

Einführung in LabVIEW

Bearbeitet von
Wolfgang Georgi, Philipp Hohl

6., erweiterte Auflage 2015. Buch. 596 S. Kartoniert

ISBN 978 3 446 44272 6

Format (B x L): 16,4 x 24,4 cm

Gewicht: 1120 g

[Weitere Fachgebiete > EDV, Informatik > Programmiersprachen: Methoden > Funktionale, logische, parallele und visuelle Programmierung](#)

schnell und portofrei erhältlich bei

The logo for beck-shop.de features the text 'beck-shop.de' in a bold, red, sans-serif font. Above the 'i' in 'shop' are three red dots of increasing size. Below the main text, 'DIE FACHBUCHHANDLUNG' is written in a smaller, red, all-caps, sans-serif font.

beck-shop.de
DIE FACHBUCHHANDLUNG

Die Online-Fachbuchhandlung beck-shop.de ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.



Leseprobe

Einführung in LabVIEW

ISBN (Buch): 978-3-446-44272-6

ISBN (E-Book): 978-3-446-44407-2

Weitere Informationen oder Bestellungen unter

<http://www.hanser-fachbuch.de/978-3-446-44272-6>

sowie im Buchhandel.

Vorwort zur sechsten Auflage

Dieses Lehrbuch führt wie in der ersten Auflage in das Programmieren mit LabVIEW ein. Damals mussten wir noch erklären, dass sich LabVIEW für messtechnische Anwendungen eignet und in der Industrie mehr und mehr geschätzt wird. Heute ist das allgemein bekannt, auch, dass es sich bei dieser von der Firma National Instruments entwickelten Software um ein Werkzeug handelt, das sich weit über die Messtechnik hinaus vorteilhaft anwenden lässt.

Um einen guten Lernerfolg zu erzielen, sollte der Leser möglichst viele Beispiele und Übungen am PC durcharbeiten. Alle Beispielprogramme wurden für die LabVIEW-Version 2014 geschrieben. Wir setzen also voraus, dass der Leser die Version 2014 von LabVIEW installiert hat. Frühere Versionen wie LabVIEW 2009 oder LabVIEW 8.0 sind im Kern recht ähnlich. Programme, die mit diesen Versionen erstellt werden, laufen auch unter der Version 2014. Doch trifft das Umgekehrte naturgemäß nicht zu, weil jede neue Version auch neue Möglichkeiten bietet. Weiter wird vorausgesetzt, dass der PC unter einem der Betriebssysteme Windows 7 oder Linux arbeitet.

Das Buch wendet sich an Studierende, aber auch an Ingenieure, die unter dem Stichwort "Lebenslanges Lernen" versuchen, neueren Trends in der Industrie zu folgen.

Das Lehrbuch gliedert sich in vier Teile:

Teil I: Grundlagen des Programmierens in LabVIEW

Teil II: Technische Anwendungen

Teil III: Kommunikation

Teil IV: Fortgeschrittene Techniken

In Teil I werden Installation und Aufruf von LabVIEW, grundlegende Arbeitsmittel wie (Front-)Panel, Diagramm, Paletten für Eingabe/Ausgabe, Funktionen und Werkzeuge behandelt, ferner Konzepte von LabVIEW, Datentypen, Grundlagen der Programmierung und Visualisierungstechniken.

Teil II befasst sich mit Anwendungen wie Fouriertransformation, Filterung, Lösen von Differenzialgleichungen und Differenzialgleichungssystemen in der Technik.

Teil III geht auf die Kommunikation ein. Hier sind zwei Aspekte von Bedeutung:

- Externe Kommunikation mit anderen Geräten und Rechnern, z.B. über USB, Datenerfassungskarten, TCP/IP (Internetanbindung),
- Kommunikation mit anderen Softwarepaketen, z.B. mit der Erstellung und Anbindung selbst geschriebener C-Module.

Teil IV befasst sich mit Zustandsautomaten, mit der objektorientierten Programmierung (OOP), mit Tabellenkalkulation (Excel) und Datenbankanwendungen (Access), mit dem

Datenaustausch über Intra- und Internet, mit dem Compact RIO-System von National Instruments samt FPGA-Programmierung und Verwendung von XControls und XNodes.

Neu hinzugekommen sind in der sechsten Auflage Kapitel 23 und 24, in denen wir Scripting und das bisher recht unzugängliche Konstrukt der XNodes behandeln.

Weitgehend neu haben wir das Kapitel 20 gestaltet. Abschnitt 20.5 wurde ins Internet (<http://www.geho-labview.de>) gestellt und ersetzt durch das seit LabVIEW 2013 neu eingeführte Konzept des Webdienstes.

Mehr als in der fünften Auflage haben wir verschiedene Kapitel ergänzt mit dem Ziel, dem Leser anhand von Beispielen 'guten Programmierstil' zu vermitteln.

Das Lehrbuch ist trotz seines dadurch stark gewachsenen Umfangs immer noch knapp gehalten. Es kann also nur eine Einführung sein, die allerdings versucht, die wichtigsten Aspekte von LabVIEW zu berücksichtigen. Bei einem so umfangreichen Softwaresystem wie LabVIEW sind jedoch Lücken unvermeidlich. Hier verweisen wir auf weiterführende Literatur, auf die Veröffentlichungen von National Instruments, auf User Groups und auf das Internet ganz allgemein. Diese Hinweise werden wir in den verschiedenen Kapiteln des Lehrbuchs noch vertiefen.

