Controlling a Data Acquisition System with Java

D. Brummond
Advised by Dr. Christian Stoeckl

Laboratory for Laser Energetics
University of Rochester
250 East River Road
Rochester, NY 14623
August 29, 2008

Abstract

PC-based data acquisition systems have the capability to replace oscilloscopes, signal generators, and certain digital input/output communications in laboratory diagnostics setups. Though proprietary software exists for controlling these systems, this software has no flexibility to be adapted for LLE use. Instead an open-source C library for Linux machines (Linux Control and Measurement Device Interface (COMEDI)) has been used. This library supplies functions for communicating with a PC-based data acquisition system. The goal of this project was to adapt ScopeControl, a preexisting program written in Java for controlling oscilloscopes, to monitor and save data from a PC-based data acquisition system using the COMEDI library. Programs were written in both C and Java and the Java Native Interface was used to allow these programs to communicate. Once ScopeControl had been successfully modified, additional hardware and software features present in the data acquisition systems were explored, including analog waveform generation and remote data monitoring through the network.

Outline

1 Introduction

The purpose of the University of Rochester Laboratory for Laser Energetics (LLE) is to conduct experiments in laser-driven inertial confinement fusion. Experiments are conducted in the Omega and Omega EP laser systems and vast amounts of data are produced. In order to collect this data and monitor lab equipment, many oscilloscopes and signal generators are utilized. Recently an alternative technology is available in PC-based data acquisition systems that would simplify data collection and give laboratory operators a more flexible user interface. However, a program to support PC-based data acquisition systems needed to be developed.

1.1 Current Hardware

Figure 1.1: A standard oscilloscope

For most data collection needs at LLE, oscilloscopes are used (see fig. 1.1). Oscilloscopes are a common piece of laboratory hardware that can connect to an analog channel and collect any signals that are generated. Oscilloscopes have been utilized in laboratories for decades, but the possibility of replacing some with PC-based data acquisition systems is being investigated. Oscilloscopes can have a very high sampling rate and are easily capable of sampling signals on the order of nanoseconds (or smaller), such as those produced during laser shots on Omega or Omega EP. However, for slower signals and monitoring lab equipment, a regular oscilloscope may be more accurate than is necessary. Also, most oscilloscopes are limited to monitoring four analog channels at a time. For specialized tasks this is acceptable, but if many signals need to be monitored at any given time this could be a costly limitation.

1.2 PC-Based Data Acquisition Systems

With the advent of the PC in the laboratory setting, many pieces of specialized laboratory equipment have been replaced by peripheral devices connected to a computer. A PC-based data acquisition system (see fig. 1.2) is a peripheral device that works with a computer to perform tasks formerly done by oscilloscopes or signal

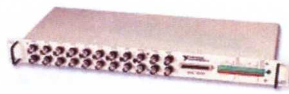

Figure 1.2: A PC-based Data Acquisition System

generators. These data acquisition systems have the capability of monitoring and producing both analog and digital signals. The number of analog channels that can be monitored on a PC-based data acquisition system is generally greater than that of oscilloscopes. By combining a large number of analog channels with the capability to produce both digital and analog signals, PC-based data acquisition systems can greatly simplify tasks normally performed by numerous lab instruments.

1.3 Program Overview

A preexisting program, ScopeControl, was modified so that it could receive analog input from a data acquisition through the Comedi library (see section 2.4). ScopeControl was created to provide a user interface that could control a multitude of various makes and models of oscilloscopes. The Comedi library already supports a large number of various PC-based data acquisition systems. The goal of this project was to give ScopeControl the capability to monitor data from any of the data acquisition systems supported by Comedi, thereby increasing the functionality of ScopeControl. A program was written in the C programming language (gpibni) that provided an interface to the Comedi library for ScopeControl. ScopeControl was then modified to communicate with gpibni through the Java Native Interface.

2 Programming Technologies

Between ScopeControl and the Comedi library, many layers of code exist that must communicate with each other successfully in order for the program to function. To understand how this code works, it is appropriate to have an understanding of the various programming technologies involved.

