Learning with LabVIEW™ 2009
Learning with LabVIEW™ 2009

Robert H. Bishop

The University of Texas at Austin
To my parents, W. Robert Bishop
and Anna Maria DiPietro Bishop
CONTENTS

Preface xvii

1 LabVIEW Basics 1
  1.1 System Configuration Requirements 2
  1.2 Installing the LabVIEW Student Edition 2
  1.3 The LabVIEW Environment 3
  1.4 The Getting Started Screen 5
  1.5 Panel and Diagram Windows 8
    1.5.1 Front Panel Toolbar 10
    1.5.2 Block Diagram Toolbar 12
  1.6 Shortcut Menus 14
  1.7 Pull-Down Menus 16
    1.7.1 File Menu 16
    1.7.2 Edit Menu 16
    1.7.3 View Menu 17
    1.7.4 Project Menu 19
    1.7.5 Operate Menu 19
    1.7.6 Tools Menu 20
    1.7.7 Window Menu 22
    1.7.8 Help Menu 22
  1.8 Palettes 23
    1.8.1 Tools Palette 23
    1.8.2 Controls Palette 24
    1.8.3 Functions Palette 26
    1.8.4 Searching the Palettes and Quick Drop 27
  1.9 Opening, Loading, and Saving VIs 30
Contents

1.10 LabVIEW Help Options 33
   1.10.1 Context Help Window 33
   1.10.2 LabVIEW Help 34

1.11 Building Blocks: Pulse Width Modulation 35

1.12 Relaxed Reading: Controlling the World’s Largest Particle Accelerator 37

1.13 Summary 39
   Exercises 41
   Problems 45
   Design Problems 48

2 Virtual Instruments 52

2.1 What Are Virtual Instruments? 53

2.2 Several Worked Examples 55

2.3 The Front Panel 62
   2.3.1 Numeric Controls and Indicators 62
   2.3.2 Boolean Controls and Indicators 64
   2.3.3 Configuring Controls and Indicators 68

2.4 The Block Diagram 68
   2.4.1 VIs and Express VIs 69
   2.4.2 Nodes 70
   2.4.3 Terminals 71
   2.4.4 Wiring 74

2.5 Building Your First VI 77

2.6 Data Flow Programming 82

2.7 Building a VI Using Express VIs 84

2.8 Building Blocks: Pulse Width Modulation 91

2.9 Relaxed Reading: Autonomous Driving in the DARPA Urban Challenge 93

2.10 Summary 96
   Exercises 97
   Problems 102
   Design Problems 106

3 Editing and Debugging Virtual Instruments 111

3.1 Editing Techniques 112
   3.1.1 Creating Controls and Indicators on the Block Diagram 112
   3.1.2 Selecting Objects 114
Contents

3.1.3 Moving Objects 116
3.1.4 Deleting and Duplicating Objects 117
3.1.5 Resizing Objects 117
3.1.6 Labeling Objects 119
3.1.7 Changing Font, Style, and Size of Text 120
3.1.8 Selecting and Deleting Wires 122
3.1.9 Wire Stretching and Broken Wires 123
3.1.10 Aligning, Distributing, and Resizing Objects 128
3.1.11 Coloring Objects 130
3.1.12 Cleaning Up the Block Diagram 137
3.1.13 Routing Wires 139
3.1.14 Reusing Snippets of Code 140

3.2 Debugging Techniques 141
3.2.1 Finding Errors 142
3.2.2 Highlight Execution 144
3.2.3 Single-Stepping Through a VI and Its SubVIs 146
3.2.4 Breakpoints and Probes 147
3.2.5 Navigation Window 152

3.3 Property Nodes 153

3.4 A Few Shortcuts 155

3.5 Building Blocks: Pulse Width Modulation 156

3.6 Relaxed Reading: Using Graphical System Design for Tumor Treatments 158

3.7 Summary 160

Exercises 162
Problems 164
Design Problems 168

4 SubVIs 172

4.1 What Is a SubVI? 173

4.2 Review of the Basics 174

4.3 Editing the Icon and Connector 177
  4.3.1 Icons 177
  4.3.2 Connectors 180
  4.3.3 Selecting and Modifying Terminal Patterns 182
  4.3.4 Assigning Terminals to Controls and Indicators 183

