Learning with LabVIEW™
Learning with LabVIEW™

Robert H. Bishop

University of South Florida
CONTENTS

Preface xv

1 LabVIEW Basics 1
  1.1 System Configuration Requirements 2
  1.2 The LabVIEW Environment 2
  1.3 The Getting Started Screen 4
  1.4 Panel and Diagram Windows 6
     1.4.1 Front Panel Toolbar 8
     1.4.2 Block Diagram Toolbar 12
  1.5 Shortcut Menus 14
  1.6 Pull-Down Menus 16
     1.6.1 File Menu 16
     1.6.2 Edit Menu 16
     1.6.3 View Menu 17
     1.6.4 Project Menu 18
     1.6.5 Operate Menu 19
     1.6.6 Tools Menu 20
     1.6.7 Window Menu 21
     1.6.8 Help Menu 22
  1.7 Palettes 22
     1.7.1 Tools Palette 23
     1.7.2 Controls Palette 24
     1.7.3 Functions Palette 26
     1.7.4 Searching the Palettes and Quick Drop 27
  1.8 Opening, Loading, and Saving VIs 30
  1.9 LabVIEW Help Options 32
Contents

1.9.1 Context Help Window 32
1.9.2 LabVIEW Help 33
1.10 Building Blocks: Pulse Width Modulation 35
1.11 Relaxed Reading: Revolutionary Cancer Treatment Minimizes Damage to Healthy Tissue 37
1.12 myDAQ Building Blocks 39
  1.12.1 Building a Basic LED Circuit 39
  1.12.2 NI ELVISmx Instrument Launcher 40
  1.12.3 Testing with the ELVISmx Instrument Panel 41
1.13 Summary 42
  Exercises 44
  Problems 49
  Design Problems 51

2 Virtual Instruments 55
  2.1 What Are Virtual Instruments? 56
  2.2 Several Worked Examples 58
  2.3 The Front Panel 65
    2.3.1 Numeric Controls and Indicators 65
    2.3.2 Boolean Controls and Indicators 67
    2.3.3 Configuring Controls and Indicators 71
  2.4 The Block Diagram 71
    2.4.1 VIs and Express VIs 72
    2.4.2 Nodes 73
    2.4.3 Terminals 74
    2.4.4 Wiring 77
  2.5 Building Your First VI 79
  2.6 Data Flow Programming 85
  2.7 Building a VI Using Express VIs 87
  2.8 Building Blocks: Pulse Width Modulation 94
  2.9 Relaxed Reading: Building A Semiautonomous Vehicle 96
  2.10 myDAQ Building Blocks 99
    2.10.1 Building a Basic Thermistor Circuit 99
    2.10.2 The Thermistor Circuit Test VI 100
    2.10.3 Testing the Thermistor Circuit 100
2.11 Summary 101
Exercises 102
Problems 107
Design Problems 111

3 Editing and Debugging Virtual Instruments 115
3.1 Editing Techniques 116
3.1.1 Creating Controls and Indicators on the Block Diagram 116
3.1.2 Selecting Objects 119
3.1.3 Moving Objects 120
3.1.4 Deleting and Duplicating Objects 120
3.1.5 Resizing Objects 122
3.1.6 Labeling Objects 123
3.1.7 Changing Font, Style, and Size of Text 126
3.1.8 Selecting and Deleting Wires 129
3.1.9 Wire Stretching and Broken Wires 129
3.1.10 Aligning, Distributing, and Resizing Objects 133
3.1.11 Coloring Objects 134
3.1.12 Cleaning Up the Block Diagram 143
3.1.13 Routing Wires 146
3.1.14 Reusing Snippets of Code 147
3.2 Debugging Techniques 149
3.2.1 Finding Errors 149
3.2.2 Highlight Execution 151
3.2.3 Single-Stepping Through a VI and Its SubVIs 152
3.2.4 Breakpoints and Probes 154
3.2.5 Navigation Window 158
3.3 Property Nodes 158
3.4 A Few Shortcuts 161
3.5 Building Blocks: Pulse Width Modulation 161
3.6 Relaxed Reading: Using Graphical System Design for Tumor Treatments 164
3.7 myDAQ Building Blocks 166
3.8 Summary 169
Exercises 170
Problems 173
Design Problems 177
Contents

