LabVIEW Graphical Programming

Fourth Edition

Gary W. Johnson

Richard Jennings

McGraw-Hill

**New York Chicago San Francisco Lisbon London
Madrid Mexico City Milan New Delhi San Juan
Seoul Singapore Sydney Toronto**

The McGraw·Hill Companies

McGraw-Hill books are available at special quantity discounts to use as premiums and sales promotions, or for use in corporate training programs. For more information, please write to the Director of Special Sales, Professional Publishing, McGraw-Hill, Two Penn Plaza, New York, NY 10121-2298. Or contact your local bookstore.

LabVIEW Graphical Programming, Fourth Edition

1234567890 DOC DOC 019876

ISBN 0-07-145146-3

Sponsoring Editor
Wendy Rinaldi

Editorial Supervisor
Jody McKenzie

Project Manager
Samik Roy Chowdhury (Sam)

Acquisitions Coordinator
Alexander McDonald

Copy Editor
Patti Scott

Proofreader
Prachi Ghildiyal

Indexer
WordCo Indexing Services, Inc.

Production Supervisor
James Kussow

Composition
International Typesetting and Composition

Illustration
International Typesetting and Composition

Art Director, Cover
Margaret Webster-Shapiro

Contents

Preface

Twenty years have passed since the release of LabVIEW. During this period, it has become the dominant programming language in the world of instrumentation, data acquisition, and control. A product of National Instruments Corporation (Austin, Texas), it is built upon a purely graphical, general-purpose programming language, *G*, with extensive libraries of functions, an integral compiler and debugger, and an application builder for stand-alone applications. The LabVIEW development environment runs on Apple Macintosh computers and IBM PC compatibles with Linux or Microsoft Windows. Programs are portable among the various development platforms. The concept of *virtual instruments* (VIs), pioneered by LabVIEW, permits you to transform a real instrument (such as a voltmeter) into another, software-based instrument (such as a chart recorder), thus increasing the versatility of available hardware. Control panels mimic real panels, right down to the switches and lights. All programming is done via a block diagram, consisting of icons and wires, that is directly compiled to executable code; there is no underlying procedural language or menu-driven system.

Working with research instrumentation, we find LabVIEW indispensable—a flexible, time-saving package without all the frustrating aspects of ordinary programming languages. The one thing LabVIEW had been missing all these years was a useful application-oriented book. The manuals are fine, once you know what you want to accomplish, and the classes offered by National Instruments are highly recommended if you are just starting out. But how do you get past that first blank window? What are the methods for designing an efficient LabVIEW application? What about interface hardware and real-world signal-conditioning problems? In this book, we describe practical problem-solving techniques that aren't in the manual or in the introductory classes—methods you learn only by experience. The principles and techniques discussed in these pages are fundamental to the work of a LabVIEW programmer. This is by no means a rewrite of the manuals or other introductory books, nor is it a substitute for a course in

LabVIEW basics. You are encouraged to consult those sources, as well, in the process of becoming a skilled LabVIEW developer.

This fourth edition is founded on LabVIEW 8, but we've worked closely with National Instruments to ensure its relevance now and through future versions of LabVIEW. Chapter 1, "Roots," starts off with an entertaining history of the development of LabVIEW. New to this edition is coverage of material on National Instruments' certification exams. There are three levels of LabVIEW certification: Certified LabVIEW Associate Developer (CLAD), Certified LabVIEW Developer (CLD), and Certified LabVIEW Architect (CLA). Each exam builds on the knowledge required for the previous exam. We have worked closely with National Instruments to highlight study material in this book for the first two certification exams. Throughout Chapters 2 through 9 you will find **CLAD** or **CLD** icons next to sections covered on the certification exams.

In Chapters 2 through 9, we get down to the principles of programming in G. After a discussion of the principles of dataflow programming, we discuss programming structures, data types, timing, synchronization, and file I/O. Chapter 8, "Building an Application," shows you how to design a LabVIEW application. Here we assume that you are not a formally trained software engineer, but rather a technically skilled person with a job to do (that certainly describes us!). We'll walk through the development of a real application that Gary wrote, starting with selection of hardware, then prototyping, designing, and testing the program. Chapter 9, "Documentation," covers this important but oft-neglected topic. We discuss recommended practice for creating effective documentation as it pertains to the world of LabVIEW. If you know the material in Chapters 2 through 9 you should have no problems with the certification exams. At the end of Chapter 8 we provide three practice exams for the Certified LabVIEW Developer (CLD) Exam. Use the knowledge you gained in Chapters 2 through 9 to complete these practice exams in four hours and you are well on your way to certification.