2.1 The Java Programming Language

The Java Programming Language is a portable, object-oriented programming language. Portability was a necessity of this project, as LLE is host to many computers and a variety of operating systems. ScopeControl was originally designed with this portability in mind, and it assisted in the final incorporation of data acquisition system support. More important for Java though is its object-oriented nature. An object-oriented programming language possesses a hierarchical program structure with abstract and super classes at the highest levels and specific subclasses at the lower levels. Without an object-oriented programming language, this project would have been next to impossible. As ScopeControl holds a Scope super class, only a data acquisition system subclass needed to be created in ScopeControl. This prevented the necessity of a full program overhaul and simplified programming.

2.2 The C Programming Language

The C Programming language is a relatively fast, lower level programming language. It is machine specific, which reduces overhead that exists in Java, but also removes portability. The C programming language was used for this project because it offers the speed necessary for a data acquisition system as well as the ability to communicate with data acquisition systems through the Comedi library.

2.3 The Java Native Interface

The Java Native Interface, or JNI, is a programming framework that allows programs written in Java to call and be called by machine-specific native programs such as those written in the C programming language. For the purposes of this project, the JNI allowed ScopeControl to give commands to C programs for data acquisition, as well as receive the data generated by these programs.

2.4 The Comedi Library

Comedi, or Linux Control and Measurement Device Interface, is a set of device drivers, tools, and libraries that can be used to control certain PC-based data acquisition systems. Comedi supplies the ability for programs written in C to control and obtain data from data acquisition hardware. This removes the tedious task of hardware-level programming for data acquisition and allows focus to remain on higher-level control programs.

3 Program Design

Due to the object-oriented nature of ScopeControl, adding PC-based data acquisition system features required few changes to existing code. A new program had to be written in C that could talk to the Comedi library and certain classes had to be added to ScopeControl. In a few situations, existing ScopeControl code had to be modified, such as the graphical user interface (GUI) and the 16-channel support.

3.1 Class Overview

ScopeControl has an abstract class Scope whose properties are inherited by the individual Scope classes. These classes hold the information on how the program should operate with each individual make and model of scope. A new Scope class was written, ScopePCI6251 (named after the PC-based data acquisition system that it was used on), that would allow ScopeControl to function though the Comedi library. However, it would be impractical for ScopeControl to talk directly to the Comedi library because of the limitations of Java. In order to circumvent this, a C program was written (gpibni) that receives commands from ScopePCI6251 through the Java Native Interface. It then executes any commands given, receives data from Comedi, and processes the data. Once the data is in the form of a byte buffer, it can be sent back through the Java Native Interface to ScopePCI6251. ScopePCI6251 works together with gpibni and the rest of ScopeControl through the abstract class ScopeConnection. A new ScopeConnection class was written that could utilize the Java Native Interface and communicate with both gpibni and ScopePCI6251.