Wir bedanken uns ganz herzlich bei allen, die uns geholfen haben:

Besonders bei Herrn Thakur Adhikari, der sich in seiner Masterthese mit XNodes befasst und damit regelungstechnische Anwendungen entwickelt hat. Ohne seine gründlichen Recherchen zu den in der Literatur nur unzulänglich dokumentierten XNodes wäre es nicht möglich gewesen, Kapitel 24 noch in dieser Auflage herauszugeben.

Schließlich danken wir allen Lesern, die mit ihren Fragen Verständnisprobleme deutlich gemacht und uns damit zur Verbesserung mancher Erklärung angeregt haben. Wir sind auch weiterhin für Anregungen und Kritik dankbar. Diese können Sie uns jetzt auch direkt über die Internetseite <http://www.geho-labview.de> übermitteln.

Weitere Informationen zu LabVIEW sowie die Downloads der Test- bzw. Studentenversionen finden Sie unter: www.ni.com/download-labview/d/

Dem Fachbuchverlag Leipzig, und hier besonders Frau Werner, Frau Jacob und Frau Kaufmann, danken wir für die gründliche Korrektur und ihre Ratschläge zur Gestaltung des Layouts.

Weingarten, Juni 2015
Wolfegg

W. Georgi
P. Hohl

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, Paletten	35
2.1 Einstellungen	35
2.1.1 Einstellungen von LabVIEW	35
2.1.2 Frontpanel.....	36
2.1.3 Blockdiagramm.....	37
2.1.4 Ausrichtungsgitter	39
2.1.5 Wiederherstellungen	39
2.2 Paletten	39
2.2.1 Werkzeugpalette (Tools Palette)	40
2.2.2 Eingabe-/Ausgabe-Elemente.....	41
2.2.3 Funktionenpalette.....	44
2.2.4 Palette konfigurieren	47
3 Programmstrukturen	48
3.1 Strukturiertes Programmieren.....	48
3.2 Sequenz	50
3.3 Case-Struktur	53

3.4	Schleifen	57
3.5	Guter Programmierstil.....	61
4	Datentypen	63
4.1	Numerische Datentypen	63
4.1.1	Kontextmenü: 'Darstellung'	63
4.1.2	Kontextmenü: 'Anzeigeformat...'	64
4.2	Boolesche Datentypen	66
4.3	String und Pfad	68
4.4	Arrays	71
4.4.1	Definition und Initialisierung eines 1-dimensionalen Arrays	71
4.4.2	Definition und Initialisierung eines 2-dimensionalen Arrays	73
4.4.3	Array erstellen.....	74
4.4.4	Rechnen mit Arrays: Addition	75
4.4.5	Rechnen mit Arrays: Multiplikation.....	76
4.4.6	Steuerung von For-Schleifen mit Arrays	77
4.4.7	Behandlung einzelner Arrayelemente	79
4.5	Cluster	81
4.5.1	Erzeugung eines Clusters	82
4.5.2	Clusterwerte ändern	83
4.5.3	Aufschlüsseln eines Clusters	85
4.5.4	Umordnen der Elemente eines Clusters	86
4.5.5	Cluster-Arrays	87
4.6	Ring & Enum.....	88
4.7	Datentyp FXP	90
4.8	Datentyp Variant	92
4.9	Guter Programmierstil.....	94
5	Unterprogramme und Typdefinitionen	96
5.1	Wozu Unterprogramme (SubVIs)?	96
5.2	Erstellen von Unterprogrammen	97
5.2.1	Einführendes Beispiel	97
5.2.2	Weitere Hinweise für die Erstellung eines Unterprogramms	100
5.2.3	Einstellungen für Programme und Unterprogramme	102
5.2.4	Erstellen von Unterprogrammen mit internem Zustand	104
5.2.5	Erstellen von polymorphen Unterprogrammen.....	105
5.3	Aufruf von Unterprogrammen	108
5.3.1	Statische Bindung.....	108
5.3.2	Dynamische Bindung	109
5.3.2.1	VI-Referenz öffnen und schließen.....	109
5.3.2.2	Aufruf eines VI über seine Referenz	110
5.3.2.3	Beispiel für den SubVI-Austausch während der Laufzeit.....	112
5.3.2.4	Rekursiver Aufruf von Unterprogrammen	112
5.3.2.5	Testen (Debugging) von ablaufinvarianten SubVIs	113

