

Contents

Foreword	xv
Preface	xix
Acknowledgments	xxi
1 Roots	1
LabVIEW and Automation	1
Virtual Instruments: LabVIEW's Foundation	3
Why Use LabVIEW?	6
The Origin of LabVIEW	7
Introduction	8
A Vision Emerges	8
All the World's an Instrument	10
A Hard-Core UNIX Guy Won over by the Macintosh	11
Putting It All Together with Pictures	12
Favoring the Underdog Platform for System Design	14
Ramping up Development	15
Stretching the Limits of Tools and Machine	16
Facing Reality on Estimated Development Times	17
Shipping the First Version	19
Apple Catches up with the Potential Offered by LabVIEW	19
LabVIEW 2: A First-Rate Instrument Control Product Becomes a World-Class Programming System	23
Info-LabVIEW List: The Beginnings of an Ecosystem	24
The Port to Windows and Sun	25
LabVIEW 3	26
LabVIEW 4	27
LabVIEW Continues to Improve	28
LabVIEW 5	28
The LabVIEW RT Branch	30

LabVIEW FPGA	30
LabVIEW 6i	31
NI Forums: The Beginning of the Ecosystem	33
LAVA Forums: The LabVIEW Ecosystem Continues to Grow	33
LabVIEW 7	33
LabVIEW Champions	34
LabVIEW 8	34
LabVIEW 8.2	36
LabVIEW Tools Network	37
LabVIEW 8.6	37
LabVIEW Switches to Yearly Releases	38
LabVIEW 2009	38
CLA Summits, LabVIEW Community-Driven Events	39
Idea Exchange	40
LabVIEW 2010	40
LabVIEW 2011	41
LabVIEW 2012	42
LabVIEW 2013	43
LabVIEW 2014	45
LabVIEW 2015	45
The Knights of NI	46
LabVIEW 2016	48
LabVIEW 2017	49
LabVIEW 2018	51
GDevCon—The Ecosystem Independent from NI	53
Future Versions of LabVIEW	54
LabVIEW NXG 1.0–3.0	54
Dr. T and Jeff Kodosky Inducted into the National Inventors Hall of Fame	56
LabVIEW Release Timeline	56
LabVIEW Handles Big Jobs	58
CERN	58
2 LabVIEW Fundamentals	61
Dataflow	61
LabVIEW under the Hood	62
The Parts of a VI	62
How VIs Are Compiled	63
The LabVIEW Environment	67
Front Panel	67
Controls	68
Property Nodes	69

Block Diagram	70
Looping	71
While Loops	71
For Loops	72
Shift Registers	73
Uninitialized Shift Registers	75
Variables	77
Local Variables	77
Global Variables	78
State Machines	81
SubVIs	83
Data Types	84
Numeric Types	85
Strings	87
Parsing Strings	88
Spreadsheets, Strings, and Arrays	92
Arrays	95
Clusters	103
Waveforms	106
Data-Type Conversions	109
Conversion and Coercion	110
Intricate Conversions and Type Casting	111
Flatten To String (... Do <i>what</i> ?)	114
Enumerated Types (Enums)	115
Get Carried Away Department	116
Timing	116
Where Do Little Timers Come From?	117
Using the Built-In Timing Functions	118
Intervals	118
Timed Structures	120
Execution and Priority	123
Timing Guidelines	123
Absolute Timing Functions	124
High-Resolution and High-Accuracy Timing	126
Synchronization	127
Polling	127
Events	128
Occurrences	131
Notifiers	134

Queues	136
Me and You, Rendezvous	141
Benchmarking Code	142
3 Data Acquisition	147
Inputs and Outputs	147
Origins of Signals	147
Transducers and Sensors	148
Actuators	150
Categories of Signals	151
Connections	156
Sampling Signals	173
Sampling Theorem	173
Filtering and Averaging	175
About ADCs, DACs, and Multiplexers	177
Digital-to-Analog Converters	182
Triggering and Timing	183
Throughput	187
Writing a Data Acquisition Program	188
Bibliography	204
4 LabVIEW Object-Oriented Programming	207
What, Where, When, and Why	207
Background	207
What?	208
Where and When?	219
Why?	224
How?	226
HAL: Hardware Abstraction Layers	243
MAL: Measurement Abstraction Layers	244
Actor Framework: The Most Recognizable	
LVOOP Architecture	244
SOLID Principles of Object-Oriented Design	245
SRP: Single Responsibility Principle	247
OCP: The Open-Closed Principle	250
LSP: The Liskov Substitution Principle	252
ISP: The Interface Segregation Principle	255
DIP: The Dependency Inversion Principle	263
Caveats	268
Accessors Get/Set Instead of Read/Write	269
Classes in LabVIEW NXG	269
References	269