If you connect your computer to any external instruments, you will want to read Chapters 10 and 11, "Instrument Driver Basics" and "Instrument Driver Development Techniques." We begin with the basics of communications and I/O hardware (GPIB, serial, and VXI), then cover recommended driver development techniques and programming practices, especially the virtual instrument standard architecture (VISA) methods. Instrument drivers can be fairly challenging to write. Since it's one of our specialties, we hope to pass along a few tricks.

The basics of interface hardware, signal conditioning, and analog/digital conversion are discussed in Chapter 12, "Inputs and Outputs," and Chapter 13, "Sampling Signals." Notably, these chapters contain no LabVIEW programming examples whatsoever. The reason is simple: more than half the "LabVIEW" questions that coworkers ask us turn out to be hardware- and signal-related. Information in this chapter is vital and will be useful no matter what software you may use for measurement and control. Chapter 14, "Writing a Data Acquisition Program," contains a practical view of data acquisition (DAQ) applications. Some topics may seem at first to be presented backward—but for good reasons. The first topic is data analysis. Why not talk about sampling rates and throughput first? Because the only reason for doing data acquisition is to collect data for analysis. If you are out of touch with the data analysis needs, you will probably write the wrong data acquisition program. Other topics in this chapter are sampling speed, throughput optimization, and configuration management. We finish with some of the real applications that you can use right out of the box.

LabVIEW RT brings the ease of graphical programming to the arcane world of real-time system programming. In Chapter 15, "LabVIEW RT," we show you how LabVIEW RT works and how to achieve top performance by paying attention to code optimization, scheduling, and communications.

When software-timed real-time applications won't fit the bill, LabVIEW FPGA is the way to go. LabVIEW FPGA applications are not constrained by processor or operating system overhead. With LabVIEW FPGA you can write massively parallel hardware-timed digital control applications with closed loop rates in the tens of megahertz. Chapter 16, "LabVIEW FPGA," gives a solid introduction to programming FPGAs with LabVIEW.

Embedded computer systems are all around us—in our cars, VCRs, appliances, test equipment, and a thousand other applications. But until now, LabVIEW has not been a viable development system for those miniaturized computers. Chapter 17, "LabVIEW Embedded," introduces a new version of LabVIEW capable of targeting any 32-bit microprocessor.

Chapter 18, "Process Control Applications," covers industrial control and all types of measurement and control situations. We'll look at human-machine interfaces, sequential and continuous control, trending, alarm handling, and interfacing to industrial controllers, particularly programmable logic controllers (PLCs).We frequently mention a very useful add-on toolkit that you install on top of LabVIEW, called the Datalogging and Supervisory Control Module (formerly available as BridgeVIEW), which adds many important features for industrial automation.

LabVIEW has a large following in physics research, so we wrote Chapter 19, "Physics Applications." Particular situations and solutions in this chapter are electromagnetic field and plasma diagnostics, measuring fast pulses with transient recorders, and handling very large data sets. This last topic, in particular, is of interest to almost all users because it discusses techniques for optimizing memory usage. (There are tidbits like this all through the book—by all means, read it cover to cover!)

Chapter 20, "Data Visualization, Imaging, and Sound," shows off some of the data presentation capabilities of LabVIEW. Some third-party products and toolkits (such as IMAQ for imaging) are featured. They enable you to acquire video signals, process and display images, make three-dimensional plots, and record and play sound.

As far as possible, this book is platform-independent, as is LabVIEW itself. Occasional topics arise where functionality is available on only one or two of the computer platforms. The LabVIEW user manual contains a portability guide that you can consult when developing applications that you intend to propagate among various platforms.

Many important resources are available only via the Internet. For your convenience, Internet addresses are interspersed in the text. While writing this book, we found that user-supplied example VIs were hard to obtain, owing to the fact that so many of us work for government laboratories and places that just don't like to give away their software. Where it was not possible to obtain the actual code, we attempted to reconstruct the important aspects of real applications to give you an idea of how you might solve similar problems.