3.2 Program Interface

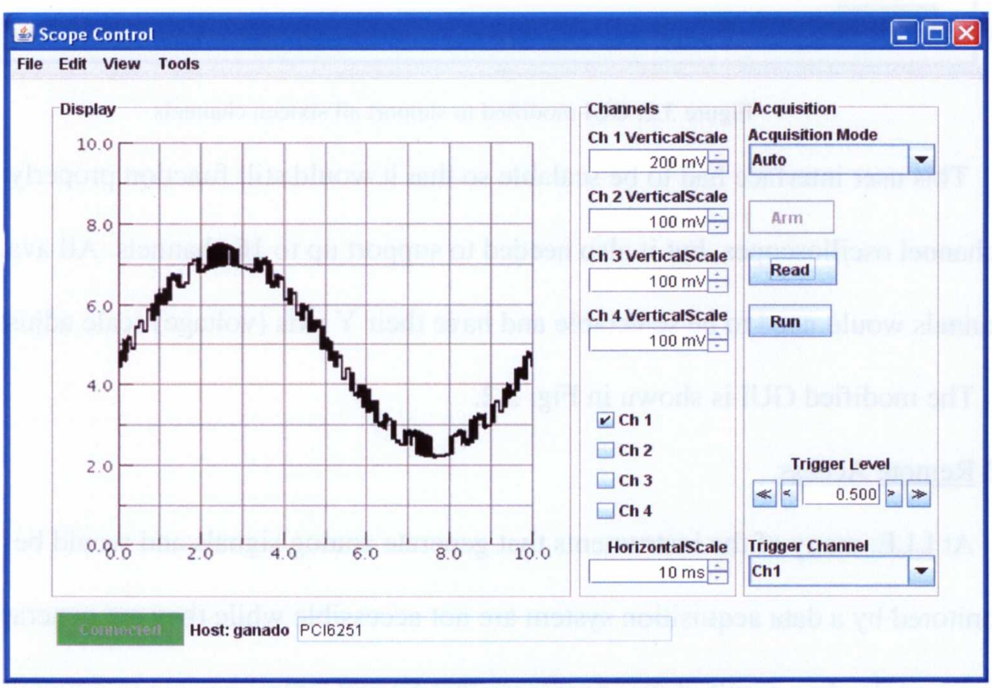

Figure 3.1: GUI with only four-channel support

ScopeControl was originally written to communicate with standard oscilloscopes, not data acquisition systems. As most oscilloscopes are limited to 4 analog input channels, ScopeControl was written to support only 4 channels (see Fig. 3.1). In some cases, modifying ScopeControl was as simple as changing the value of a variable. However, in the case of the GUI, a new user interface had to be created.

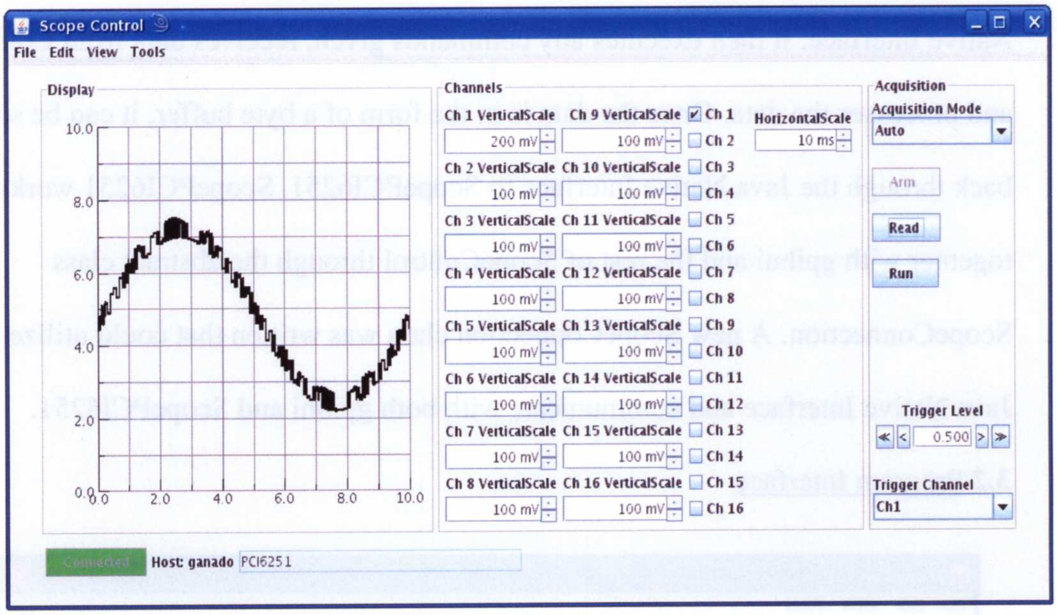

Figure 3.2: GUI modified to support all sixteen channels

This user interface had to be scalable so that it would still function properly with 4-channel oscilloscopes, but it also needed to support up to 16 channels. All available channels would need to be selectable and have their Y axis (voltage) scale adjustable. The modified GUI is shown in Fig. 3.2.

3.3 Remote Access

At LLE, many of the instruments that generate analog signals and would be monitored by a data acquisition system are not accessible while they are generating these signals. As a result, it was necessary that ScopeControl be able to remotely monitor analog signals from a data acquisition system. TEK7000Server is a preexisting program that already has the capability to create a server socket and communicate with a client computer through a network connection. It already had the capability to monitor an oscilloscope, so it was only necessary to add the code to interface to the PC-base data acquisition system . The client computer can then run ScopeControl and connect to the server through a network connection.