4 SubVIs 181
  4.1 What Is a SubVI? 182
  4.2 Review of the Basics 183
  4.3 Editing the Icon and Connector 186
    4.3.1 Icons 186
    4.3.2 Connectors 190
    4.3.3 Selecting and Modifying Terminal Patterns 191
    4.3.4 Assigning Terminals to Controls and Indicators 192
  4.4 The Help Window 193
  4.5 Using a VI as a SubVI 198
  4.6 Creating a SubVI from a Selection 201
  4.7 Error Checking and Error Handling 202
    4.7.1 Automatic Error Handling 203
    4.7.2 Manual Error Handling 204
    4.7.3 Error Clusters 204
  4.8 Saving Your SubVI 205
  4.9 The VI Hierarchy Window 206
  4.10 Building Blocks: Pulse Width Modulation 208
  4.11 Relaxed Reading: Mobile Rescue Operations 210
  4.12 myDAQ Building Blocks 212
  4.13 Summary 216
    Exercises 218
    Problems 221
    Design Problems 224

5 Structures 227
  5.1 The For Loop 228
    5.1.1 Numeric Conversion 230
    5.1.2 For Loops with Conditional Terminals 234
  5.2 The While Loop 235
  5.3 Shift Registers and Feedback Nodes 241
    5.3.1 Shift Registers 241
    5.3.2 Using Shift Registers to Remember Data Values from Previous Loop Iterations 243
    5.3.3 Initializing Shift Registers 245
    5.3.4 Feedback Nodes 247
  5.4 Case Structures 251
    5.4.1 Adding and Deleting Cases 253
    5.4.2 Wiring Inputs and Outputs 256
Contents

5.5 Flat Sequence Structures 261
   5.5.1 Evaluate and Control Timing in a Sequence Structure 262
   5.5.2 Avoid the Overuse of Sequence Structures 264

5.6 The Formula Node 264
   5.6.1 Formula Node Input and Output Variables 265
   5.6.2 Formula Statements 266

5.7 Diagram Disable Structures 268

5.8 Local Variables 269
   5.8.1 Creating Local Variables 270
   5.8.2 Use Local Variables with Care 271
   5.8.3 Initializing Local Variables 272
   5.8.4 Memory and Execution-Speed Considerations 272

5.9 Common Programming Techniques 272
   5.9.1 Sequential Programming 272
   5.9.2 State Programming and State Machines 274
   5.9.3 Parallelism 277

5.10 Some Common Problems in Wiring Structures 278
   5.10.1 Failing to Wire a Tunnel in All Cases of a Case Structure 279
   5.10.2 Overlapping Tunnels 279
   5.10.3 Wiring Underneath Rather Than through a Structure 280

5.11 Building Blocks: Pulse Width Modulation 281

5.12 Relaxed Reading: Refining the Process of Steel Recycling 284

5.13 myDAQ Building Blocks 286

5.14 Summary 288

   Exercises 291
   Problems 295
   Design Problems 299

6 Arrays and Clusters 303

6.1 Arrays 304
   6.1.1 Creating Array Controls and Indicators 305
   6.1.2 Multidimensional Arrays 307

6.2 Creating Arrays with Loops 308
   6.2.1 Creating Two-Dimensional Arrays 311

6.3 Array Functions 312
   6.3.1 Array Size 312
   6.3.2 Initialize Array 313
Contents

6.3.3 Build Array 315
6.3.4 Array Subset 316
6.3.5 Index Array 317

6.4 Polymorphism 325

6.5 Clusters 328

6.6 Creating Cluster Controls and Indicators 329
  6.6.1 Cluster Order 331
  6.6.2 Using Clusters to Pass Data to and from SubVIs 333

6.7 Cluster Functions 334
  6.7.1 The Bundle Function 334
  6.7.2 The Unbundle Function 338
  6.7.3 Creating Cluster Constants on the Block Diagram 338
  6.7.4 Using Polymorphism with Clusters 340

6.8 Matrix Data Type and Matrix Functions 341
  6.8.1 Creating Matrix Controls, Indicators, and Constants 341
  6.8.2 Matrix Functions 343

