

---

# Contents

Preface	xvii
Acknowledgments	xxiii
<b>Chapter 1. Roots</b>	<b>1</b>
LabVIEW and Automation	2
Virtual Instruments: LabVIEW's Foundation	4
Why Use LabVIEW?	7
The Origin of LabVIEW	8
Introduction	9
A Vision Emerges	9
All the World's an Instrument	11
A Hard-Core UNIX Guy Won Over by the Macintosh	12
Putting It All Together with Pictures	13
Favoring the Underdog Platform for System Design	15
Ramping Up Development	16
Stretching the Limits of Tools and Machine	17
Facing Reality on Estimated Development Times	19
Shipping the First Version	20
Apple Catches Up with the Potential Offered by LabVIEW	21
LabVIEW 2: A First-Rate Instrument Control Product Becomes a World-Class Programming System	23
The Port to Windows and Sun	25
LabVIEW 3	26
LabVIEW 4	27
LabVIEW Branches to Bridge VIEW	28
LabVIEW 5	29
The LabVIEW RT Branch	30
LabVIEW 6	31
Crystal Ball Department	32
LabVIEW Influences Other Software Products	33
LabVIEW Handles Big Jobs	35
<b>Chapter 2. Getting Started</b>	<b>37</b>
How to Assemble a System	37
Use Equipment That's Familiar	38
Picking an I/O System	38
Picking a Computer	45
Other Software to Consider	48

Sources for Help and Answers	50
Get Trained	50
Use the Example VIs	51
When All Else Fails, Read the Directions	52
LabVIEW on the Web	52
The Internet Mailgroup	54
E-Mail Help from National Instruments	54
Frequently Asked Questions (FAQ) Document	55
<i>LabVIEW Technical Resource (LTR)</i>	55
User Groups and the VI UserNet	55
The Alliance Program: Consultants and More	56
Educational Applications	57
(800) IEEE 488	57
Bibliography	57
<b>Chapter 3. Inputs and Outputs</b>	<b>59</b>
Origins of Signals	59
Transducers and Sensors	60
Actuators	62
Categories of Signals	63
Connections	67
Grounding and Shielding	68
Why Use Amplifiers or Other Signal Conditioning?	75
Choosing the Right I/O Subsystem	83
Network Everything!	87
<b>Chapter 4. Sampling Signals</b>	<b>89</b>
Sampling Theorem	89
Filtering and Averaging	91
About ADCs, DACs, and Multiplexers	93
Digital to Analog Converters	98
Digital Codes	99
Triggering and Timing	99
A Little Noise Can Be a Good Thing	101
Throughput	104
Bibliography	106
<b>Chapter 5. Controlling Program Flow</b>	<b>107</b>
About the Diagrams in This Book	108
Sequencing and Dataflow	108
Sequence Structures	109
Data Dependency	111
Adding Common Threads	112
Looping	113
Subtleties of the For Loop	113
Wonderful Whiles	114
Shift Registers	115
Uninitialized Shift Registers	117
Global and Local Variables	119
Built-in Global Variables—and Their Hazards	121
Local Variables	125



x Contents

Draw a Block Diagram	211
Specify the I/O Hardware	213
Prototype the User Interface	214
User Interface Design	215
Panel Possibilities	216
Design, Then Write Your Program	219
Ask a Wizard	220
Top-Down or Bottom-Up?	220
Modularity	222
Choose an Architecture: Canonical VIs	223
The VI Hierarchy as a Design Tool	235
Programming by Plagiarizing	236
Pay Attention to Your Data	237
Sketching Program Structure	239
Pseudocoding	240
Language Translation	243
Ranges, Coercion, and Default Values	243
Handling Errors	245
Putting It All Together	248
Testing and Debugging Your Program	249
See What the SubVIs Are Up To	250
Peeking at Data	250
One Step at a Time	251
Execution Highlighting	252
Setting Breakpoints	253
Debugging Global Variables	254
Debugging Local Variables and Attribute Nodes	255
Tracing Execution	256
Checking Performance	258
Final Touches	258
VBL Epilogue	259
Bibliography	260
<b>Chapter 11. Documentation</b>	<b>261</b>
VI Descriptions	261
Control Descriptions	262
Custom Online Help	263
Documenting the Diagram	264
VI History	265
Other Ways to Document	265
Printing LabVIEW Panels and Diagrams	265
Putting LabVIEW Screen Images into Other Documents	266
Writing Formal Documents	268
Document Outline	269
Connector Pane Picture	269
VI Description	270
Terminal Descriptions	271
Programming Examples	272
Distributing Documents	272
<b>Chapter 12. Instrument Drivers</b>	<b>275</b>
About the Instrument Library	275
Interchangeable Virtual Instruments (IVI)	277
Driver Basics	278
Communication Standards	278