3.4 Future Adaptations

The object-oriented nature of ScopeControl made the inclusion of data acquisition system support possible. It also makes possible any number of future upgrades that may be added to ScopeControl. Functionality for more oscilloscopes and data acquisition systems could be added in a similar manner to the Comedi-supported data acquisition systems. Another future adaptation is hardware triggering. Comedi does not currently support hardware triggering for analog input channels. Gpibni was designed so that should this support ever be given to Comedi, Gpibni could be easily modified for hardware triggering. Although analog signal output was not heavily investigated for this project, it is possible for analog output to be added to ScopeControl. However, a separate program may fulfill this role better than a modification of ScopeControl.

4 Program Testing

Throughout the process of program design, it is necessary to test the program often so that any errors will be quickly discovered. ScopeControl now has two main methods of reading an analog signal. These are software-triggered and untriggered data acquisition. Both required extensive testing to ensure proper functionality.

4.1 Untriggered Data Acquisition

Untriggered Data Acquisition refers to simply reading signals from an analog channel and displaying them. When the computer is told to read data under untriggered data acquisition, it immediately begins reading the data. In order to test data acquisition in ScopeControl, a signal generator was connected to the data acquisition system and analog signals were produced. After the signal was displayed by ScopeControl, the analog channel was connected to a regular oscilloscope and the accuracy of ScopeControl was determined. When the signal displayed by ScopeControl was the same as that displayed by the oscilloscope, it was known that ScopeControl was properly reading the signal.

4.2 Software-Triggered Data Acquisition

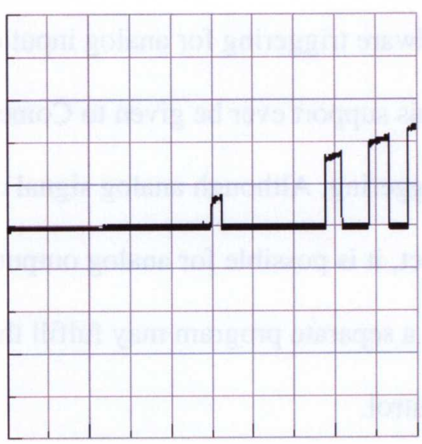

Figure 4.1: The beginning of an analog signal displayed by the data acquisition system

Software-triggered data acquisition is similar to its untriggered counterpart except that the computer now determines when to start recording the analog signal, rather than the user. As Comedi does not currently support hardware-triggered data acquisition, it was necessary that triggering be created in the software. Currently, all triggering is done by gpibni. If ScopeControl tells gpibni that the next data read is to be software triggered, gpibni creates a ring buffer with which to store incoming analog data. A ring buffer stores data points in a virtual ring. As new data comes in, it overwrites old data. When a data point is read that exceeds the trigger value, the ring buffer ceases to overwrite old data points and places any new data in a separate buffer. The two buffers are then combined into a single set of data and sent back to ScopeControl. Software triggering can be done to prevent an incoming signal from "rolling" across the display, or it can be used to catch and view the beginning of a signal (see Fig. 4.1). Because regular oscilloscopes feature triggering, testing the functionality of software-triggered data acquisition was done in the same manner as untriggered data acquisition. The signal displayed by ScopeControl was compared to the signal displayed by the oscilloscopes and any errors were corrected.

5 Conclusion

Through experiments at LLE, vast amounts of data are collected and many laboratory instruments need to be monitored. In order to collect all this data, many oscilloscopes are in use at LLE. However, certain oscilloscopes can now be replaced by newer PC-based data acquisition systems. These data acquisition systems can monitor more channels, provide higher flexibility, and are relatively inexpensive compared to oscilloscopes. In the future, data acquisition systems can continue to replace current lab equipment by acting as signal generators and digital IO in a laboratory environment.

6 Acknowledgements

I would like to thank my advisor, Dr. Christian Stoeckl, for his assistance, help, and guidance on my project. I would also like to thank Benjamin Smith for helping me to understand the Comedi library and ScopeControl. Finally, I would like to thank

Dr. Stephen Craxton for giving me the opportunity to participate in the 2008 University of Rochester Laboratory for Laser Energetics Summer Program.