5.4	Typdefinitionen	115
5.4.1	Beispiel einer Typdefinition für Enum-Variablen.....	115
5.4.2	Beispiel einer Typdefinition für Registerkarten	117
5.5	Guter Programmierstil	118
5.5.1	Vereinfachung durch Unterprogramme und Typdefinitionen	118
5.5.2	Aussagekräftige Symbole (Icons).....	120
5.5.3	Anordnung häufig verwendeter Elemente.....	120
5.5.4	Kommentierung der Elemente und Funktionen eines VI	120
5.5.5	Detaillierte Hilfe.....	122
6	Prozessvisualisierung	123
6.1	OOP-Konzepte	123
6.2	Eigenschafts- und Methodenknoten	123
6.3	Grafische Ausgabe	127
6.3.1	Chart (Signalverlaufsdiagramm).....	127
6.3.1.1	Darstellung einer Sinuskurve	127
6.3.1.2	Darstellung von zwei oder mehr Kurven in einem Chart	129
6.3.1.3	Legende zu einem Chart oder Graphen	130
6.3.1.4	Skalierung der Ordinate in einem Chart.....	131
6.3.2	Graph (Signalverlaufsgraph)	133
6.3.2.1	Darstellung einer Sinuskurve	133
6.3.2.2	Darstellung von zwei oder mehr Kurven in einem Graphen	134
6.3.2.3	Skalierung der Abszisse in einem Graphen	136
6.3.3	XY-Graph.....	138
6.3.3.1	Darstellung einer Relation im XY-Graphen	139
6.3.3.2	Darstellung mehrerer Relationen in einem XY-Graphen	140
6.3.4	Signalverlauf	141
6.4	Express-VIs, Programmierstil	146
6.4.1	Express-VI zur Erzeugung von Kurven	146
6.4.2	Express-VI zur Erstellung von Berichten.....	147
7	Referenzen, Fehlerfunktionen	149
7.1	Einführendes Beispiel.....	149
7.1.1	Vertauschung von zwei Variablenwerten	149
7.1.2	Referenzen auf Bedien- und Anzeigeelemente	150
7.1.3	Lösung des Vertauschungsproblems	151
7.2	Vererbung.....	152
7.2.1	Eigenschaften der Basisklasse	154
7.2.2	Eigenschaften von abgeleiteten Klassen.....	154
7.3	Fehlerfunktionen	156
7.3.1	Fehlermeldungen mit oder ohne Dialog.....	156
7.3.2	Wo findet man wichtige Fehlerelemente und Fehlerfunktionen?	157
7.3.3	Verschiedene Fehlerarten	158
7.3.3.1	Standardfehlerleitung.....	158
7.3.3.2	Funktionen ohne oder mit vereinfachter Fehlerleitung	158

7.3.4	Ausgang aus While-Schleifen	160
7.3.5	Erzwingung von sequenziellem Ablauf	161
8	Datentransfer von und zur Festplatte	162
8.1	Dateifunktionen	162
8.1.1	Allgemeines zur Speicherung von Dateien.....	162
8.1.2	Palette Dateifunktionen	164
8.1.3	Einführendes Beispiel	165
8.1.4	Modifiziertes Beispiel	166
8.1.5	Beispiel: Anlegen einer Protokolldatei	167
8.1.6	Überschreiben ohne Warnung	167
8.2	Pfade	168
8.2.1	Pfadkonstanten	168
8.2.2	Pfadkonstante 'Standardverzeichnis'.....	169
8.2.3	'Standardverzeichnis' ändern	170
8.2.4	'Standarddatenverzeichnis' ändern	171
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	176
8.4.2	Diadem.....	179
8.4.3	ZIP-Dateien	179
8.4.4	Konfigurationsdateien	181
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	Erstellung einer EXE-Datei	192
9.3.2	EXE-Datei auf einem Rechner ohne LabVIEW-System.....	194
9.4	Strukturen zur Programmentwicklung.....	197
9.4.1	Deaktivierungsstrukturen.....	197
9.4.2	Debug-Einstellung in der Projektverwaltung	199
9.5	LabVIEW-Bibliotheken.....	200
9.6	Umwandeln von LLB-Bibliotheken	202

9.7	Einbindung von C-Funktionen unter Windows	204
9.7.1	Reihe in C#	205
9.7.2	Reihe in C++	209
9.7.3	Reihe mit MathScript	213
9.8	Hilfen zu LabVIEW	213
9.9	Schnelleinfügeleiste (Quickdrop)	215
9.10	Der VI Package Manager	217
9.10.1	Verwalten der LabVIEW-Entwicklungsumgebung	218
9.10.2	Eigenes Paket erstellen	220

Teil II: Technische Anwendungen

221

10 Fouriertransformation

222

10.1	Zeit- und Frequenzbereich	222
10.1.1	Die reelle Fouriertransformation	223
10.1.2	Darstellung der Fourierkoeffizienten \underline{c}_k in LabVIEW	226
10.2	Diskrete Fouriertransformation	229
10.2.1	Satz von Shannon	229
10.2.2	Aliasing	231
10.2.3	Frequenzauflösung	232

11 Filterung

234

11.1	Filtertypen	234
11.1.1	Ideale und reale Filter	234
11.1.2	Beispiel eines digitalen Filters	235
11.2	LabVIEW-Filterfunktionen	238
11.3	Filterung im Frequenzbereich	240
11.3.1	Idee der Filterung im Frequenzbereich	240
11.3.2	Die inverse Fouriertransformation in LabVIEW	240
11.3.3	Beispiel eines Tiefpasses	241

12 Differenzialgleichungen

243

12.1	Lösen mit LabVIEW-ODE-Funktionen	243
12.2	Lösen nach dem Analogrechnerprinzip	245
12.2.1	Blockdiagramm-Darstellung	245
12.2.2	Vereinfachungen	248
12.3	Genauigkeit numerischer Verfahren	250

