Inhalt

Teil I: Grundlagen des Programmierens in LabVIEW	17
1 Was ist LabVIEW?	19
1.1 Entwicklungsstufen	19
1.2 Was will dieses Lehrbuch?	21
1.3 Installation	21
1.4 Einführendes Beispiel	21
1.4.1 Programmierung von $c = a + b$	25
1.4.2 Speicherung als Programm Add.vi	28
1.4.3 Starten und Stoppen von Add.vi	28
1.4.4 Fehlersuche in Add.vi (Debugging)	29
1.5 Beispiel für eine Grafik in LabVIEW	30
1.6 Grundlegende Konzepte von LabVIEW	31
1.6.1 Frontpanel	31
1.6.2 Blockdiagramm	31
1.7 Rezepte	32
1.8 Shortcuts	33
2 Einstellungen, Sprachenwahl, Paletten	35
2.1 Einstellungen	35
2.1.1 'Neuerungen und Änderungen'	35
2.1.2 Frontpanel	36
2.1.3 Blockdiagramm	37
2.1.4 Ausrichtungsgitter	40
2.1.5 Wiederherstellungen	40
2.2 Sprachenwahl	40
2.2.1 Menüpunkte	41
2.2.2 Werkzeuge – Optionen	41
2.2.3 Datei – VI-Einstellungen	41
2.3 Paletten	42
2.3.1 Werkzeugpalette (Tools Palette)	42
2.3.2 Eingabe-/Ausgabe-Elemente	44
2.3.3 Funktionenpalette	46
3 Programmstrukturen	49
3.1 Strukturiertes Programmieren	49
3.2 Sequenz	51
3.3 Alternative, Case-Struktur	56
3.4 Schleifen	60

4	Datentypen	65
4.1	Numerische Datentypen	65
4.1.1	Kontextmenü: 'Darstellung'	65
4.1.2	Kontextmenü: 'Anzeigeformat...'	66
4.2	Boolesche Datentypen	68
4.3	String und Pfad	70
4.4	Arrays	73
4.4.1	Definition und Initialisierung eines 1-dimensionalen Arrays	73
4.4.2	Definition und Initialisierung eines 2-dimensionalen Arrays	75
4.4.3	Array erstellen	76
4.4.4	Rechnen mit Arrays: Addition	77
4.4.5	Rechnen mit Arrays: Multiplikation	78
4.4.6	Steuerung von For-Schleifen mit Arrays	79
4.4.7	Behandlung einzelner Arrayelemente	81
4.5	Cluster	83
4.5.1	Erzeugung eines Clusters	84
4.5.2	Clusterwerte ändern	85
4.5.3	Aufschlüsseln eines Clusters	87
4.5.4	Umordnen der Elemente eines Clusters	88
4.5.5	Cluster-Arrays	89
4.6	Ring & Enum	90
4.7	Datentyp Variant	93
4.8	Guter Programmierstil	94
5	Unterprogramme und Typdefinitionen	95
5.1	Wozu Unterprogramme (SubVIs)?	95
5.2	Erstellen von Unterprogrammen	96
5.2.1	Einführendes Beispiel	96
5.2.2	Weitere Hinweise für die Erstellung eines Unterprogramms	99
5.2.3	Einstellungen für Programme und Unterprogramme	101
5.2.4	Erstellen von Unterprogrammen mit internem Zustand	103
5.2.5	Erstellen von polymorphen Unterprogrammen	104
5.3	Aufruf von Unterprogrammen	107
5.3.1	Statische Bindung	107
5.3.2	Dynamische Bindung	108
5.3.2.1	VI-Referenz öffnen und schließen	108
5.3.2.2	Aufruf eines VI über seine Referenz	110
5.3.2.3	Beispiel für den SubVI-Austausch während der Laufzeit	111
5.3.2.4	Rekursiver Aufruf von Unterprogrammen	111
5.3.2.5	Testen (Debugging) von ablaufinvarianten SubVIs	112
5.4	Typdefinitionen	114
5.4.1	Beispiel einer Typdefinition für Enum-Variablen	114
5.4.2	Beispiel einer Typdefinition für Registerkarten	116
5.4.3	Beispiel einer Cluster-Typdefinition	117
5.5	Guter Programmierstil	118
5.5.1	Vereinfachung durch Unterprogramme und Typdefinitionen	118
5.5.2	Kommentierung der Elemente und Funktionen eines VI	119