4.4 The Help Window 184

4.5 Using a VI as a SubVI 188

4.6 Creating a SubVI from a Selection 191
## Contents

4.7  Error Checking and Error Handling  193  
   4.7.1  Automatic Error Handling  193  
   4.7.2  Manual Error Handling  194  
   4.7.3  Error Clusters  194  
4.8  Saving Your SubVI  195  
4.9  The VI Hierarchy Window  196  
4.10 Building Blocks: Pulse Width Modulation  198  
4.11  Relaxed Reading: Embedded Graphical System Design Empowers Life-Saving Spider Robots  201  
4.12  Summary  203  
     Exercises  204  
     Problems  207  
     Design Problems  210  

## 5 Structures  213  

5.1  The For Loop  214  
   5.1.1  Numeric Conversion  216  
   5.1.2  For Loops with Conditional Terminals  220  
5.2  The While Loop  221  
5.3  Shift Registers and Feedback Nodes  227  
   5.3.1  Shift Registers  227  
   5.3.2  Using Shift Registers to Remember Data Values from Previous Loop Iterations  229  
   5.3.3  Initializing Shift Registers  231  
   5.3.4  Feedback Nodes  233  
5.4  Case Structures  236  
   5.4.1  Adding and Deleting Cases  239  
   5.4.2  Wiring Inputs and Outputs  241  
5.5  Flat Sequence Structures  247  
   5.5.1  Evaluate and Control Timing in a Sequence Structure  248  
   5.5.2  Avoid the Overuse of Sequence Structures  250  
5.6  The Formula Node  250  
   5.6.1  Formula Node Input and Output Variables  251  
   5.6.2  Formula Statements  251  
5.7  Diagram Disable Structures  254  
5.8  Local Variables  255  
   5.8.1  Creating Local Variables  255  
   5.8.2  Use Local Variables with Care  257  

---

5  Structures  

5.1  The For Loop  214  
   5.1.1  Numeric Conversion  216  
   5.1.2  For Loops with Conditional Terminals  220  
5.2  The While Loop  221  
5.3  Shift Registers and Feedback Nodes  227  
   5.3.1  Shift Registers  227  
   5.3.2  Using Shift Registers to Remember Data Values from Previous Loop Iterations  229  
   5.3.3  Initializing Shift Registers  231  
   5.3.4  Feedback Nodes  233  
5.4  Case Structures  236  
   5.4.1  Adding and Deleting Cases  239  
   5.4.2  Wiring Inputs and Outputs  241  
5.5  Flat Sequence Structures  247  
   5.5.1  Evaluate and Control Timing in a Sequence Structure  248  
   5.5.2  Avoid the Overuse of Sequence Structures  250  
5.6  The Formula Node  250  
   5.6.1  Formula Node Input and Output Variables  251  
   5.6.2  Formula Statements  251  
5.7  Diagram Disable Structures  254  
5.8  Local Variables  255  
   5.8.1  Creating Local Variables  255  
   5.8.2  Use Local Variables with Care  257
Contents

5.8.3 Initializing Local Variables 257
5.8.4 Memory and Execution-Speed Considerations 258

5.9 Common Programming Techniques 258
5.9.1 Sequential Programming 258
5.9.2 State Programming and State Machines 260
5.9.3 Parallelism 264

5.10 Some Common Problems in Wiring Structures 265
5.10.1 Failing to Wire a Tunnel in All Cases of a Case Structure 265
5.10.2 Overlapping Tunnels 266
5.10.3 Wiring Underneath Rather Than through a Structure 266

5.11 Building Blocks: Pulse Width Modulation 267

5.12 Relaxed Reading: Refining the Process of Steel Recycling 271

5.13 Summary 272
Exercises 275
Problems 279
Design Problems 284

6 Arrays and Clusters 288

6.1 Arrays 289
6.1.1 Creating Array Controls and Indicators 290
6.1.2 Multidimensional Arrays 292

6.2 Creating Arrays with Loops 293
6.2.1 Creating Two-Dimensional Arrays 296

6.3 Array Functions 297
6.3.1 Array Size 297
6.3.2 Initialize Array 298
6.3.3 Build Array 300
6.3.4 Array Subset 301
6.3.5 Index Array 302

6.4 Polymorphism 310

6.5 Clusters 313

6.6 Creating Cluster Controls and Indicators 314
6.6.1 Cluster Order 316
6.6.2 Using Clusters to Pass Data to and from SubVIs 318

6.7 Cluster Functions 319
6.7.1 The Bundle Function 319
6.7.2 The Unbundle Function 323
Contents

6.7.3 Creating Cluster Constants on the Block Diagram 323
6.7.4 Using Polymorphism with Clusters 325

6.8 Matrix Data Type and Matrix Functions 326
6.8.1 Creating Matrix Controls, Indicators, and Constants 326
6.8.2 Matrix Functions 328