13 Systeme von Differenzialgleichungen

253

13.1	Systeme gewöhnlicher Differenzialgleichungen	253
13.2	Gekoppeltes Feder-Masse-System	253
13.2.1	Lösung mit eingebauter ODE-Funktion	254
13.2.2	Lösung mit Blockdiagramm wie in MATLAB®	255
13.3	Umwelt und Tourismus	257

14	Parallelverarbeitung, Laufzeiten, Ereignisse	260
14.1	Einführendes Beispiel	260
14.2	Grundbegriffe der Parallelverarbeitung	262
14.2.1	Multiprocessing, Multitasking, Multithreading	262
14.2.2	Synchronisierung von Prozessen	263
14.3	Parallelverarbeitung unter LabVIEW	264
14.3.1	Erzeugen von Ressourcen für die Prozesskommunikation	265
14.3.2	Freigabe von Ressourcen der Prozesskommunikation	267
14.3.3	Zeitbegrenzung Ressource schont Prozessor	268
14.4	Prozess-Synchronisierung ohne Datenaustausch	268
14.4.1	Occurrences	268
14.4.2	Semaphor	269
14.4.3	Rendezvous	271
14.5	Prozess-Synchronisierung mit Datenaustausch	272
14.5.1	Melder-Operationen	272
14.5.2	Queue-Operationen	273
14.6	Globale Variablen	274
14.7	Laufzeitprobleme und ihre Behandlung	275
14.7.1	Laufzeitprobleme bei lokalen Variablen	275
14.7.2	Laufzeitprobleme bei globalen Variablen	278
14.8	Ereignisgesteuerte Programmierung	279
14.8.1	Frontpanel-Ereignisse	279
14.8.2	Wertänderungs-Ereignisse	284
14.8.3	Gefilterte Ereignisse	285
14.9	Zeitschleifen	287
Teil III: Kommunikation		289
15	Serielle Eingabe/Ausgabe	290
15.1	RS-232	290
15.2	Programmierung der RS-232 in LabVIEW	292
15.3	Die USB-Schnittstelle	295
15.4	Feld-Bus, CAN-Bus	299
15.4.1	CAN-Protokoll	299
15.4.2	CAN-Interface	301
15.4.3	CANopen-Protokoll, ZILA-Sensor	302
15.4.4	CAN-Bus mit Laptop und zwei Sensoren	304
15.4.5	XNET-System von National Instruments	304
15.5	Der byte-serielle GPIB-Bus	315
16	Datenerfassungsgeräte	317
16.1	Datenerfassungskarten/Datenerfassungsgeräte	317
16.2	Allgemeines	318
16.2.1	Treiber, MAX (Measurement and Automation Explorer)	318
16.2.2	Physikalische und virtuelle Kanäle, Task	324
16.2.3	Programmierung von Datenerfassungs-VIs, simulierte Geräte	325

16.2.4	Programmierung von VIs zur Analogausgabe	330
16.2.5	Programmierung von VIs zum Digital-I/O	331
16.2.6	Programmierung mit Hilfe des DAQ-Assistenten	331
16.2.7	Programmatische Task-Erstellung	333
16.3	USB-Gerät NI USB-6251	334
16.3.1	Begriffe 'differenziell', 'RSE' und 'NRSE'	334
16.3.2	Zwei Analogsignale mit der NI USB-6521 lesen	336
16.3.3	Triggern mit NI USB-6521	337
16.3.4	Streaming mit NI USB-6521	338
16.4	Ältere Datenerfassungskarten/-geräte	345
16.5	TEDS	345
16.6	IVI-Gerät NI USB-513	349

Teil IV: Fortgeschrittene Techniken

355

17 Professionelle Programmentwicklung

356

17.1	Sequenzstruktur	356
17.2	Zustandsautomaten.....	357
17.2.1	Notation für Zustandsautomaten	358
17.2.2	Umsetzung Zustandsdiagramm → LabVIEW-Programm	359
17.2.2.1	Strings für die Zustandsauswahl	360
17.2.2.2	Enum für die Zustandsauswahl.....	362
17.3	Münzautomat	363
17.4	Münzautomat mit Queues und Ereignisstrukturen.....	372
17.5	Programmierhilfen	376
17.5.1	Arbeiten mit vorgefertigten Strukturen (Templates)	376
17.5.2	Beurteilung Programmeffizienz und geeignete Werkzeuge dazu	376

18 Objektorientierte Programmierung

380

18.1	Warum objektorientiert?	380
18.2	Erstes Beispiel zur objektorientierten Programmierung	383
18.2.1	Bildung einer Klasse.....	383
18.2.2	Private Eigenschaften der Klasse	384
18.2.3	Methoden der Klasse	385
18.3	Weitere Beispiele zur OOP.....	389
18.3.1	Vererbung	389
18.3.2	Polymorphie.....	393
18.3.3	Modulaustausch	397
18.4	Schutz einer Klassenbibliothek	405

19 LabVIEW: Tabellenkalkulation, Datenbanken

408

19.1	Schreib-/Lesebefehle zur Tabellenkalkulation	408
19.2	Allgemeines über ActiveX.....	410
19.2.1	ActiveX-Container in LabVIEW	411
19.2.2	ActiveX in LabVIEW zur Steuerung von Anwendungen.....	413