6.9 VI Memory Usage 345

6.10 Building Blocks: Pulse Width Modulation 347

6.11 Relaxed Reading: Automatic Laser-Assisted Neuron Growth 350

6.12 myDAQ Building Blocks 352

6.13 Summary 355

Exercises 357
Problems 360
Design Problems 363

7 Charts and Graphs 367

7.1 Waveform Charts 368

7.2 Waveform Graphs 376

7.3 XY Graphs 384

7.4 Customizing Charts and Graphs 387
  7.4.1 Axes Scaling 387
  7.4.2 The Plot Legend 391
  7.4.3 The Graph Palette and Scale Legend 393
  7.4.4 Special Chart Customization Features 396
  7.4.5 Special Graph Customization Features: Cursor Legend 398
  7.4.6 Using Graph Annotations 400
Contents

7.4.7 Exporting Images of Graphs, Charts, and Tables 402
7.4.8 Using Context Help 402

7.5 Using Math Plots for 2D and 3D Graphs 403
7.5.1 2D Graphs 403
7.5.2 3D Graphs 405

7.6 Building Blocks: Pulse Width Modulation 409

7.7 Relaxed Reading: Environmental Monitoring in the Costa Rican Rain Forest 412

7.8 myDAQ Building Blocks 415

7.9 Summary 417
Exercises 418
Problems 420
Design Problems 422

8 Data Acquisition 425

8.1 Components of a DAQ System 426

8.2 Types of Signals 427
8.2.1 Digital Signals 429
8.2.2 Analog DC Signals 430
8.2.3 Analog AC Signals 431
8.2.4 Analog Frequency-Domain Signals 433
8.2.5 One Signal—Five Measurement Perspectives 435

8.3 Common Transducers and Signal Conditioning 435

8.4 Signal Grounding and Measurements 439
8.4.1 Signal Source Reference Configuration 439
8.4.2 Measurement System 441

8.5 Analog-to-Digital Conversion Considerations 445

8.6 DAQ VI Organization 450

8.7 Choosing Your Data Acquisition Device 451
8.7.1 X Series Data Acquisition Devices 452
8.7.2 Low Cost Data Acquisition for Students 452
8.7.3 Simulated Data Acquisition 452
8.7.4 Macintosh, Linux, and Mobile Devices 453

8.8 DAQ Hardware Configuration 454
8.8.1 Windows 454
8.8.2 Channels and Tasks 461

8.9 Using the DAQ Assistant 463
8.9.1 DAQmx Task Name Constant 466
8.10 Analog Input 470
  8.10.1 Task Timing 470
  8.10.2 Task Triggering 471
8.11 Analog Output 475
  8.11.1 Task Timing 476
  8.11.2 Task Triggering 477
8.12 Digital Input and Output 482
8.13 Building Blocks: Pulse Width Modulation 488
  8.13.1 Generating Pulse Width Modulated Signals with Hardware Counters 489
  8.13.2 Applications of Pulse Width Modulation 490
8.14 Relaxed Reading: Reducing Seismic Risk for an Ancient Roman Amphitheater 490
8.15 myDAQ Building Blocks 492
8.16 Summary 495
  *Exercises* 498
  *Problems* 500
  *Design Problems* 500

9 Strings and File I/O 502
9.1 Strings 503
  9.1.1 Converting Numeric Values to Strings with Build Text Express VI 510
9.2 File I/O 511
  9.2.1 Writing Data to a File 515
  9.2.2 Reading Data from a File 518
  9.2.3 Manipulating Spreadsheet Files 519
  9.2.4 File I/O Express VIs 522
  9.2.5 Obtaining the Path to the System Directories 527
9.3 Building Blocks: Pulse Width Modulation 528
9.4 Relaxed Reading: On the Science of Cycling Speed 530
9.5 myDAQ Building Blocks 532
9.6 Summary 534
  *Exercises* 535
  *Problems* 537
  *Design Problems* 538

10 NI LabVIEW MathScript RT Module 541
10.1 What is MathScript RT Module? 542
10.2 Accessing the MathScript Interactive Window 543
## Contents

10.2.1 The Command History and Output Windows 545
10.2.2 Viewing Data in a Variety of Formats 546

**10.3 MathScript Help** 549

**10.4 Syntax** 551
10.4.1 Key MathScript Functions 560

**10.5 Defining Functions and Creating Scripts** 561
10.5.1 User-Defined Functions 562
10.5.2 Scripts 565

**10.6 Saving, Loading, and Exporting Data Files** 569
10.6.1 Saving and Loading Data Files 569
10.6.2 Exporting Data 571

