Summary

For upper level undergraduate or graduate level Mechanical Engineering, Chemistry, Physics, Electrical Engineering, Electrical Technology, or Computer Science courses where data acquisition and analysis are required. It is also essential as a reference guide to using LabVIEW for data acquisition and analysis for professionals in these fields working in industrial, scientific, academic, or laboratory environments.

This book teaches how to use LabVIEW for data acquisition involving monitoring and/or controlling electrical signals with a computer in a scientific, testing, or manufacturing environment. *LabVIEW for Data Acquisition* is a practical book that demonstrates how LabVIEW is most effective in a variety of real-world data acquisition applications. It is full of hands-on tips, tricks, and techniques.

Features

- **Real-world programming techniques unique to each data acquisition application.**
  - Demonstrates how LabVIEW can be customized to perform as an effective tool under a wide range of practical situations.

- **CD-ROM.**
  - Contains the evaluation version of LabVIEW 6i for Windows and software examples that are in the book.

- **Graphics and illustrations.**
  - Facilitates comprehension of these topics.
Table of Contents

Preface.
1. Learning LabVIEW for the First Time.


2. Signals and DAQ.

3. Basic DAQ Programming Using LabVIEW.
Assign Device Numbers to DAQ Devices. LabVIEW DAQ Functions: An Overview. Using the Basic LabVIEW DAQ Functions to Display Data. LabVIEW File I/O with DAQ.

4. Simulation Techniques.

5. DAQ Debugging Techniques.

7. Transducers.

8. Non-NI Hardware Alternatives.
   Non-NI DAQ Hardware Companies. Homemade DAQ Hardware.


10. DAQ at a Distance-Networked and Distributed Systems.

11. Alternate Software for DAQ.
    LabWindows/CVI. Microsoft Visual Basic with DAQ (ComponentWorks). Microsoft Visual C++ with DAQ (ComponentWorks++). Other Programming Languages with DAQ.

12. Finalizing Your LabVIEW Software.
    Finalizing Your LabVIEW Software on Your Development Computer. Moving Your LabVIEW Software to Other Computers.

Appendix A: Fundamentals: Bits, Bytes, Files, and Data.
Appendix B: Top Ten DAQ Problems and Their Solutions.
Appendix C: Saving LabVIEW's Vis.
Appendix D: Example Applications.
Appendix E: LabVIEW/DAQ Tips and Tricks.
Index.
About the Author.
About the CD.

Preface

I've been interested in gadgets and computers since grade school, particularly gadgets connected to computers. My first real gadget-to-computer project, at age 16, was an alarm for my car made from an Atari computer with 16 Kbytes of RAM and a photoresistor. I caught no car thieves, but I did catch a cat trying to steal a nap on my car! Six years later, I was more than a little happy to get my first job out of college at NI (National Instruments), where I worked for years as a LabVIEW developer, actually getting paid to connect gadgets to computers! I quickly learned that data acquisition is the professional term for connecting certain types of gadgets to computers, like the photoresistor in my first car alarm.

If you already have an NI data acquisition board with analog input connected to a computer with LabVIEW 6i, and you are dying to watch
it work, skip right ahead to start on Chapter 3, up to and including Section 3.1.5, then come back here.

**LabVIEW and Data Acquisition**

This book is written for people who intend to use National Instruments' LabVIEW (Laboratory Virtual Instrument Engineering Workstation) for data acquisition. LabVIEW has become very popular as the programming language of choice in the context of industrial, scientific, academic, and laboratory environments. LabVIEW is a graphical programming language in which you build the programs with pictures, not words. Data acquisition involves connecting computers to a wide variety of gadgets via electronic signals; the computers then control these gadgets or read data from these gadgets. The term DAQ will be used throughout the book instead of data acquisition.

In general, any place that "scientific measurements" must be taken is an appropriate place for LabVIEW. Following are a few examples of LabVIEW applications:

- Measuring pressure, temperature, and vibration in an airplane during the course of its flight
- Monitoring the pH of a chemical solution during processing
- Analyzing sound waves in an acoustics laboratory
- Monitoring and recording flow rates of liquids or gasses

To use LabVIEW for any of these four example applications, you need the following four components:

1. LabVIEW
2. A computer on which LabVIEW can run
3. A data acquisition device (changes an electrical signal into something the computer can read)
4. A transducer (changes a wide variety of real-world phenomena, like pressure, temperature, pH, sound, etc., into an electrical signal for the data acquisition device)

National Instruments currently makes LabVIEW and a wide variety of data acquisition devices. If you are going to use LabVIEW, I recommend buying your data acquisition device from National Instruments, as it generally simplifies its integration with LabVIEW.

Because LabVIEW is a graphical programming language, it is often quicker to develop than using a text-based language, and its programs are often much more robust.