6.9 VI Memory Usage 330

6.10 Building Blocks: Pulse Width Modulation 331

6.11 Relaxed Reading: User Friendly and Intelligent Acupuncture 334

6.12 Summary 336

Exercises 338
Problems 341
Design Problems 344

7 Charts and Graphs 348

7.1 Waveform Charts 349
7.2 Waveform Graphs 357
7.3 XY Graphs 365

7.4 Customizing Charts and Graphs 368
7.4.1 Axes Scaling 368
7.4.2 The Plot Legend 372
7.4.3 The Graph Palette and Scale Legend 374
7.4.4 Special Chart Customization Features 376
7.4.5 Special Graph Customization Features: Cursor Legend 377
7.4.6 Using Graph Annotations 380
7.4.7 Exporting Images of Graphs, Charts, and Tables 381
7.4.8 Using Context Help 381

7.5 Using Math Plots for 2D and 3D Graphs 382
7.5.1 2D Graphs 382
7.5.2 3D Graphs 386

7.6 Building Blocks: Pulse Width Modulation 390

7.7 Relaxed Reading: Environmental Monitoring in the Costa Rican Rain Forest 391

7.8 Summary 394

Exercises 396
Problems 397
Design Problems 399
Contents

8 Data Acquisition 402
  8.1 Components of a DAQ System 403
  8.2 Types of Signals 404
    8.2.1 Digital Signals 406
    8.2.2 Analog DC Signals 407
    8.2.3 Analog AC Signals 408
    8.2.4 Analog Frequency-Domain Signals 409
    8.2.5 One Signal—Five Measurement Perspectives 411
  8.3 Common Transducers and Signal Conditioning 412
  8.4 Signal Grounding and Measurements 416
    8.4.1 Signal Source Reference Configuration 416
    8.4.2 Measurement System 417
  8.5 Analog to Digital Conversion Considerations 422
  8.6 DAQ VI Organization 427
  8.7 Choosing Your Data Acquisition Device 428
    8.7.1 M Series Data Acquisition Devices 429
    8.7.2 Low Cost Data Acquisition for Students 429
    8.7.3 Simulated Data Acquisition 429
    8.7.4 Macintosh, Linux, Palm OS, Windows Mobile for Pocket PC, and Select Windows CE OS Devices 430
  8.8 DAQ Hardware Configuration 431
    8.8.1 Windows 431
    8.8.2 Channels and Tasks 439
  8.9 Using the DAQ Assistant 441
    8.9.1 DAQmx Task Name Constant 445
  8.10 Analog Input 448
    8.10.1 Task Timing 448
    8.10.2 Task Triggering 449
  8.11 Analog Output 453
    8.11.1 Task Timing 454
    8.11.2 Task Triggering 455
  8.12 Digital Input and Output 460
  8.13 Building Blocks: Pulse Width Modulation 466
    8.13.1 Generating Pulse Width Modulated Signals with Hardware Counters 467
    8.13.2 Applications of Pulse Width Modulation 468
  8.14 Relaxed Reading: Structural Health Monitoring of the Olympic Venues 469
## Contents

8.15 Summary 471  
*Exercises* 474  
*Problems* 475  
*Design Problems* 476

### 9 Strings and File I/O 478

#### 9.1 Strings 479  
9.1.1 Converting Numeric Values to Strings with Build Text Express VI 486

#### 9.2 File I/O 487  
9.2.1 Writing Data to a File 491  
9.2.2 Reading Data from a File 494  
9.2.3 Manipulating Spreadsheet Files 495  
9.2.4 File I/O Express VIs 498  
9.2.5 Obtaining the Path to the System Directories 503

#### 9.3 Building Blocks: Pulse Width Modulation 504

#### 9.4 Relaxed Reading: Optimizing Professional Cyclist Performance 506

#### 9.5 Summary 508  
*Exercises* 509  
*Problems* 511  
*Design Problems* 512

### 10 MathScript RT Module 516

#### 10.1 What Is MathScript RT Module? 517

#### 10.2 Accessing the MathScript Interactive Window 518  
10.2.1 The Command History and Output Windows 520  
10.2.2 Viewing Data in a Variety of Formats 521

#### 10.3 MathScript Help 524

#### 10.4 Syntax 526  
10.4.1 Key MathScript Functions 535

#### 10.5 Defining Functions and Creating Scripts 536  
10.5.1 User-Defined Functions 537  
10.5.2 Scripts 540

#### 10.6 Saving, Loading, and Exporting Data Files 543  
10.6.1 Saving and Loading Data Files 543  
10.6.2 Exporting Data 546
Contents