**10.7 MathScript Nodes** 572
10.7.1 Accessing the MathScript Node 574
10.7.2 Entering Scripts into the MathScript Node 574
10.7.3 Input and Output Variables 575
10.7.4 Script Highlighting 579
10.7.5 Debugging Scripts 580
10.7.6 Saving Scripts from within the MathScript Node 587

**10.8 Applications of MathScript RT Module** 589
10.8.1 Instrument Your Algorithms 589
10.8.2 Graphical Signal Processing, Analysis, and Mathematics 590
10.8.3 Integrating Measurement Hardware 591

**10.9 Building Blocks: Pulse Width Modulation** 592

**10.10 Relaxed Reading: Acquiring and Analyzing the Bioacoustic Communication of Killer Whales** 594

**10.11 myDAQ Building Blocks** 596

**10.12 Summary** 598

*Exercises* 600

*Problems* 601

*Design Problems* 602

## 11 Analysis 604

**11.1 Linear Algebra** 605
11.1.1 Review of Matrices 605
11.1.2 Systems of Algebraic Equations 610
11.1.3 Linear System VIs 612

**11.2 Statistics and Curve Fitting** 615
11.2.1 Curve Fits Based on Least Squares Methods 616
11.2.2 Fitting a Curve to Data with Normal Distributions 620
Contents

11.2.3 The Curve Fitting Express VI 622
11.3 Differential Equations 626
11.4 Finding Zeroes of Functions 634
11.5 Integration and Differentiation 637
11.6 Signal Generation 638
  11.6.1 Normalized Frequency 639
  11.6.2 Wave, Pattern, and Noise VIs 643
  11.6.3 The Simulate Signal Express VI 646
11.7 Signal Processing 648
  11.7.1 The Fourier Transform 648
  11.7.2 Smoothing Windows 652
  11.7.3 The Spectral Measurements Express VI 657
  11.7.4 Filtering 660
  11.7.5 The Filter Express VI 669
11.8 Building Blocks: Pulse Width Modulation 672
11.9 Relaxed Reading: High-Speed Control System To Test MEMs Microshutters 674
11.10 myDAQ Building Blocks 676
11.11 Summary 679

Exercises 681
Problems 682
Design Problems 683

A Instrument Control 684

A.1 Components of an Instrument Control System 684
  A.1.1 What Is GPIB? 684
  A.1.2 GPIB Messages 685
  A.1.3 GPIB Devices and Configurations 686
  A.1.4 Serial Port Communication 689
  A.1.5 Other Bus Technologies 691
A.2 Detecting and Configuring Instruments 692
  A.2.1 Windows 692
  A.2.2 Macintosh OS X 693
A.3 Using the Instrument I/O Assistant 696
A.4 Instrument Drivers 702
  A.4.1 Developing Your Own Instrument Driver 708
A.5 Future of Instrument Drivers and Instrument Control 710
A.6 Summary 711

Index 000
Learning with LabVIEW™ presents 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. LabVIEW is 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 Learning with LabVIEW book 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 500 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.
Preface

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 tens of thousands of 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.

FEATURES OF LEARNING WITH LABVIEW

The demand for LabVIEW in colleges and universities led to the development of Learning with LabVIEW. Originally intended as a resource to accompany the LabVIEW Student Edition, this book now is more generally applicable to the industry standard version of LabVIEW and can be used to assist students at all levels who are new to graphical programming. Students that want to learn more about graphical programming with LabVIEW can enhance their learning process in and out of the classroom using Learning with LabVIEW as a guide. This is especially germane to industry professionals along their journey of lifelong learning. The Learning with LabVIEW features includes the following:

- Step-by-step tutorials that show how to construct, debug, and run LabVIEW VIs.
- Pedagogical features such as building block examples, helpful hints and warnings, relaxed readings, end-of-chapter summaries and glossaries ideal for self-paced instruction.
- Fully developed Learning directory that includes key VIs to assist in achieving desired learning outcomes.
- Building blocks across chapters that highlight each chapter material using the popular NI myDAQ device.
LabVIEW Learning Badges and Certifications

- Over 500 detailed figures with additional notes and pointers to clearly illustrate the technical material
- Exercises and problems that reinforce the main topics of the chapter.

**WHAT’S NEW IN THIS EDITION**

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

- Text and learning materials revised to reflect latest version of LabVIEW 2019.
- Available in both e-copy and printed format, including introductory videos in each chapter.
- All new LabVIEW screen captures to accurately reflect the latest version of LabVIEW in the many chapter figures.
- Set of learning outcomes established for each chapter.
- Updated VIs in the *Learning* directory to assist with student understanding and achievement of desired learning outcomes.