6	Prozessvisualisierung, Programmierstil	121
6.1	OOP-Konzepte	121
6.2	Eigenschafts- und Methodenknoten	121
6.3	Grafische Ausgabe	125
6.3.1	Chart (Signalverlaufdiagramm).....	125
6.3.1.1	Darstellung einer Sinuskurve.....	125
6.3.1.2	Darstellung von zwei oder mehr Kurven in einem Chart.....	127
6.3.1.3	Legende zu einem Chart oder Graphen	128
6.3.1.4	Skalierung der Ordinate in einem Chart	129
6.3.2	Graph (Signalverlaufsgraph)	131
6.3.2.1	Darstellung einer Sinuskurve.....	131
6.3.2.2	Darstellung von zwei oder mehr Kurven in einem Graphen	133
6.3.2.3	Skalierung der Abszisse in einem Graphen.....	135
6.3.3	XY-Graph	137
6.3.3.1	Darstellung einer Relation im XY-Graphen	138
6.3.3.2	Darstellung mehrerer Relationen in einem XY-Graphen	139
6.3.4	Signalverlauf	140
6.4	Express-VIs, Programmierstil.....	145
6.4.1	Express-VI zur Erzeugung von Kurven	145
6.4.2	Express-VI zur Erstellung von Berichten.....	146
7	Referenzen, Fehlerfunktionen	148
7.1	Einführendes Beispiel.....	148
7.1.1	Vertauschung von zwei Variablenwerten	148
7.1.2	Referenzen auf Bedien- und Anzeigeelemente	149
7.1.3	Lösung des Vertauschungsproblems	150
7.2	Vererbung.....	151
7.2.1	Eigenschaften der Basisklasse	152
7.2.2	Eigenschaften von abgeleiteten Klassen	153
7.3	Fehlerfunktionen	155
7.3.1	Fehlermeldungen mit oder ohne Dialog.....	155
7.3.2	Wo findet man wichtige Fehlerelemente und Fehlerfunktionen?.....	156
7.3.3	Verschiedene Fehlerarten	157
7.3.3.1	Standardfehlerleitung	157
7.3.3.2	Funktionen ohne oder mit vereinfachter Fehlerleitung	157
7.3.4	Ausgang aus While-Schleifen	159
7.3.5	Erzwingung von sequenziellem Ablauf.....	160
8	Datentransfer von und zur Festplatte	161
8.1	Dateifunktionen.....	161
8.1.1	Allgemeines zur Speicherung von Dateien	161
8.1.2	Palette Dateifunktionen	163
8.1.3	Einführendes Beispiel	164
8.1.4	Modifiziertes Beispiel	165
8.1.5	Beispiel: Anlegen einer Protokolldatei	166
8.1.6	Überschreiben ohne Warnung.....	166
8.2	Pfade	167
8.2.1	Pfadkonstanten	167
8.2.2	Pfadkonstante 'Standardverzeichnis'	169