211	Learn About Your Instrument	291
213	Determine Which Functions to Program	292
214	Establish Communications	293
215		
216		
219	<b>Chapter 13. Instrument Driver Development Techniques</b>	<b>301</b>
220	Driver Architectures	302
220	Modularity by Grouping of Functions	303
222	Error I/O Flow Control	304
223	Start with a Template	310
235	Open a Connection	311
236	Using a Communications SubVI	312
237	GPIB Communications SubVI	312
239	Serial Communications SubVI	314
240	VXI Communications SubVIs	318
243	Read and Write in the Same SubVI	319
243	Many Functions in One Driver VI	321
245	Timing, Handshaking, and Triggering	323
248	GPIB Handshaking	323
249	Serial Handshaking	325
250	Range Checking	327
250	Setup Management	329
251	Documentation	332
253	Bibliography	334
254		
255		
256	<b>Chapter 14. Using the DAQ Library</b>	<b>335</b>
258	The NI-DAQ Driver	336
258	The NI-DAQ Channel Wizard	337
259	Hardware Options	338
260	Pick a Platform, Pick a Bus	338
	Multifunction I/O Boards	339
	Specialized Analog I/O Boards	339
	Digital and Timing I/O Boards	340
	SCXI	341
	Portable DAQ	342
	Direct Memory Access (DMA)	342
	Basics of the DAQ Library	343
	DAQ Library Hierarchy	343
	Conventions Used in the DAQ Library	345
	DAQ Examples and Wizards	353
<b>261</b>		
261		
262		
263		
264		
265		
265		
265		
265		
266		
268		
269		
269		
270		
271		
272		
272		
<b>275</b>		
275		
277		
278		
278		
	<b>Chapter 15. Analog DAQ</b>	<b>355</b>
	Analog Inputs	355
	Configuration, Starting, and Stopping	355
	Low-Speed Acquisition	357
	Medium-Speed Acquisition and Processing	362
	High-Speed Disk Streaming	365
	Special Sampling	368
	Analog Outputs	380
	Configuring and Stopping	381
	Simple Updates	381
	Waveform Generation	383
	Simultaneous Analog Input and Output	388
	Bibliography	389

<b>Chapter 16. Digital DAQ</b>	<b>391</b>
Digital I/O	391
Simple, Bitwise I/O	391
Digital I/O with Handshaking	394
Counters and Timers	397
About the Am9513 System Timing Controller (STC)	398
Event Counter	399
Interval, Duration, and Period Timers	402
Pulse Generation	405
Frequency Counting	406
<b>Chapter 17. Writing a Data Acquisition Program</b>	<b>409</b>
Data Analysis and Storage	411
Post-Run Analysis	412
Real-Time Analysis and Display	423
Sampling and Throughput	431
Signal Bandwidth	431
Oversampling and Digital Filtering	432
Timing Techniques	438
Configuration Management	438
What to Configure	439
Configuration Editors	441
Configuration Compilers	451
Saving and Recalling Configurations	454
A Low-Speed Data Acquisition Example	459
Bibliography	463
<b>Chapter 18. Process Control Applications</b>	<b>465</b>
Process Control Basics	466
Industrial Standards	466
Control = Manipulating Outputs	472
Process Signals	475
Control System Architectures	477
Working with Smart Controllers	483
Programmable Logic Controllers (PLCs)	483
Single-Loop Controllers	488
Other Smart I/O Subsystems	492
Man-Machine Interfaces	492
Display Hierarchy	498
Other Interesting Display Techniques	502
Handling All Those Front Panel Items	503
Data Distribution	504
Input Scanners as Servers	505
Handling Output Data	506
Display VIs as Clients	509
Using Network Connections	511
Real-Time Process Control Databases	514
Simulation for Validation	515
Sequential Control	516
Interlocking with Logic and Tables	516
State Machines	517
Initialization Problems	519
GrafsetVIEW—A Graphical Process Control Package	520
Continuous Control	522
Designing a Control Strategy	523