**LABVIEW LEARNING BADGES AND CERTIFICATIONS**

Badges and certifications provide credible ways to validate your LabVIEW expertise and can help you differentiate yourself whether you are searching for a job or just looking to move to the next level in your current organization. You can find many resources to help document and promote your LabVIEW programming accomplishments. With learning badges, you can document your understanding of engineering fundamentals and best practices using LabVIEW (and other National Instruments products). With professional industry-recognized certifications, you can demonstrate that you have the skills needed to create high-quality applications.

National Instruments offers many types of digital badges and three different levels of professional certifications. For more information on the options available (especially including preparation materials), visit the National Instruments website

https://learn.ni.com/badges

What is a badge? A badge is a digital credential issued by National Instruments in recognition of activities and knowledge levels reached. National Instruments uses Acclaim, a Pearson VUE company, to deliver and manage the digital badges. For more information, visit the site

https://www.youracclaim.com/organizations/national-instruments/badges
Preface

You should consider tracking your knowledge growth with milestones as evidenced by badge assessments. Of course, then you can share your successes with others on social media and job sites. Very importantly, employers can quickly and accurately verify your LabVIEW knowledge in key technical areas. And best of all, there is no cost to participate in the NI Badge Program and acceptance of any badges you are issued is at your discretion.

What is a professional certification? Similar to badges, the certifications are evidence of deeper understanding of key LabVIEW concepts and of reaching certain skill levels. However, with certifications you are providing evidence that you possess the skills needed to create high-quality applications, as well as fundamental knowledge. As with badges, you can share your skills on social media and job sites to advance your career and illuminate your skills. National Instruments offers three certification designations to indicate different levels of expertise with LabVIEW. The Certified LabVIEW Associate Developer (CLAD), Certified LabVIEW Developer (CLD), and Certified LabVIEW Architect (CLA) designations represent basic, advanced, and expert knowledge, respectively.

At each level of certification, you must demonstrate the following:

- **Certified LabVIEW Associate Developer**: A basic understanding of the LabVIEW environment, VI development best practices and an understanding of and ability to interpret existing code.

- **Certified LabVIEW Developer**: The ability to design and develop functional programs while minimizing development time and ensuring maintainability.
Certified LabVIEW Architect: The skill to develop a framework for an application to be executed by a team of developers given a set of high-level requirements.

In order to take a higher-level certification examination, you must have successfully passed the lower level examination(s).

Since the CLAD is the entry-level LabVIEW certification, it will be of most interest to readers of Learning with LabVIEW. There are many concepts to master to become recognized as a Certified LabVIEW Associate Developer, and note that while this textbook does not explicitly prepare you for the digital badges and professional certifications, it does prepare you to learn through discovery. The CLAD examination is a one-hour multiple-choice test consisting of forty questions administered and proctored by Pearson Vue. Visit the website

http://www.pearsonvue.com/ni/

to locate a test center near you, to access practice tests, and to schedule an examination. If you pass the examination, your certification will be valid for two years.

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.
Preface

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: NI LabVIEW MathScript RT Module—This chapter introduces the interactive MathScript environment, which combines a mathematics-oriented text-based language with the intuitive graphical data-flow 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.
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 learning outcomes 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. At the end of each chapter, we include a **myDAQ Building Blocks** section to provide the opportunity to apply the concepts introduced in the chapter using the myDAQ hardware. Each myDAQ Building Blocks exercise will help you build a project that will involve controlling LEDs and reading temperature from a thermistor to create a temperature monitoring system. As with the Building Blocks, the same VI is used as the starting point and then improved in each subsequent chapter as a means for the student to apply the chapter concepts in a my DAQ hardware environment.
7. 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. 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.

8. 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.

9. 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 references are the various LabVIEW help manuals found at the website www.ni.com/manuals. They provide 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 similar material in LabVIEW by selecting Help>>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 various LabVIEW help manuals 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 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, download software from the Internet, 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

All of the VI examples in this book were tested by the author on a Dell tablet running Windows 10. 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.

KEEP IN TOUCH!

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

Keep in touch!

ROBERT H. BISHOP
robertbishop@usf.edu
Learning with LabVIEW™