19.3	Beispiele zur Anwendung auf Excel	414
19.3.1	Öffnen und Schließen von Excel.....	415
19.3.2	Sichtbarmachen einer Excel-Tabelle	416
19.3.3	Eintragen von Daten in eine Excel-Tabelle	418
19.3.4	Geschwindigkeit der Datenspeicherung	420
19.3.5	Erstellen von Makros zum Umwandeln einer Tabelle in eine Grafik	421
19.3.6	Aufruf von Makros in LabVIEW mit Hilfe von ActiveX.....	424
19.3.7	Erhöhung der Geschwindigkeit	425
19.3.8	Schreiben mehrerer Dateien	428
19.4	Microsoft-Datenbank Access	432
19.4.1	Einführung.....	432
19.4.2	Verbindung mit der Datenbank	433
19.4.3	SQL	435
19.4.4	Verwendung von SubVIs	436
20	Internet, Server und Client	437
20.1	Allgemeine Bemerkungen zum Internet	437
20.1.1	Ethernet.....	437
20.1.2	Ethernet-Karten, MAC- und IP-Adresse	438
20.1.3	TCP/IP-Protokoll.....	438
20.2	Einfaches LabVIEW-Beispiel: Ping.....	439
20.3	Programmieren mit DataSocket.....	441
20.4	Programmieren mit TCP/IP.....	443
20.4.1	Server und Client.....	443
20.4.2	Beispiel für die Übertragung von Sinusdaten über TCP/IP.....	444
20.5	Webdienste	447
20.5.1	Grundbegriffe	447
20.5.2	Struktur der Webdienstkommunikation	448
20.5.3	Erstes einfaches Beispiel.....	448
20.5.4	Zweites einfaches Beispiel	453
20.5.5	Drittes Beispiel	456
20.5.6	Dreiecksberechnung	457
20.5.7	Webserver im Internet	462
20.5.7.1	Firmeninternes Netz.....	462
20.5.7.2	Aufruf im Internet	463
21	Compact RIO-System und FPGA	464
21.1	Definition	464
21.2	Installation.....	466
	Schritt 1: Software-Installation auf dem PC.....	466
	Schritt 2: Zusammenstellen der cRIO-Hardware.....	467
	Schritt 3: Zuweisung einer IP-Adresse zum cRIO-System.....	467
	Schritt 4: Installation weiterer Software auf dem cRIO-System	470
	Schritt 5: Verbindung eines PC mit einem cRIO-System im Netz	470

21.3	Programmierbeispiele für FPGA	471
21.3.1	Beispiel zur Digitalausgabe	472
21.3.2	Beispiel eines Zählers	477
21.3.3	FPGA-Anwendung: Ermittlung eines Frequenzganges	479
21.3.4	Umgebungsvariablen	489
21.3.4.1	Projekt 'Shared_Einzeln'	490
21.3.4.2	Projekt 'Shared_Netzwerk'	492
21.3.4.3	Projekt 'Shared_cRIO'	495
21.3.5	FPGA-Anwendungen auf dem cRIO-9014 ohne PC-Unterstützung	497
21.3.5.1	Projekt 'RIO_MOD1_Switch'	497
21.3.5.2	Projekt 'RIO_User1_Switch'	499
21.3.5.3	Umstellung des cRIO-Systems von einem Standalone-Projekt zum nächsten	502
22	XControls	504
22.1	Unterschied zu einfachen Ctls	504
22.2	Anzeige der Flugbahn eines Steines	504
22.3	Erstellen eines XControls	506
22.3.1	Allgemeines Rezept	506
22.3.2	Beispiel XControl_Pfeil.xctl	508
22.3.3	Eigenschaften in einem XControl	514
22.3.4	Bedeutung der Rahmen [1] bis [4] im Fassaden-VI	517
22.3.5	Weitere Verbesserungen	523
22.4	XControl zur Erstellung von Symbolleisten	525
22.4.1	Zustand der Symbolleiste	526
22.4.2	Funktionen der Symbolleiste	527
22.4.2.1	Symbole hinzufügen	527
22.4.2.2	Alle Symbole löschen	528
22.4.2.3	Rückmeldung des Symbols, das unter dem Mauszeiger liegt	528
22.4.2.4	Anpassung des Erscheinungsbilds an eigene Bedürfnisse	532
22.4.3	Leistungsmerkmal 'Status für Speichern umwandeln'	532
23	LabVIEW VI-Skripte	534
23.1	Was sind VI-Skripte	534
23.2	Die VI-Skripte Funktionen in der Palette anzeigen	534
23.3	Die VI-Skripte Funktionen	536
23.3.1	Neues VI	536
23.3.2	Neues VI-Objekt	537
23.3.3	VI-Objektreferenz öffnen	539
23.3.4	Abstand des neuen VI-Objekts von Referenzobjekt	539
23.3.5	GObjects suchen	541
23.3.6	GObject-Beschriftung abfragen	542
23.3.7	Klassenhierarchie mittels Klassennamen ermitteln	542
23.3.8	Weiterführende Informationen	544
23.4	Wo werden VI-Skripte eingesetzt?	545

