. Additional licenses can be purchased, but cost can be an issue with this strategy. Having to buy more licenses can put a strain on a tight budget. Also, this alternative is not available for Macintosh computer users. It is a good option if a large number of licenses can be bought and there are no anticipated Macintosh users. Another nice aspect is the simplicity of the approach and the requirement of very little extra programming besides the main application.

3. Labview and AppletView.

This software combination is the one selected for use at MIBL. In order to make the connection to the Web, it is necessary to build an application with Labview that would collect the data points of interest from the main DAQ program. The data objects in this program are allowed to be only of a certain type (string, integer32, single and Boolean type). Then a Labview Server built with AppletView libraries would broker the data transfer between LabView DAQ program, the newly built Labview application and the built-in AppletView server running in the background. The required program selection, the AppletView server interface and the Labview server interface are presented in fig. 3. In fig. 4 can be seen the MIBL control panel for the local computer.



C:\WINNT\Profiles\Administrator\Desktop\Proton Irradiation Start-up

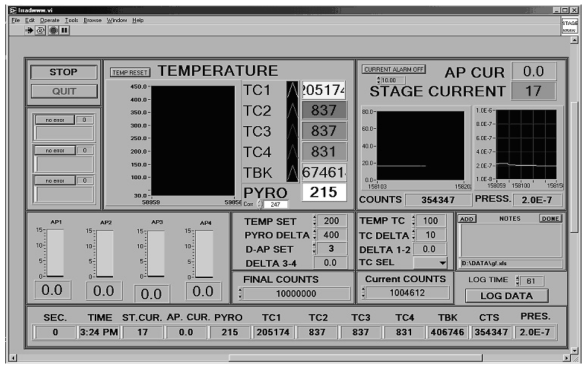
Name	Size	Modified
Shortcut to AppletVIEW.exe	1 KB	6/5/2002 11:33 AM
Shortcut to AppletVIEWAdap...	1 KB	6/5/2002 11:34 AM
WebStage.vi	1 KB	9/16/2002 2:57 PM

Shortcut to AppletVIEWAdapter.llb
Shortcut
Location: E:\NI\AppletVIEW 2.1.1\labview6
Modified: 6/5/2002 11:34 AM
Size: 582 bytes

Adapter Proxy Server — Clients 0, Packets 0
HTTP Server — Clients 0, Packets 0
Instruments Server — Clients 0, Packets 0

Status: waiting for connection

FIGURE 3. AppletView menu server.



TEMPERATURE

TC1	205174
TC2	837
TC3	837
TC4	831
TBK	67461
PYRO	215

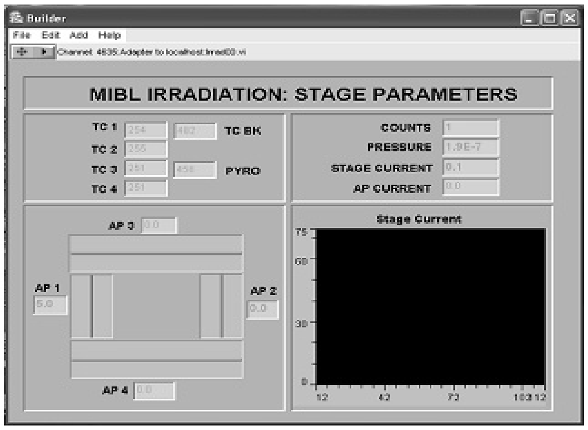
AP CUR	0.0
STAGE CURRENT	17
COUNTS	354347
PRESS.	2.0E-7
AP1	0.0
AP2	0.0
AP3	0.0
AP4	0.0
TEMP SET	200
PYRO DELTA	400
D-AP SET	3
DELTA 3-4	0.0
TEMP TC	100
TC DELTA	10
DELTA 1-2	0.0
TC SEL	
FINAL COUNTS	10000000
Current COUNTS	1004612
LOG TIME	61

D:\DATA\gf.xls

SEC.	TIME	ST.CUR.	AP. CUR.	PYRO	TC1	TC2	TC3	TC4	TBK	CTS	PRES.
0	3:24 PM	17	0.0	215	205174	837	837	831	406746	354347	2.0E-7

FIGURE 4. GUI at MIBL for irradiation monitor

The system is reliable and it can be viewed by anybody with a web browser accessing the MIBL address (http://www-ners.engin.umich.edu/labs/mibl/) Fig. 5 presents the graphical user interface available to remote users.



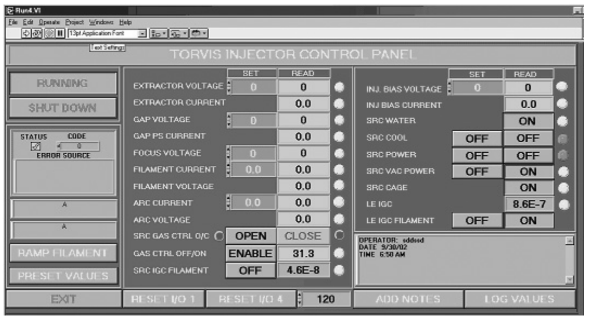
Channel: 4635:Adapter to localhost:Irrad00.vi

MIBL IRRADIATION: STAGE PARAMETERS

TC 1	254
TC BK	482
TC 2	255
TC 3	251
PYRO	456
TC 4	251
COUNTS	1
PRESSURE	1.9E-7
STAGE CURRENT	0.1
AP CURRENT	0.0
AP 3	0.0
AP 1	5.0
AP 2	0.0
AP 4	0.0

FIGURE 5. GUI AppletView stage monitoring

The same procedure is used to monitor/control the Torvis ion source. The local computer control panel of the source is given in fig. 6, with the corresponding web panel in fig. 7.