10.7 MathScript Nodes 547
  10.7.1 Accessing the MathScript Node 548
  10.7.2 Entering Scripts into the MathScript Node 550
  10.7.3 Input and Output Variables 550
  10.7.4 Script Highlighting 554
  10.7.5 Debugging Scripts 556
  10.7.6 Saving Scripts from within the MathScript Node 561

10.8 Applications of MathScript RT Module 563
  10.8.1 Instrument Your Algorithms 563
  10.8.2 Graphical Signal Processing, Analysis, and Mathematics 564
  10.8.3 Integrating Measurement Hardware 565

10.9 Building Blocks: Pulse Width Modulation 566

10.10 Relaxed Reading: Acquiring and Analyzing the Bioacoustic Communication of Killer Whales 568

10.11 Summary 570

Exercises 572
Problems 573
Design Problems 574

11 Analysis 576

11.1 Linear Algebra 577
  11.1.1 Review of Matrices 577
  11.1.2 Systems of Algebraic Equations 581
  11.1.3 Linear System VIs 584

11.2 Statistics and Curve Fitting 587
  11.2.1 Curve Fits Based on Least Squares Methods 587
  11.2.2 Fitting a Curve to Data with Normal Distributions 591
  11.2.3 The Curve Fitting Express VI 593

11.3 Differential Equations 597

11.4 Finding Zeroes of Functions 606

11.5 Integration and Differentiation 609

11.6 Signal Generation 611
  11.6.1 Normalized Frequency 611
  11.6.2 Wave, Pattern, and Noise VIs 615
  11.6.3 The Simulate Signal Express VI 618

11.7 Signal Processing 620
  11.7.1 The Fourier Transform 620
  11.7.2 Smoothing Windows 624
Contents

11.7.3 The Spectral Measurements Express VI 629
11.7.4 Filtering 632
11.7.5 The Filter Express VI 641

11.8 Building Blocks: Pulse Width Modulation 644
11.9 Relaxed Reading: Controlling the World’s Largest Telescope in Real Time 646

11.10 Summary 648
   Exercises 650
   Problems 651
   Design Problems 651

A Instrument Control 653
A.1 Components of an Instrument Control System 654
   A.1.1 What Is GPIB? 654
   A.1.2 GPIB Messages 655
   A.1.3 GPIB Devices and Configurations 657
   A.1.4 Serial Port Communication 659
   A.1.5 Other Bus Technologies 661
A.2 Detecting and Configuring Instruments 662
   A.2.1 Windows 662
   A.2.2 Macintosh OS X 662
A.3 Using the Instrument I/O Assistant 665
A.4 Instrument Drivers 672
   A.4.1 Developing Your Own Instrument Driver 679
A.5 Future of Instrument Drivers and Instrument Control 680
A.6 Summary 681

B LabVIEW Developer Certification 684
B.1 Overview of the NI LabVIEW Certification Structure 685
B.2 Logistics of the CLAD Examination 685
B.3 Benefits of CLAD 686
B.4 Sample CLAD Examination 687
B.5 Detailed Sample CLAD Test Solutions 698
B.6 Additional Study Resources 705
B.7 Summary 705

Index 707
Learning with LabVIEW™ is the textbook that accompanies the LabVIEW Student Edition from National Instruments, Inc. This textbook, as well as the LabVIEW software, has undergone a significant revision from the previous edition. Learning with LabVIEW teaches basic programming concepts in a graphical environment and relates them to real-world applications in academia and industry. Understanding and using the intuitive and powerful LabVIEW software is easier than ever before. As you read through the book and work through the examples, we hope you will agree that this book is more of a personal tour guide than a software manual.

The LabVIEW graphical development environment was built specifically for applications in engineering and science, with built-in functionality designed to reduce development time for design and simulation in signal processing, control, communications, electronics and more. The LabVIEW Student Edition delivers all the capabilities of the full version of LabVIEW, widely considered the industry standard for design, test, measurement, automation, and control applications. With LabVIEW, students can design graphical programming solutions to their homework problems and laboratory experiments—an ideal tool for science and engineering applications—that is also fun to use! The LabVIEW Student Edition affords students the opportunity for self-paced learning and independent project development.

The goal of this book is to help students learn to use LabVIEW on their own. With that goal in mind, this book is very art-intensive with over 400 figures in all. That means that there are numerous screen captures in each section taken from a typical LabVIEW session. The figures contain additional labels and pointers added to the LabVIEW screen captures to help students understand what they are seeing on their computer screens as they follow along in the book.