23.5	Modifizierung der Projektvorlage "Leeres VI"	546
23.6	Erstellen eines Quickdrop Plugins mit VI-Skripting.....	550
24	XNodes	553
24.1	Einführung	553
24.2	Regelungstechnische Anwendung	554
24.3	Aufbau eines XNodes	556
24.4	Wie bildet man einen XNode?.....	556
24.4.1	Vorbereitende Überlegungen	556
24.4.2	Programmierung von NeueKuh.xnode.....	561
24.4.2.1	Template-VI	561
24.4.2.2	Ability-VIs	562
24.5	Wie ändert man einen XNode?	571
24.6	XNodes in der Funktionspalette speichern.....	571
Literatur		575
Index		577

2 Einstellungen, Paletten

Lernziele

1. Einige wichtige Voreinstellungen für die Gestaltung von Panel und Diagramm kennen und nach persönlichem Bedarf verändern können.
2. Mehr Details über Werkzeug-, Funktionen- und Elemente-Palette kennen.
3. Palettenansicht an Benutzerwünsche anpassen können.

Abschnitt 2.3 kann von eiligen Lesern übersprungen werden.

2.1 Einstellungen

Das Beispiel in Kapitel 1 wurde unter einer Reihe von Bedingungen entwickelt, die dem Anwender zunächst vermutlich unbewusst geblieben sind. Sie betreffen das Erscheinungsbild der Terminals als quadratische Symbole oder 'Icons' im Diagramm, das Raster auf dem Panel während der Erstellung der Bedienoberfläche und vieles andere. Die Parameter dafür sind voreingestellte Standardwerte ('Default-Werte'), doch kann sie der Anwender ändern.

Dazu gibt es zwei Wahlmöglichkeiten:

1. 'Werkzeuge' – 'Optionen...' (Einstellungen von LabVIEW)
2. 'Datei' – 'VI-Einstellungen...' (Einstellungen des aktuellen VI)

Wir befassen uns hier mit der ersten Wahlmöglichkeit. Öffnet man vom Panel oder vom Diagramm aus das Optionsangebot, erhält man (voreingestellt) eine Ansicht nach Bild 2.1.

2.1.1 Einstellungen von LabVIEW

Die Optionen in Bild 2.1 zeigen verschiedene Möglichkeiten der Parameteränderung. Man kann unter anderem die Einstellungen beeinflussen für:

- Frontpanel,
- Blockdiagramm,
- Elemente- und Funktionenpalette,
- Pfade usw.

Das in älteren LabVIEW-Versionen vorhandene 'Ausrichtungsgitter' ist bei LabVIEW 2014 zum Teil unter 'Frontpanel', zum anderen Teil unter 'Blockdiagramm' zu finden.

Wir wollen uns hier näher mit dem Frontpanel und dem Blockdiagramm befassen.

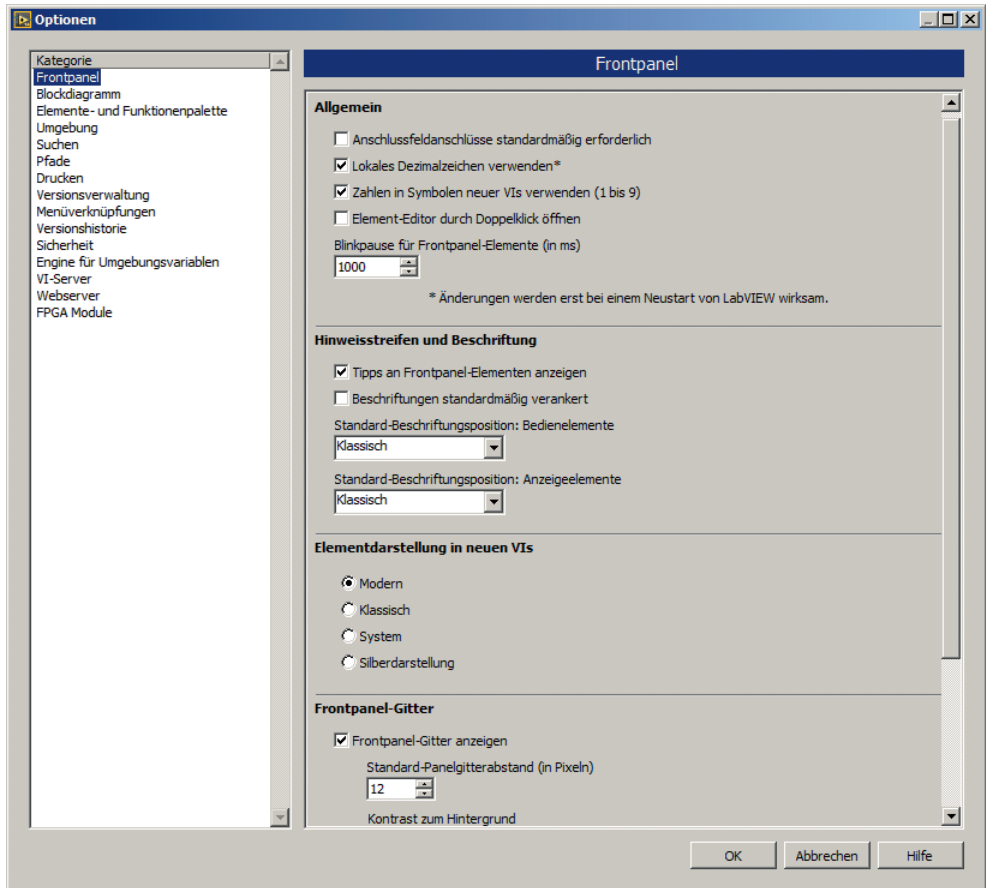


Bild 2.1 Optionen zur Gestaltung der verschiedenen Kategorien, hier Frontpanel

2.1.2 Frontpanel

Die generellen Frontpanel-Einstellungen erreicht man mit 'Werkzeuge' – 'Optionen...' – 'Frontpanel'. Bild 2.1 zeigt das zugehörige Fenster.