8.2.3	'Standardverzeichnis' ändern	170
8.2.4	'Standarddatenverzeichnis' ändern	170
8.2.5	Lesen und Schreiben anderer Datentypen	171
8.2.6	Verketteten von Schreib- und Lesefunktionen	172
8.2.7	Tabellenkalkulation	173
8.3	Pfade in einer EXE-Datei	173
8.4	Fortgeschrittene Dateitypen.....	175
8.4.1	LVM-, TDMS- und TDM-Dateien.....	175
8.4.2	Diadem	178
8.4.3	ZIP-Dateien	179
8.4.4	Konfigurationsdateien	180
9	LabVIEW-Kurzüberblick	185
9.1	Aufbau des LabVIEW-Systems	185
9.1.1	Programmierung in G	185
9.1.1.1	Interpretieren oder kompilieren?	185
9.1.1.2	Datenflussprogrammierung	187
9.1.2	Hardware-Unterstützung	187
9.1.3	Bibliotheken mathematischer und technischer Funktionen.....	188
9.1.4	Benutzerschnittstelle	189
9.1.5	Technologische Abstraktion	190
9.1.6	Rechenmodelle	190
9.2	Projekte	190
9.3	Erstellung von EXE-Dateien	192
9.3.1	Warum EXE-Dateien?	192
9.3.2	Erstellung einer EXE-Datei.....	192
9.3.3	EXE-Datei auf einem Rechner ohne LabVIEW-System.....	195
9.4	Strukturen zur Programmentwicklung	198
9.4.1	Deaktivierungsstrukturen.....	198
9.4.2	Debug-Einstellung in der Projektverwaltung	200
9.5	LabVIEW-Bibliotheken	201
9.6	Umwandeln von LLB-Bibliotheken.....	204
9.7	Einbindung von C-Funktionen unter Windows.....	205
9.7.1	Reihe in C#	207
9.7.2	Reihe in C++	211
9.7.3	Reihe mit MathScript	215
9.8	Hilfen zu LabVIEW	215

Teil II: Technische Anwendungen

219

10	Fouriertransformation	220
10.1	Zeit- und Frequenzbereich.....	220
10.1.1	Die reelle Fouriertransformation	221
10.1.2	Darstellung der Fourierkoeffizienten c_k in LabVIEW	224
10.2	Diskrete Fouriertransformation.....	227
10.2.1	Satz von Shannon.....	228
10.2.2	Aliasing.....	229
10.2.3	Frequenzauflösung.....	230

11	Filterung	232
11.1	Filtertypen	232
11.1.1	Ideale und reale Filter.....	232
11.1.2	Beispiel eines digitalen Filters	233
11.2	LabVIEW-Filterfunktionen	236
11.3	Filterung im Frequenzbereich	238
11.3.1	Idee der Filterung im Frequenzbereich	239
11.3.2	Die inverse Fouriertransformation in LabVIEW.....	239
11.3.3	Beispiel eines Tiefpasses	239
12	Differenzialgleichungen	242
12.1	Lösen mit LabVIEW-ODE-Funktionen	242
12.2	Lösen nach dem Analogrechnerprinzip	245
12.2.1	Blockdiagramm-Darstellung	245
12.2.2	Vereinfachungen.....	248
12.3	Globale Variablen	249
12.4	Genauigkeit numerischer Verfahren	250
13	Systeme von Differenzialgleichungen	254
13.1	Systeme gewöhnlicher Differenzialgleichungen	254
13.2	Gekoppeltes Feder-Masse-System	254
13.2.1	Lösung mit eingebauter ODE-Funktion	255
13.2.2	Lösung mit Blockdiagramm wie in MATLAB®	256
13.3	Umwelt und Tourismus	258
14	Parallelverarbeitung, Laufzeiten, Ereignisse	261
14.1	Einführendes Beispiel	261
14.2	Grundbegriffe der Parallelverarbeitung.....	263
14.2.1	Multiprocessing, Multitasking, Multithreading	263
14.2.2	Synchronisierung von Prozessen.....	264
14.3	Parallelverarbeitung unter LabVIEW	265
14.3.1	Erzeugen von Ressourcen für die Prozesskommunikation.....	267
14.3.2	Freigabe von Ressourcen der Prozesskommunikation	269
14.3.3	Zeitbegrenzung Ressource schont Prozessor	270
14.4	Prozess-Synchronisierung ohne Datenaustausch	270
14.4.1	Occurrences.....	270
14.4.2	Semaphor.....	271
14.4.3	Rendezvous	273
14.5	Prozess-Synchronisierung mit Datenaustausch	274
14.5.1	Melder-Operationen	274
14.5.2	Queue-Operationen.....	275
14.6	Laufzeitprobleme und ihre Behandlung.....	276
14.6.1	Laufzeitprobleme bei lokalen Variablen	276
14.6.2	Laufzeitprobleme bei globalen Variablen	279
14.7	Ereignisgesteuerte Programmierung	280
14.7.1	Frontpanel-Ereignisse.....	280
14.7.2	Wertänderungs-Ereignisse	285
14.7.3	Gefilterte Ereignisse	286
14.8	Zeitschleifen	288