The most effective way to use Learning with LabVIEW is to have a concurrent LabVIEW session in progress on your computer and to follow along with the steps in the book. A directory of virtual instruments has been developed by the author exclusively for use by students using Learning with LabVIEW and is available on www.pearsonhighered.com/bishop. These virtual instruments
complement the material in the book. In most situations, the students are asked to develop the virtual instrument themselves following instructions given in the book, and then compare their solutions with the solutions provided by the author to obtain immediate feedback. In other cases, students are asked to run a specified virtual instrument as a way to demonstrate an important LabVIEW concept.

GAINING PRACTICAL EXPERIENCE AND SOLVING REAL-WORLD PROBLEMS

With higher education emphasizing hands-on laboratory experience, many educational institutions have improved their laboratory facilities in order to increase student exposure to practical problems. College graduates are gaining vital experience in acquiring and analyzing data, constructing computer-based simulations of physical systems, and multipurpose computer programming. LabVIEW offers a powerful, efficient, and easy-to-use development environment, allowing educators to teach their students a wide range of topics with just one open, industry-standard tool. It can also transform the way engineers, scientists, and students around the world design, prototype, and deploy cutting-edge technology. Customers and students at more than 25,000 companies and schools are using LabVIEW and modular hardware from National Instruments to simplify technology development and increase productivity. From testing next-generation gaming systems to creating breakthrough medical devices, the resulting innovative technologies are impacting millions of people worldwide.

The cover of this edition of Learning with LabVIEW shows thirteen interesting application areas that use LabVIEW in the solution process.

1. Killer Whales
2. Airliners
3. Advanced Fighter Jets
4. Wind Power
5. RF Communications
6. Mobile Instrumentation
7. Medical Devices
8. DARwIn
9. Rion-Antirion
10. Olympic Stadium
11. Video Games
12. Robotics Education
13. Motorcycles
1. **Killer Whales**

LabVIEW was used to develop a reliable data acquisition system that can collect and analyze ultrasonic signals produced by killer whales to perform live audio monitoring and movement tracking.

See page 568 for more details.

2. **Airliners**

LabVIEW was used to develop an intelligent fire monitoring and suppression control system for FedEx Express freighter aircrafts. This program will prevent catastrophic fires within the aircraft and will keep pilots, packages, and planes safe from fires that may start in shipping containers.

3. **Advanced Fighter Jets**

Engineers developed a real-time data acquisition and control system for jet and rocket engine hot-fire testing. The system needed to preserve the facility while testing article safety, data-recording reliability, accuracy, configuration flexibility and efficiency. Using LabVIEW allowed the implementation of a high-performance system that can be operated and maintained by a minimum set of technicians at costs two to three times lower than competitors’ systems.

4. **Wind Power**

Aeroponics is a method for growing plants using air or mist without soil. Energy is not invested in extending and growing the plants’ roots. Instead, it is used to increase the quality of the cultivation, as size, flavors, and nutritive properties improve. An intelligent greenhouse prototype was built offering flexibility and ease of use that suit it to places such as restaurants or domestic settings in cities. This system is powered by a 400-W wind turbine and a single 60-W photovoltaic cell that, even under poor environmental conditions, feeds the system with 3.3 kWh per day.

5. **RF Communications**

The Oscillator Development Instrument (ODIN), powered by LabVIEW, is a compact, fully integrated instrument that makes phase noise analysis simple and cost effective, vastly expands the range of applications, and improves the quality and throughput of components and signal-transmission systems.

6. **Mobile Instrumentation**

The lack of developed health care facilities and electrical distribution to rural areas presents challenges in the Republic of Malawi in southeast Africa. The remote health clinic in Makata is designed as a basic workspace with counter surfaces, an examination and treatment bed, a sink, cupboards, and a refrigerator/freezer. The facility has been in operation for more than a year. Based on data obtained using LabVIEW, DAQ, and a laptop, a very clear concept of how it functions and uses energy under all types of weather conditions was analyzed to identify key areas of energy waste.
7. **Medical Devices**
A device that will profoundly benefit people with amyotrophic lateral sclerosis (ALS, commonly known as Lou Gehrig’s disease), cerebral palsy, spinal cord injury, and other neurological disorders was developed by Michael Callahan while studying entrepreneurial engineering at the University of Illinois at Urbana-Champaign. The device, dubbed “The Audeo,” acquires and translates neurological signals so subjects who cannot speak or move can communicate.

8. **DARwIn**
Using LabVIEW, the Dynamic Anthropomorphic Robot with Intelligence (DARwIn) was programmed to perform high-level functions, such as competing in RoboCup, the international soccer tournament for autonomous robots.