5	Why Would You Want to Use a Framework?	273
	What?	274
	What Is a Process?	274
	What Is an Abstraction Layer?	274
	What Is a Framework?	275
	When Do You Make the Decision to Break Your Application into Parallel Tasks?	275
	What Are the Design Decisions That Need to Be Nailed Down at the Beginning to Make a Parallel Design Successful?	276
	What Project-Level Enforcement Is Available to Help Make Better Decisions?	278
	Project Libraries	279
	Frameworks for LabVIEW	281
	Advantages of Using a Framework	282
	Disadvantages of Using a Framework	283
	Framework versus Design Pattern	283
	The Contract between the Framework and the Programmer	283
	Why Not Make Your Own Framework?	284
	Criteria to Evaluate Frameworks	285
	Key Components	286
	Interprocess Communication	287
	Module Initialization	287
	Stop Processes Gracefully	288
	Error Handling Strategy	288
	Sharing Modules	289
	Configuring Source Code Control Repository Dependencies	290
	Packaging Modules Using VIPM	290
	Packaging Modules in PPLs	292
	DQMH	298
	Use Cases	298
	What Is DQMH?	298
	How to Use DQMH	312
	DQMH: Behind the Scenes	343
	Other DQMH Tools	351
	Sharing Reusable DQMH Modules	352
	Actor Framework	354
	Use Cases	354
	What Is Actor Framework?	354
	How to Use Actor Framework	366
	Actor Framework: Behind the Scenes	387

Other Actor Framework Tools	391
Advanced Actor Framework	392
Sharing Reusable Actors	398
References	399
6 Unit Testing	401
What Is Unit Testing?	402
Unit Testing as Code Documentation	403
What Makes a Good Unit Test?	403
What Are Assertions?	404
Writing VIs with Testing in Mind	405
Test Harness versus Automated Test Frameworks	406
What Are the Automated Test Frameworks Available for LabVIEW? ...	406
What Is the Difference Between Black Box and White Box Testing? ...	407
What Is TDD?	407
What Is Regression Testing?	408
Getting Started with Unit Testing	408
Test Coverage	411
Test Coverage Example	411
Test Vectors	419
What about Testing Classes? Public versus Private VIs in Libraries	424
Example of Unit Testing for a LabVIEW Class	425
When Are Setup and Teardown Required?	447
Should You Add Test Cases That Are Designed to Fail?	449
What about Testing When the Expected Output Is an Array of NaN (Not a Number)?	449
When Would You Create Unit Tests for a DQMH Public API?	449
Unit Testing for a DQMH Module	450
What about RT?	464
Example of Unit Testing for LabVIEW RT	464
What about FPGA VIs?	466
Example of Unit Testing for LabVIEW FPGA	470
Unit Tests for the GUI?	475
Unit Test Reporting	475
Example of Assertions	475
Assertions with Caraya	477
Assertions with AssertAPI	479
Unit Testing and LabVIEW NXG	482
Tool Comparison	483
Unit Testing Tool Alternatives for LabVIEW	484
References	485

7	Developing in LabVIEW for Teams	487
	Where Is Your Team At?	488
	What Is the Problem You Are Trying to Solve?	490
	What Is Technical Wealth?	491
	From Model to Code	494
	What Is a Model?	494
	Wasn't LabVIEW Supposed to Remove the Need for Modeling?	494
	Why and When to Use Models	497
	Types of Models	497
	Modelling Tools	501
	Source Code Control—The Developer's Time Machine	501
	What Is Source Code Control?	502
	Centralized Source Code Control	503
	Distributed Source Code Control	506
	Branch Merge versus Code Merge	509
	LabVIEW Compare	509
	LabVIEW Merge	513
	Good Practices for Source Code Control	514
	Establish a Source Code Workflow	518
	How to Select the Source Code Control Tool and the Workflow for Your Team	522
	Workstations	523
	Virtual Machines	525
	Workstations Setup	526
	Build Server	526
	Test Computer	527
	LabVIEW Style Guidelines	527
	Code Review Process	528
	Code Review Frequency	530
	VI Analyzer Tests Configuration	531
	VI Analyzer Report	532
	Prepare Code and Documentation for a Code Review	533
	LabVIEW Compare for Code Reviews	533
	Code Review Checklist	535
	Conducting a Code Review	535
	Postreview Actions	536

CASE Tools	536
Why Do You Want to Use LabVIEW VI Scripting?	537
Common Areas That Use LabVIEW Scripting	537
Five Steps to Become a VI Scripting Ninja	538
Deployment, Continuous Integration, and Continuous Delivery	542
Build Engine	542
Build Environment	546
Other Tips	548
References	549
8 Enterprise and IoT Messaging	551
MQTT Messaging Protocol	555
Install a Message Broker	558
MQTTDQMH Client	561
Getting Started	561
MQTTDQMH Application	567
Messaging in the Cloud	571
Toolkits	572
AWS IoT	573
Enabling the IoT Message Broker and CloudWatch	574
Create Certificates and Keys	574
IoT Policies	577
DynamoDB	578
IoT Rules	580
Abbreviation	589
Index	591