Interessant ist hier die zweite Zeile rechts mit 'Lokales Dezimalzeichen verwenden*'. Ist sie angekreuzt, nutzt LabVIEW dasjenige Zeichen als Dezimalpunkt, das unter Windows in der Systemsteuerung bei 'Ländereinstellungen' – 'Zahlen' eingetragen wurde. Ist dort der Punkt das Dezimaltrennzeichen, übernimmt LabVIEW diese Einstellung. Ist das Komma Dezimaltrennzeichen, geschieht Entsprechendes.

Ist aber die Zeile 'Lokales Dezimalzeichen verwenden*' **nicht** angekreuzt, arbeitet LabVIEW generell mit dem Dezimalpunkt. Das kann zu Fehlern führen, wenn ein LabVIEW-

Programm z.B. mit Microsoft Excel zusammenwirken soll. Haben beide Programmsysteme verschiedene Dezimaltrennzeichen, lesen sie wechselseitig die Daten falsch.

Im Frontpanel kann auch der Stil der Elemente für Ein- und Ausgabe voreingestellt werden. Der Standardwert ist 'Modern', wir nutzen in diesem Buch aber auch die 'Silberdarstellung'.

2.1.3 Blockdiagramm

Wählt man die Kategorie 'Blockdiagramm', erhält man eine Übersicht, deren oberer Teil in Bild 2.2 dargestellt ist. Dort sieht man rechts oben unter 'Allgemein' die Eintragung 'Frontpanel-Elemente als Symbole darstellen'. Sie ist im vorliegenden Fall **nicht** markiert. Das bedeutet, dass per Voreinstellung die Terminals als kleine Rechtecke dargestellt werden, wie wir das z.B. von Bild 1.13 her kennen. Markiert man dagegen diese Option, werden die Terminals als größere quadratische Symbole (Icons, siehe Bild 2.3) dargestellt. Man kann trotzdem später im Blockdiagramm von einer Darstellung zur anderen wechseln, indem man individuell für jedes Terminal das Kontextmenü aufruft und bei 'Als Symbol anzeigen' die Markierung setzt oder entfernt.