9. **Rion-Antirion**
The longest cable-stayed bridge in the world, the Rion-Antirion in Greece, is being monitored by LabVIEW. The structural monitoring system measures and defines the behavior of the bridge during normal operation, strong winds, and earthquakes.

10. **Olympic Stadium**
LabVIEW was used to develop a state-of-the-art solution, employing contemporary computing, sensor, and communication technology, to monitor structural health characteristics of the 2008 Summer Olympic venues in Beijing, including stability and reliability, in real time.

11. **Video Games**
When developing the controllers for the Xbox 360, Microsoft had to create a new series of tests. With the use of LabVIEW graphical development environment more than 100 tests were created and implemented to ensure a high-quality user experience.

12. **Robotics Education**
National Instruments and LEGO, sharing a vision of inspiring creativity and innovation, have partnered to develop LEGO MINDSTORMS® programmable robots that are smarter, stronger, and more intuitive than ever. Today’s efforts will help ensure a strong network of technically proficient talent for addressing tomorrow’s problems through scientific and technological innovation.

13. **Motorcycles**
Engine control requires deterministic loop times on the order of milliseconds and precise fuel and spark timing on the order of microseconds. In addition, the target engine revs to 15,500 RPM. At this speed, the crankshaft rotates in less than 4 ms, and the system must precisely control fuel and spark events in the angle domain to less than 1 degree. Precision is key and
What's New with the LabVIEW Student Edition?

there is no room for error, which is why LabVIEW was used to build this high-performance motorcycle engine.

To learn more about these amazing engineering accomplishments, visit www.ni.com/labviewse and click on the LabVIEW Student Edition link.

THE LABVIEW STUDENT EDITION SOFTWARE

The LabVIEW Student Edition software package is a powerful and flexible instrumentation, analysis, and control software platform for PCs running Microsoft Windows or Apple Macintosh OS X. The student edition is designed to give students early exposure to the many uses of graphical programming. LabVIEW not only helps reinforce basic scientific, mathematical, and engineering principles, but it encourages students to explore advanced topics as well. Students can run LabVIEW programs designed to teach a specific topic, or they can use their skills to develop their own applications. LabVIEW provides a real-world, hands-on experience that complements the entire learning process.

WHAT’S NEW WITH THE LABVIEW STUDENT EDITION?

The demand for LabVIEW in colleges and universities has led to the development of LabVIEW Student Edition based on the industry version of LabVIEW. This is a new and significant software revision that delivers all of the graphical programming capabilities of the full edition. With the student edition, students can design graphical programming solutions for their classroom problems and laboratory experiments on their personal computers. The LabVIEW Student Edition features include the following:

- Express VIs that bring interactive, configuration-based application design for acquiring, analyzing, and presenting data.
- Interactive measurement assistants to make creating data acquisition and instrument control applications easier than ever.
- Full LabVIEW advanced analysis capability.
- Full compatibility with all National Instruments data acquisition and instrument control hardware.
- Support for all data types used in the LabVIEW Full Development System.

New LabVIEW software features introduced in this new edition of Learning with LabVIEW include:

- Using VI Snippets to store, share, and reuse small portions of LabVIEW code.
- Quickly finding and placing LabVIEW palette objects with Quick Drop.
Preface

- Simplified debugging by managing all LabVIEW probes in one window.
- A Block Diagram Cleanup Tool to automatically arrange portions of code.
- An enhanced set of editing tools for creating icons in the Icon Editor dialog box.
- Many new math plots for 2D and 3D graphs, including 2D compass plots, error plots, and feather plots, and 3D bar plots, comet plots, contour, stem, waterfall, and others.

This latest edition of *Learning with LabVIEW* also features:

- Updated exercises and design problems that reinforce the main topics of the chapter.
- New relaxed readings that illustrate how students, engineers, and scientists are using LabVIEW to solve real-world problems.
- Information on how to become a certified LabVIEW user for career advancement and employment opportunities.

**ORGANIZATION OF LEARNING WITH LABVIEW**

This textbook serves as a LabVIEW resource for students. The pace of instruction is intended for both undergraduate and graduate students. The book is comprised of 11 chapters and should be read sequentially when first learning LabVIEW. For more experienced students, the book can be used as a reference book by using the index to find the desired topics. The 11 chapters are as follows:

**CHAPTER 1:** LabVIEW Basics—This chapter introduces the LabVIEW environment and helps orient students when they open a virtual instrument. Concepts such as windows, toolbars, menus, and palettes are discussed.