Teil III: Kommunikation	291
15 Serielle Eingabe/Ausgabe	292
15.1 RS-232	292
15.2 Programmierung der RS-232 in LabVIEW	294
15.3 Die USB-Schnittstelle	297
15.3.1 Programmierung unter Windows	298
15.3.2 Programmierung unter Linux	301
15.4 Feld-Bus, CAN-Bus	302
15.4.1 CAN-Protokoll	302
15.4.2 CAN-Interface	304
15.4.3 CAN-Bus mit Laptop und einem Sensor	307
15.4.4 CANopen-Protokoll, ZILA-Sensor	311
15.4.5 CAN-Bus mit Laptop und zwei Sensoren	313
15.4.6 XNET-System von National Instruments	314
15.5 Der byte-serielle GPIB-Bus	325
15.5.1 Installation der Software	327
15.5.2 Entwicklung eines VI	328
16 Datenerfassungsgeräte	331
16.1 Datenerfassungskarten / Datenerfassungsgeräte	331
16.2 Allgemeines	332
16.2.1 Treiber, MAX (Measurement and Automation Explorer)	332
16.2.2 Physikalische und virtuelle Kanäle, Task	338
16.2.3 Programmierung von Datenerfassungs-VIs, simulierte Geräte	339
16.2.4 Programmierung von VIs zur Analogausgabe	344
16.2.5 Programmierung von VIs zum Digital-I/O	345
16.2.6 Programmierung mit Hilfe des DAQ-Assistenten	345
16.2.7 Programmatische Task-Erstellung	347
16.3 USB-Gerät NI USB-6251	348
16.3.1 Begriffe 'differenziell', 'RSE' und 'NRSE'	349
16.3.2 Zwei Analogsignale mit der NI USB-6521 lesen	350
16.3.3 Triggern mit NI USB-6521	351
16.3.4 Streaming mit NI USB-6521	353
16.4 Veraltete Datenerfassungskarten/-geräte	359
16.5 TEDS	359
16.6 IVI-Gerät NI USB-513	363
Teil IV: Fortgeschrittene Techniken	369
17 Professionelle Programmentwicklung	370
17.1 Sequenzstruktur	370
17.2 Zustandsautomaten	371
17.2.1 Notation für Zustandsautomaten	372
17.2.2 Umsetzung Zustandsdiagramm → LabVIEW-Programm	373
17.2.2.1 Strings für die Zustandsauswahl	374
17.2.2.2 Enum für die Zustandsauswahl	376
17.3 Münzautomat	377

17.4	Münzautomat mit Queues und Ereignisstrukturen	386
17.5	Programmierhilfen	389
17.5.1	Arbeiten mit vorgefertigten Strukturen (Templates)	389
17.5.2	Beurteilung Programmeffizienz und geeignete Werkzeuge dazu	390
18	Objektorientierte Programmierung	394
18.1	Warum objektorientiert?	394
18.2	Erstes Beispiel zur objektorientierten Programmierung	397
18.2.1	Bildung einer Klasse	397
18.2.2	Private Eigenschaften der Klasse	398
18.2.3	Methoden der Klasse	399
18.3	Weitere Beispiele zur OOP	403
18.3.1	Vererbung	403
18.3.2	Polymorphie	406
18.3.3	Modulaustausch	410
18.4	Schutz einer Klassenbibliothek	417
19	LabVIEW: Tabellenkalkulation, Datenbanken	420
19.1	Schreib-/Lesebefehle zur Tabellenkalkulation	420
19.2	Allgemeines über ActiveX	422
19.2.1	ActiveX-Container in LabVIEW	423
19.2.2	ActiveX in LabVIEW zur Steuerung von Anwendungen	425
19.3	Beispiele zur Anwendung auf Excel	426
19.3.1	Öffnen und Schließen von Excel	427
19.3.2	Sichtbarmachen einer Excel-Tabelle	428
19.3.3	Eintragen von Daten in eine Excel-Tabelle	430
19.3.4	Geschwindigkeit der Datenspeicherung	432
19.3.5	Erstellen von Makros zum Umwandeln einer Tabelle in eine Grafik	433
19.3.6	Aufruf von Makros in LabVIEW mit Hilfe von ActiveX	437
19.3.7	Erhöhung der Geschwindigkeit	438
19.3.8	Schreiben mehrerer Dateien	440
19.4	Microsoft-Datenbank Access	444
19.4.1	Einführung	444
19.4.2	Verbindung mit der Datenbank	445
19.4.3	SQL	447
19.4.4	Verwendung von SubVIs	448
20	Internet, Server und Client	449
20.1	Allgemeine Bemerkungen zum Internet	449
20.1.1	Ethernet	449
20.1.2	Ethernet-Karten, MAC- und IP-Adresse	450
20.1.3	TCP/IP-Protokoll	450
20.2	Einfaches LabVIEW-Beispiel: Ping	452
20.3	Programmieren mit DataSocket	453
20.4	Programmieren mit TCP/IP	456
20.4.1	Server und Client	456
20.4.2	Beispiel für die Übertragung von Sinusdaten über TCP/IP	456