Third-party LabVIEW products, such as driver and analysis packages, are described where appropriate. They satisfy important niche requirements in the user community at reasonable cost, thus expanding the wide applicability of LabVIEW.

If nothing else, we hope that our enthusiasm for LabVIEW rubs off on you.

Gary W. Johnson and Richard Jennings

Acknowledgments

We would like to thank the engineers, developers, and managers at National Instruments who supplied vital information without which this book would not be possible, particularly Jeff Kodosky, David Gardner, Joel Sumner, Newton Petersen, P. J. Tanzillo, and Kathy Brown. A special thanks goes to Zaki Chasmawala of National Instruments for proofreading and highlighting the pertinent parts of Chapters 2 through 9 covered by the certification exams.

Credit also goes to our wives, Katharine Decker Johnson and Patty Jennings, whose patience during this project cannot be overstated. And to Elizabeth, Danny, Chris, and David Jennings—thank you for understanding.

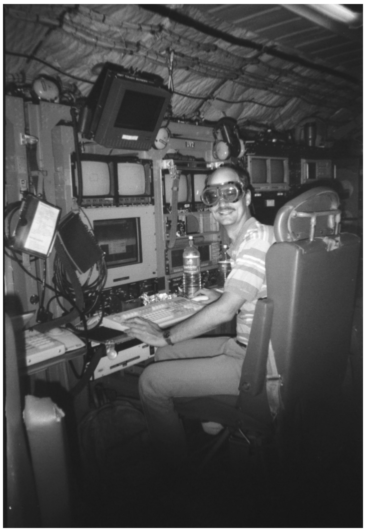
Finally, thanks to our editor, Wendy Rinaldi.

Gary W. Johnson:
To my wife, Katharine

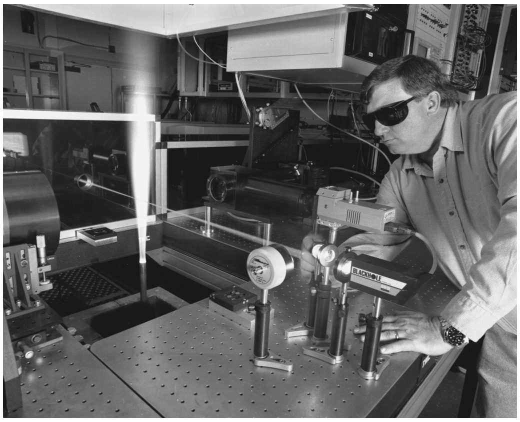
Richard Jennings:
To my Lord and Savior, Jesus Christ

ABOUT THE AUTHORS

Gary W. Johnson is an instrumentation engineer at the Lawrence Livermore National Laboratory. He has a BS degree in electrical engineering/bioengineering from the University of Illinois. His professional interests include measurement and control systems, electro-optics, communications, transducers, circuit design, and technical writing. In his spare time, he enjoys woodworking, bicycling, and amateur radio. He and his wife, Katharine, a scientific illustrator, live in Livermore, California, with their twin Afghan hounds, Chloe and Derby.



LabVIEW goes aloft. Gary works on a LabVIEW-based laser wavelength controller for an airborne LIDAR system aboard an Air Force C-135.



Do not look into laser with remaining good eye. Richard wisely wears his laser eye protection while manually aligning laser beams through a piloted jet burner in the Turbulent Combustion Laboratory.

Richard Jennings is president of Jennings Embedded Services, LLC in San Antonio, Texas. His company was founded in 2005 to serve as a hardware and software resource for companies engaged in embedded software and hardware development in emerging embedded markets such as industrial control, wireless, and embedded instrumentation. He is a 15-year veteran hardware and software engineer. Prior to starting Jennings Embedded Services, a National Instruments Certified Alliance partner, Jennings worked as a system integrator at Sandia National Laboratories and at Lawrence Livermore National Laboratories in Livermore, California. He holds an Associate Degree in Laser-ElectroOptics from Texas State Technical Institute in Waco, Texas, and is a Certified LabVIEW Developer. In 2003 he was awarded National Instruments' prestigious Virtual Instrumentation Professional (VIP) award. *www.jembedded.com*