**CHAPTER 2:** Virtual Instruments—The components of a virtual instrument are introduced in this chapter: front panel, block diagram, and icon/connector pair. This chapter also introduces the concept of controls (inputs) and indicators (outputs) and how to wire objects together in the block diagram. Express VIs are introduced in the chapter.

**CHAPTER 3:** Editing and Debugging Virtual Instruments—Resizing, coloring, and labeling objects are just some of the editing techniques introduced in this chapter. Students can find errors using execution highlighting, probes, single-stepping, and breakpoints, just to name a few of the available debugging tools.

**CHAPTER 4:** SubVIs—This chapter emphasizes the importance of reusing code and illustrates how to create a VI icon/connector. It also shows parallels between LabVIEW and text-based programming languages.
CHAPTER 5: Structures—This chapter presents loops, case structures, and flat sequence structures that govern the execution flow in a VI. The Formula Node is introduced as a way to implement complex mathematical equations.

CHAPTER 6: Arrays and Clusters—This chapter shows how data can be grouped, either with elements of the same type (arrays) or elements of a different type (clusters). This chapter also illustrates how to create and manipulate arrays and clusters.

CHAPTER 7: Charts and Graphs—This chapter shows how to display and customize the appearance of single and multiple charts and graphs.

CHAPTER 8: Data Acquisition—The basic characteristics of analog and digital signals are discussed in this chapter, as well as the factors students need to consider when acquiring and generating these signals. This chapter introduces students to the Measurement and Automation Explorer (MAX) and the DAQ Assistant.

CHAPTER 9: Strings and File I/O—This chapter shows how to create and manipulate strings on the front panel and block diagram. This chapter also explains how to write data to and read data from files.

CHAPTER 10: MathScript RT Module—This chapter introduces the interactive MathScript environment, which combines a mathematics-oriented text-based language with the intuitive graphical dataflow programming of LabVIEW. Both the interactive MathScript environment for command line computation and the MathScript Node for integrating textual scripts within the LabVIEW block diagram are discussed.

CHAPTER 11: Analysis—LabVIEW can be used in a variety of ways to support analysis of signals and systems. Several important analysis topics are discussed in this chapter, including how to use LabVIEW for signal generation, signal processing, linear algebra, curve fitting, formula display on the front panel, differential equations, finding roots (zero finder), and integration and differentiation.

APPENDIX A: Instrument Control—The components of an instrument control system using a GPIB or serial interface are presented in this appendix. Students are introduced to the notion of instrument drivers and of using the Measurement and Automation Explorer (MAX) to detect and install instrument drivers. The Instrument I/O Assistant is introduced.

APPENDIX B: LabVIEW Developer Certification—discusses the certification process to validate your expertise, beginning with the Certified LabVIEW Associate Developer (CLAD), continuing with the Certified LabVIEW Developer (CLD), and culminating with the Certified LabVIEW Architect (CLA). It includes a CLAD introductory-level certification practice
Preface

test with complete answers, along with information on additional resources to help you prepare for the examination.

The important pedagogical elements in each chapter include the following:

1. A brief table of contents and a short preview of what to expect in the chapter.
2. A list of chapter goals to help focus the chapter discussions.
3. Margin icons that focus attention on a helpful hint or on a cautionary note.

4. An end-of-chapter summary and list of key terms.

5. Sections entitled Building Blocks near the end of each chapter present the continuous development and modification of a virtual instrument for calculating and generating a pulse-width modulated signal. The student is expected to construct the VIs based on the instructions given in the sections. The same VI is used as the starting point and then improved in each subsequent chapter as a means for the student to practice with the newly introduced chapter concepts.

6. Many worked examples are included in each chapter including several new examples introduced in this edition. In most cases, students construct the VIs discussed in the examples by following a series of instructions given in the text. In the early chapters, the instructions for building the VIs are quite specific, but in the later chapters, students are expected to construct the VIs without precise step-by-step instructions. Of course, in all chapters, working
versions of the VIs are provided for all examples in the Learning directory included as part of the LabVIEW Student Edition. Here is a sample of the worked examples:

- Temperature system demonstration.
- Solving a set of linear differential equations.
- Building your first virtual instrument.
- Computing area, diameter, and radius of a circle.
- Computing and graphing the time value of money.
- Studying chaos using the logistic difference equation.
- Acquiring data.
- Writing ASCII data to a file.

7. A section entitled Relaxed Reading that describes how LabVIEW is being utilized to solve interesting real-world problems. The material is intended to give students a break from the technical aspects of learning LabVIEW and to stimulate thinking about how LabVIEW can be used in various other situations.