TORVIS INJECTOR CONTROL PANEL

	SET	READ
EXTRACTOR VOLTAGE	0	0
EXTRACTOR CURRENT		0.0
GAP VOLTAGE	0	0
GAP PS CURRENT		0.0
FOCUS VOLTAGE	0	0
FILAMENT CURRENT	0.0	0.0
FILAMENT VOLTAGE		0.0
ARC CURRENT	0.0	0.0
ARC VOLTAGE		0.0
SRC GAS CTRL O/C	OPEN	CLOSE
GAS CTRL OFF/ON	ENABLE	31.3
SRC IGC FILAMENT	OFF	4.6E-8

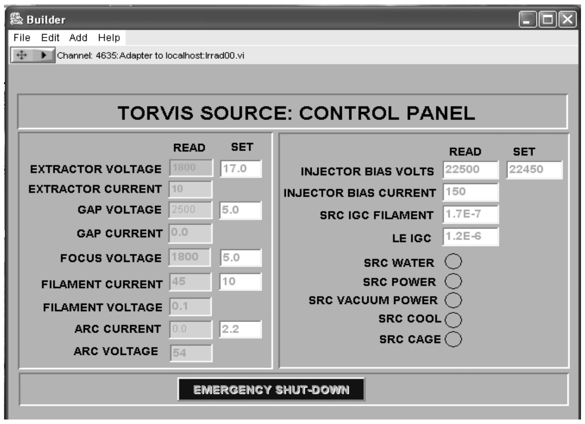
	SET	READ
INJ. BIAS VOLTAGE	0	0
INJ BIAS CURRENT		0.0
SRC WATER		ON
SRC COOL	OFF	OFF
SRC POWER	OFF	OFF
SRC VAC POWER	OFF	ON
SRC CAGE		ON
LE IGC		8.6E-7
LE IGC FILAMENT	OFF	ON

STATUS CODE 0
ERROR SOURCE

OPERATOR: sddsd
DATE 9/30/02
TIME 6:50 AM

120

FIGURE 6. MIBL GUI source control panel



Channel: 4635:Adapter to localhost:Irrad00.vi

TORVIS SOURCE: CONTROL PANEL

	READ	SET
EXTRACTOR VOLTAGE	1800	17.0
EXTRACTOR CURRENT	10	
GAP VOLTAGE	2500	5.0
GAP CURRENT	0.0	
FOCUS VOLTAGE	1800	5.0
FILAMENT CURRENT	45	10
FILAMENT VOLTAGE	0.1	
ARC CURRENT	0.0	2.2
ARC VOLTAGE	54	

	READ	SET
INJECTOR BIAS VOLTS	22500	22450
INJECTOR BIAS CURRENT	150	
SRC IGC FILAMENT	1.7E-7	
LE IGC	1.2E-6	
SRC WATER	○	
SRC POWER	○	
SRC VACUUM POWER	○	
SRC COOL	○	
SRC CAGE	○	

FIGURE 7 Source AppletView Interface

AppletView is not as user friendly as Labview and requires a type of programming that is similar in many ways to Labview programming incorporating Java features. The interfaces are built by the user with the software tools that are available and in consideration for the data objects that Java can support. This approach to remotely access instruments is convenient as it works on any type of computer with any Web browser, by simply connecting to a web site without the need of extra downloads. The weak point of this combination is the existence of two separate software packages that have to work in complete agreement. Occasional miscommunications can occur, but can easily be fixed by re-starting the AppletView server.

SUMMARY

At the Michigan Ion Beam Laboratory, we were able to find a solution to remotely monitor and control irradiation experiments that allows multiple users to log in simultaneously and access the experiment data and the control panels. A similar approach has been made to implement remote monitoring for the IBAD system. Table 1 gives a summary of the strong and weak points of each avenue for remote monitoring that was tried and implemented at MIBL.

The solution with Labview and AppletView currently in use is relatively simple, not expensive and can be developed using commercially available software packages and in-house programming. As far as future plans are concerned, we would like to improve the remote interface by adding more features, one of which is the direct image view of the irradiation stage. This feature is not yet supported by AppletView, but we are exploring other ways in which this can be accomplished.

Features	Labview & Data Sockets	Labview & Web Server	Labview & Appletview
Difficulty	medium	easy	medium
All platforms	Yes, with restrictions	No	Yes
# of users	unlimited	limited	unlimited
Reliability & security	Good	Very Good	Good
Software to download	Yes	Yes	No

TABLE 1. Web monitoring software with the pros and cons

REFERENCES

1. Nacimiento Software Company, *AppletView user guide and reference manual*, 2002.

2. Jeffrey Travis, *Internet applications with Labview*, 2000.

3. Gary Johnson, *Labview Power Programming,* 1998.

4. Douglas Comer, *Internetworking with TCP/IP* Vol.1, 2000.

5. John Pollock, *JavaScript: a Beginner's Guide*, 2001