391	Trending	529
391	Real-Time Trends	529
391	Historical Trends	532
394	Statistical Process Control (SPC)	536
397	Alarms	537
398	Using an Alarm Handler	539
399	Techniques for Operator Notification	541
402	Commercial Packages	543
405	Bibliography	544
406		
<b>409</b>	<b>Chapter 19. Physics Applications</b>	<b>547</b>
411	Special Hardware	548
412	Signal Conditioning	548
423	CAMAC	552
431	Other I/O Hardware	552
431	Field and Plasma Diagnostics	554
432	Step-and-Measure Experiments	554
438	Plasma Potential Experiments	561
438	Handling Fast Pulses	567
438	Transient Digitizers	567
439	Digital Storage Oscilloscopes (DSOs)	570
441	Timing and Triggering	571
451	Capturing Many Pulses	574
454	Recovering Signals from Synchronous Experiments	578
459	Handling Huge Data Sets	580
463	Reducing the Amount of Data	581
	Optimizing VIs for Memory Usage	582
	Bibliography	588
<b>465</b>	<b>Chapter 20. Data Visualization, Imaging, and Sound</b>	<b>589</b>
466	Graphing	590
466	Displaying Waveform and Cartesian Data	592
472	Bivariate Data	597
475	Multivariate Data	599
477	3D Graphs	604
483	The Intensity Chart	605
483	Image Acquisition and processing	606
488	System Requirements for Imaging	608
492	Using IMAQ Vision	611
492	IMAQ Components	611
498	Sound I/O	620
502	DAQ for Sound I/O	620
503	Sound I/O Functions	621
504	Sound Input	621
505	Sound Output	622
506	Sound Files	623
509	Bibliography	624
511		
514		
515		
516	<b>Chapter 21. LabVIEW RT Does Real Time</b>	<b>625</b>
516	Real Time Does Not Mean Real Fast	625
517	RT Hardware	627
519	Designing Software to Meet Real-Time Requirements	630
520	Measuring Performance	631
522	Shared Resources	635
523		

Multithreading and Multitasking	637
Organizing VIs for Best Real-Time Performance	640
Context Switching Adds Overhead	642
Scheduling	643
Round-Robin Scheduler	643
Static Scheduler	644
Communications	645
Building an Application	647
Bibliography	648
<b>Chapter 22. Embedding LabVIEW for Linux in a Virtual Machine</b>	<b>649</b>
Why Linux?	650
Linux Resources	652
Linux on the CD-ROM	652
Building Embedded Systems with PeeWeeLinux	653
System Disk Storage Requirements	654
VMware Workstation Makes Virtual Machines	654
Building a New VMware Workstation Virtual Machine in Linux	657
Six Steps to Building an Embedded Linux System	660
Step 1. Partition the Virtual Hard Drive	660
Step 2. Format the Virtual Hard Drive	662
Step 3. Mount the Target Partition	662
Step 4. Transfer the Operating System from the CD-ROM	663
Step 5. Install the Bootloader	663
Step 6. Boot Up Your New Embedded System!	664
A Tour Through Your Embedded Linux	664
Directory Structure	665
The Boot Process	666
The Configuration Files	667
Transfer the OS via NFS Mount	670
The VMware-mount Utility and Linux	672
Mounting Your Virtual Disk with VMware-mount	672
Helpful Tips	673
Doing LabVIEW Development	674
Moving to Real Hardware	674
Bibliography	675
<b>Chapter 23. LabVIEW and Linux in Embedded Systems</b>	<b>677</b>
Why Use a Desktop Operating System to Build an Embedded System?	679
Linux to the Rescue	679
Platforms	680
Store It in Flash Memory	681
I/O Hardware Drivers	683
Hardware Drivers and Where to Get Them	685
Integrating Drivers into Your System	686
Let's Build a Real Embedded System!	687
Seven Steps to Success	692
Step 1. Partition the Hard Drive	692
Step 2. Format the Hard Drive	693
Step 3. Mount the Hard Drive	694
Step 4. Transfer the Operating System and Embedded Application	694
Step 5. Modify the Installation for the New Hardware	695
Step 6. Install the Bootloader	700
Step 7. Boot Up Your New Embedded System	702

Tes  
C  
V  
Usi  
Mo  
Bib  
  
Chap  
Ov  
V  
ε  
Ins  
Pr  
Re  
F  
V  
Sh  
T  
V  
Ha  
Th  
Th  
Bit  
  
Ap  
Ap  
Inc