8. End-of-chapter exercises, problems, and design problems reinforce the main topics of the chapter and provide practice with LabVIEW.

ORIGINAL SOURCE MATERIALS

Learning with LabVIEW was developed with the aid of important references provided by National Instruments. The main reference was the manual LabVIEW 2009 Help with edition date of June 2009 and Part Number: 371361F-01. This excellent resource can be found at the website www.ni.com/manuals. It provides information on LabVIEW programming concepts, step-by-step instructions for using LabVIEW, and reference information about LabVIEW VIs, functions, palettes, menus, and tools. You can access this same material in LabVIEW by selecting Help >> Search the LabVIEW Help (see Chapter 1 of this book for more details on accessing the LabVIEW help). By design, there is a strong correlation between some of the material contained in the LabVIEW 2009 Help manual and the material presented in this book. Our goal here has been to refine the information content and make it more accessible to students learning LabVIEW on their own.

OPERATING SYSTEMS AND ADDITIONAL SOFTWARE

It is assumed that the reader has a working knowledge of either the Windows or the Mac OS X operating system. If your computer experience is limited, you
Preface

may first want to spend some time familiarizing yourself with your computer in order to understand the operation of your Mac or PC. You should know how to access pull-down menus, open and save files, install software from a CD, and use a mouse. You will find previous computer programming experience helpful—but not necessary.

A set of virtual instruments has been developed by the author for this book. You will need to obtain the Learning directory from the companion website to this book at Prentice Hall:

http://www.pearsonhighered.com/bishop

For more information, you may also want to visit the NI Student Edition website at

http://www.ni.com/labviewse

All of the VI examples in this book were tested by the author on a Dell PC running Windows Vista. Obviously, it is not possible to verify each VI on all the available Windows and Macintosh platforms that are compatible with LabVIEW so if you encounter platform-specific difficulties, please let us know.

If you would like information on upgrading to the LabVIEW Professional Version, please write to

National Instruments
att.: Academic Sales
11500 North Mopac Expressway
Austin, TX 78759

or visit the National Instruments website: http://www.ni.com

LIMITED WARRANTY

The software and the documentation are provided “as is,” without warranty of any kind, and no other warranties, either expressed or implied, are made with respect to the software. National Instruments does not warrant, guarantee, or make any representations regarding the use, or the results of the use, of the software or the documentation in terms of correctness, accuracy, reliability, or otherwise and does not warrant that the operation of the software will be uninterrupted or error-free. This software is not designed with components and testing for a level of reliability suitable for use in the diagnosis and treatment of humans or as critical components in any life-support systems whose failure to perform can reasonably be expected to cause significant injury to a human. National Instruments expressly disclaims any warranties not stated herein. Neither National Instruments nor Pearson Education shall be liable for any direct or indirect damages. The entire liability of National Instruments and its dealers, distributors, agents, or employees are set forth above. To the maximum extent permitted by
applicable law, in no event shall National Instruments or its suppliers be liable for any damages, including any special, direct, indirect, incidental, exemplary, or consequential damages, expenses, lost profits, lost savings, business interruption, lost business information, or any other damages arising out of the use, or inability to use, the software or the documentation even if National Instruments has been advised of the possibility of such damages.

ACKNOWLEDGMENTS

Thanks to all the folks at National Instruments for their assistance and input during the development of *Learning with LabVIEW*. A very special thanks to Erik Luther and Stephanie Orci of NI for providing day-to-day support during the final months of the project. Thanks to Stephanie Orci and Chris Tsai for reviewing the manuscript and providing a student perspective to help guide my updates to the new edition. Thanks also go to the following reviewers: Austin B. Asgill, Southern Polytechnic State University; Jeff Doughty, Northeastern University; Buford Furman, San Jose State University; R. Glynn Holt, Boston University; Thomas Koon, Binghamton University; Milivoje Kostic, Northern Illinois University; Jay Porter, Texas A&M University; and Yi Wu, Penn State University. Finally, I wish to express my appreciation to Lynda Bishop for assisting me with the manuscript preparation, for providing valuable comments on the text, and for handling my personal day-to-day activities associated with the entire production.

KEEP IN TOUCH!

The author and the staff at Pearson Prentice Hall and at National Instruments would like to establish an open line of communication with the users of the *LabVIEW Student Edition*. We encourage students to e-mail the author with comments and suggestions for this and future editions.

Keep in touch!

ROBERT H. BISHOP  
rhbishop@mail.utexas.edu

ERIK LUTHER  
National Instruments Academic Resources Manager  
erik.luther@ni.com
Learning with LabVIEW™ 2009