LabVIEW not only runs on all Microsoft operating systems starting with the word Windows, but it runs on Apple Macintosh O/S, Sun Microsystems's Solaris, and certain Hewlett-Packard workstations as well.

**Organization**
If you have never written a program before in any computer language, you will likely find this book difficult to follow—if this is the case, consider starting with the book *LabVIEW for Everyone*, described in Chapter 1, Section 1.1. Nobody learns how to program for the first time, in any language, without spending much time—usually more time than they expect. Be encouraged to know that LabVIEW, like Microsoft Visual Basic, is one of the least painful languages to learn.

Data acquisition is fundamental to many LabVIEW applications. For this reason, this book is written not quite as a "for dummies" book, but more from the "LabVIEW newcomer" point of view. Many people using LabVIEW for the first time want to do data acquisition—so if you are completely new to LabVIEW, Chapter 1 is designed to teach you just enough LabVIEW to perform some meaningful data acquisition.

**Chapter 1: Learning LabVIEW for the First Time.** This chapter is designed to teach a LabVIEW newcomer just enough to perform useful data acquisition with LabVIEW. It is meant to be the quickest LabVIEW tutorial ever, but as a result, it's a bit like taking a drink from a fire hose. This chapter is not explicitly DAQ-specific, but it subtly focuses on DAQ-related issues.

**Chapter 2: Signals and DAQ.** Learn or review the fundamentals of data acquisition that you will find relevant with LabVIEW—or with any programming language.

**Chapter 3: Basic DAQ Programming Using LabVIEW.** Combine LabVIEW with data acquisition at a very fundamental level. Use real hardware and real wires, and manipulate real signals in this chapter.

**Chapter 4: Simulation Techniques.** Most of your developing can be done without real hardware, right in the comfort of your home or office.

**Chapter 5: DAQ Debugging Techniques.** This chapter tells you how to track down bugs (with a focus on data acquisition programming), should you ever make a programming mistake.

**Chapter 6: Real-World DAQ Programming Techniques.** This chapter is a version of Chapter 3. In order to focus on data acquisition issues without being hindered by limitations of your specific device, all hardware is simulated in this chapter. The most common real-world scenarios are covered in detail in this chapter, and these scenarios often require the advanced techniques covered herein.

**Chapter 7: Transducers.** Exactly what device do you need to convert your temperature (or pressure, humidity, etc.) into a signal compatible with your data acquisition device? This chapter points you in the right direction.

**Chapter 8: Non-NI Hardware Alternatives.** Do you already have a data acquisition device that you want to use with LabVIEW? Or do you just want to save some money on your data acquisition device and have time to spare? Read this chapter.
Chapter 9: Real-Time Issues. Suppose you must collect data at a rate of exactly 10 Hz, or 10 times per second—it is unacceptable to wait 0.11 seconds between data samples. Such issues are not at all obvious, but they are covered in this chapter.

Chapter 10: DAQ at a Distance—Networked and Distributed Systems. Suppose your computer must be placed hundreds of feet from the data you're collecting. Or, suppose you have many data sites, widely separated from one another and your computer. This chapter is for you.

Chapter 11: Alternate Software for DAQ. Suppose you want to perform data acquisition, but you want to use some software other than LabVIEW. For example, maybe you're already very familiar with another programming environment—see this chapter.

Chapter 12: Finalizing Your LabVIEW Software. Your LabVIEW software now works perfectly, and you don't want anybody changing it, not one bit! Here's how to accomplish that.

Appendix A: Fundamentals: Bits, Bytes, Files, and Data. If you have little programming experience or less, this appendix covers programming fundamentals that are needed in any programming language. LabVIEW is no exception! It is not enough to know how to draw pretty pictures; you must understand what's going on underneath.

Appendix B: Top Ten DAQ Problems and Their Solutions. You'll find answers to some common problems in this appendix.

Appendix C: Saving LabVIEW's VIs. It is not obvious that you can sometimes cause permanent damage to LabVIEW itself, not just to your application, if you make a common blunder described herein.

Appendix D: Example Applications. Here is a collection of real-world LabVIEW applications I've personally done, presented so that you can be aware of the abilities of which LabVIEW is capable.

Appendix E: LabVIEW/DAQ Tips and Tricks. Review this list at some point while learning LabVIEW—these are time-saving tips and tricks.

Requirements

This book assumes you have version 6i of LabVIEW; you will gain the most from this book if you work through its examples with version 6i. Its most relevant and powerful features will be illustrated in this book. Ideally, you also have one of NI’s multifunction DAQ devices. To work perfectly with this book, your DAQ device should have at least two analog inputs, two digital I/O ports, a counter/timer, and an analog output. However, if your DAQ device doesn't have all of these features, then let's hope you won't need them, so you can just skip the parts of the book that use them.
If you're a programmer, but have never programmed in anything but text-based languages, get ready for a surprise. Figure P-1 shows a "simple" example program that ships with LabVIEW 6i, where the user interface is not shown—just the guts of the program.