20.5	TCP/IP-Server und mehrere Clients.....	459
20.5.1	Vorüberlegungen	459
20.5.2	TCP/IP-SubVIs	460
20.5.3	Schreiben über TCP/IP	460
20.5.4	Lesen über TCP/IP	461
20.5.5	Anwendungsbezogene Fehlerbehandlung	462
20.6	Server und Client als Zustandsautomaten.....	462
20.7	Client/Server-Anwendung	465
20.7.1	Client-Anwendung	465
20.7.2	Server mit einem Client	470
20.7.3	Server mit mehreren Clients	472
21	Compact RIO-System und FPGA	476
21.1	Definition.....	476
21.2	Installation	478
	Schritt 1: Software-Installation auf dem PC	478
	Schritt 2: Zusammenstellen der cRIO-Hardware	479
	Schritt 3: Zuweisung einer IP-Adresse zum cRIO-System	479
	Schritt 4: Installation weiterer Software auf dem cRIO-System	482
	Schritt 5: Verbindung eines PC mit einem cRIO-System im Netz	482
21.3	Programmierbeispiele für FPGA	483
21.3.1	Beispiel zur Digitalausgabe	484
21.3.2	Beispiel eines Zählers	489
21.3.3	FPGA-Anwendung: Ermittlung eines Frequenzganges	491
21.3.4	Umgebungsvariablen	502
21.3.4.1	Projekt 'Shared_Einzeln'	502
21.3.4.2	Projekt 'Shared_Netzwerk'	504
21.3.4.3	Projekt 'Shared_cRIO'	508
21.3.5	FPGA-Anwendungen auf dem cRIO-9014 ohne PC-Unterstützung	510
21.3.5.1	Projekt 'RIO_MOD1_Switch'	510
21.3.5.2	Projekt 'RIO_User1_Switch'	513
21.3.5.3	Umstellung des cRIO-Systems von einem Standalone-Projekt zum nächsten.....	515
22	XControls	517
22.1	Unterschied zu einfachen Ctls.....	517
22.2	Anzeige der Flugbahn eines Steines	517
22.3	Erstellen eines XControls	519
22.3.1	Allgemeines Rezept	519
22.3.2	Beispiel XControl_Pfeil.xctl.....	521
22.3.3	Eigenschaften in einem XControl	527
22.3.4	Bedeutung der Rahmen [1] bis [4] im Fassaden-VI	530
22.3.5	Weitere Verbesserungen	536
22.4	XControl zur Erstellung von Symbolleisten	538
22.4.1	Zustand der Symbolleiste	539
22.4.2	Funktionen der Symbolleiste	540
22.4.2.1	Symbole hinzufügen.....	540
22.4.2.2	Alle Symbole löschen	541

22.4.2.3	Rückmeldung des Symbols, das unter dem Mauszeiger liegt	541
22.4.2.4	Anpassung des Erscheinungsbilds an eigene Bedürfnisse.....	545
22.4.3	Leistungsmerkmal 'Status für Speichern umwandeln'	545
Literatur		547
Index		549