DAQ, in the context of this book, means monitoring or controlling physical phenomena with a computer via electrical signals. These electrical signals are defined by their voltage or current levels, and are usually attached to some sort of scientific or industrial equipment by means of transducers that can convert physical values like pressure, temperature, position, flow rate, and so on to electricity (or vice versa). Most types of computers can interface to these transducers by means of DAQ devices. Companies like Agilent and Tektronix make a variety of high-quality, often specialized scientific instruments that perform very accurate and precise measurements, often in a laboratory environment. These instruments are generally connected to the computer via a special interface cable (GPIB, RS-232, etc.), but they can run without the computer connection, unlike the DAQ devices discussed in this book. These instruments will not be covered in this book to any great extent, even though they fit the definition as DAQ. They really are DAQ, in my opinion, but instead, we'll mostly use this book to describe DAQ as NI does—as generic analog and digital signals, because these signals are flexible enough to be used with an extremely wide variety of transducers. The usual example of the DAQ equipment discussed in this book is a device connected to a computer, which is also connected to a terminal block outside of the computer, reading or writing electrical signals connected to whatever you're interested in. Figure P-2 is a generic diagram of a typical DAQ system.

For many DAQ systems, the Signal Conditioning unit shown in Figure P-2 may not be present—that unit might just be a simple connector block with no signal-conditioning electronics. In this case, it would be screw terminals or some sort of similar electrical/mechanical connection.

Going into a bit more detail, Figure P-3 shows more of the possible components of a DAQ system.

Figure P-4 shows what a DAQ system might really look like, using NI's SCXI hardware and built by someone who knows how to wire much more neatly than I.

A DAQ device may be a board inside the computer, or it might be some sort of external box connected to one of the computer's ports, such as the serial port (RS-232), a parallel port, USB, or IEEE 1394 (FireWire).

National Instruments categorizes LabVIEW's functionality into three parts:

1. Acquire
2. Analyze
3. Present

LabVIEW is designed to seamlessly integrate these basic components of data acquisition, and it does a great job, in my opinion. The
acquisition part of LabVIEW makes it very easy to quickly collect data from NI's DAQ devices, as you have discovered if you've already done the first section of Chapter 3 through Section 3.1.5. The analysis libraries available with the full development system have an amazing list of functions involving FFTs, (Fast Fourier Transforms) power spectrums, RMS (Root Mean Square) calculations, filters, curve-fitting, and many other mathematical functions. The presentation part of LabVIEW is perfectly tuned to very quickly present a wide variety data in a meaningful fashion.

One of NI's catch phrases is "The Software Is The Instrument." Or at least it used to be—I haven't heard that one in a while. At any rate, the comparison is that a modern lab can operate with a computer controlling everything, rather than with numerous separate instruments like oscilloscopes and function generators. This approach not only makes it inherently easier to analyze, manipulate, and control your data, since the data source is always connected to the computer by default, but it also makes it easier to upgrade or modify your instrument configuration.

This book is written as if you have a PC running a Windows operating system—not a Macintosh, UNIX platform, or other type of setup. However, if you do not have a PC, most of the information presented in this book will still be useful, provided your platform can run LabVIEW. You will need to mentally substitute the PC's <Ctrl> and <Alt> keys for whatever the corresponding keys are for your computer's LabVIEW interface, and also try to recognize the PC-specific issues that differ on your computer.

Help! I'm Stuck!

You have quite a few options when you get stuck in LabVIEW. Here they are, in the order you should normally try.

1. **Ask a live person who knows LabVIEW.** Ideally, there is somebody you work with, or know, who can answer some of your LabVIEW questions.
2. **Use LabVIEW's Help.** See Chapter 1, Section 1.3 for details on LabVIEW's internal help.
3. **Internet help.** Knowing how often Internet links die, I'm a bit reluctant to put any such links in this book, so I'll only put in a couple that I suspect will be around for a long time. First, try NI's Web site, www.ni.com. You can hunt around for help on the site, looking for links named "support" or "technical support" to begin with. Currently, the most helpful public forums seem to be info-labview, a busy LabVIEW mailing list, and the newsgroup comp.lang.labview; you will need to find these yourself, though, lest I give you broken links.
   Try a link to other links at my own Web site, whose availability is at the mercy of my ISP, at www.LCtechnology.com/lvhelp.htm. I will keep this current with your best Internet LabVIEW help options, at least for as long as I'm alive.
4. **Call NI.** Currently, the number is 1-800-IEEE-488.
5. **Hire a consultant.** NI has the National Instruments Alliance Pro-
gram, which is a group of certifiable, uh, certified consultants like me, who can likely solve your technical problems relating to NI products. These companies range from one-man operations (such as mine at www.LCtechnology.com/consult.htm) to very large companies with better Web sites and hundreds of people.

6. *Give up.* Let's hope it won't come to this!