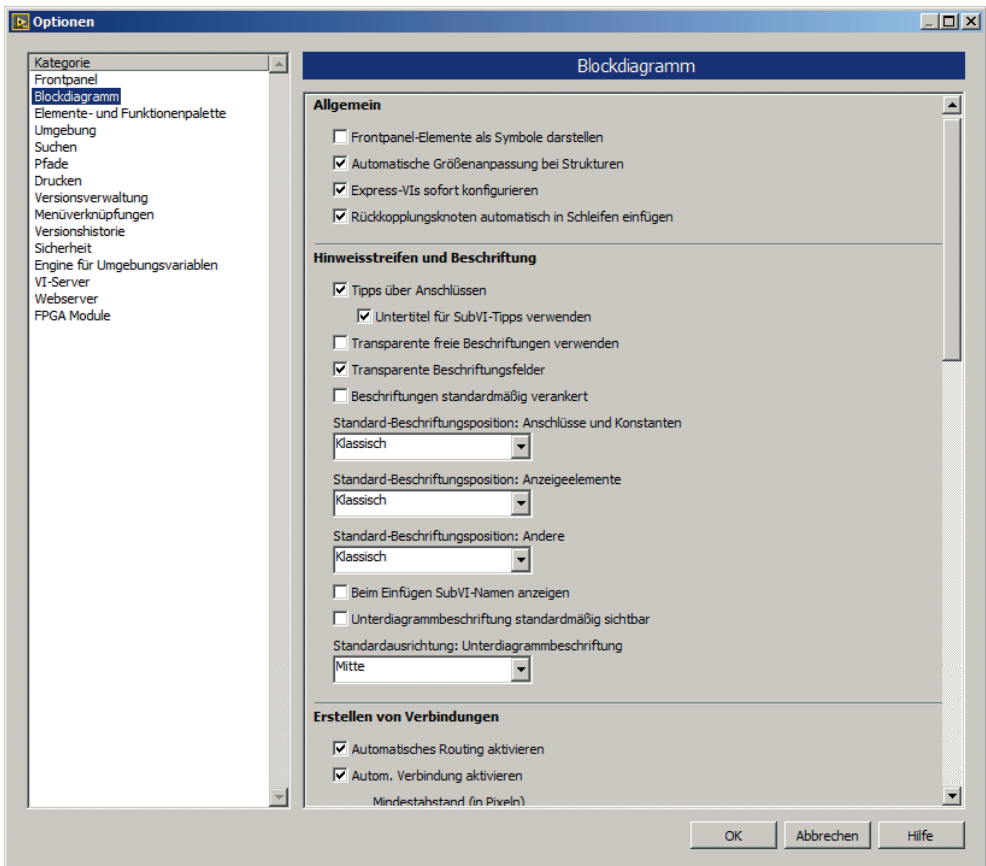


Bild 2.2 Optionen zur Gestaltung des Blockdiagramms

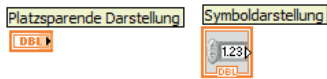


Bild 2.3 Terminaldarstellungen

Welche Art der Darstellung man wählt, ist weitgehend Geschmackssache. Zu bedenken ist allerdings, dass die quadratischen Symbole mehr Platz im Diagramm brauchen. Bereits gesetzte Terminals sind von einer generellen Umstellung nicht betroffen. Sie behalten ihr ursprüngliches Aussehen. Weiter sind folgende Optionen interessant:

- 'Transparente Beschriftungsfelder' unter 'Hinweisstreifen und Beschriftung' in Bild 2.2 ist **angekreuzt**. Das bedeutet, dass Kommentare, die man mit der A-Funktion der Werkzeugpalette schreibt, **keinen Rand** erhalten. Auch die Namen von Eingabe- und Ausgabefeldern werden zunächst randlos dargestellt. Ist dagegen das Kontrollkästchen **nicht angekreuzt**, wird im Diagramm automatisch ein **schwarzer Rand** gezeichnet, auf dem Panel erscheint die Schrift als Relief. Man kann das individuell für jeden Kommentar und jedes Terminal mit dem Farbpinsel für Vordergrund und Hintergrund in der Werkzeugpalette ändern. **Transparenz** erhält man, wenn man bei der Farbwahl das große T wählt.
- 'Verbindungspunkte an Kreuzungen' unter 'Erstellen von Verbindungen' sollte **markiert** sein. Das sorgt dafür, dass man Verbindungslinien im Diagramm, die sich nur kreuzen, von solchen unterscheiden kann, die miteinander verbunden sind. Im letzteren Fall werden dort nämlich dicke Punkte gesetzt. Siehe dazu Bild 2.4 und Bild 2.5.

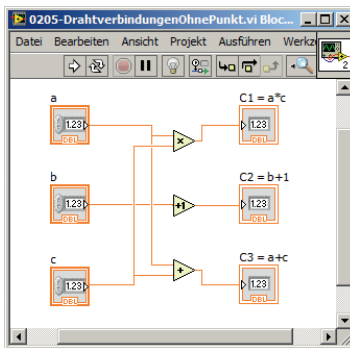


Bild 2.4 Diagramm ohne Verbindungspunkte an allen Draht-Kreuzungen und jeweils zwei Transparenzeinstellungen für Beschriftungen

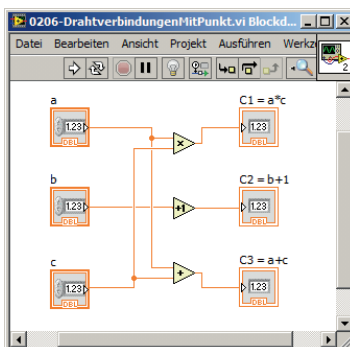


Bild 2.5 Diagramm mit Verbindungspunkten an Draht-Kreuzungen, sofern die Drähte miteinander verbunden sind

- 'Automatisches Routing aktivieren' unter 'Erstellen von Verbindungen' heißt, dass Verbindungen zwischen zwei Funktionen nicht in jedem Fall vom Programmierer gezogen werden müssen, sondern sich bei entsprechend geringer Distanz von selbst bilden (wenn auch nicht immer in sinnvoller Weise).

2.1.4 Ausrichtungsgitter

Bereits ab LabVIEW 7.0 erhielt das Frontpanel während des Editierens (Programm erstellen) ein Gitter. Es dient der besseren Ausrichtung der Eingabe- und Ausgabeelemente. Auch hier lassen sich einige Parameter ändern, siehe dazu Bild 2.1 und Bild 2.2.

Man kann auf die Anzeige des Frontpanel-Gitters verzichten, andererseits lässt sich aber auch zusätzlich ein Gitter auf dem Blockdiagramm erzeugen. All das trifft nur für die Editierphase zu, d.h. für die Entwicklung des Programms. Zur Laufzeit verschwindet das Gitter automatisch. Man erhält das Raster, wenn man unter 'Frontpanel-Gitter' nach Bild 2.1 'Frontpanel-Gitter anzeigen' markiert. Damit die Ausrichtung am Raster wirksam wird, muss man zusätzlich 'Ausrichtung an Gitter aktivieren' ankreuzen. Entsprechend kann man im 'Blockdiagramm' nach Bild 2.2 unter 'Blockdiagrammgitter' markieren 'Blockdiagrammgitter anzeigen' bzw. 'Ausrichtung an Diagrammgitter aktivieren'. Ferner kann man mit 'Standard-Diagrammgitterabstand in Pixeln' die Feinheit des Gitters bestimmen sowie den Kontrast Gitter – Hintergrund erhöhen oder abschwächen.

Hat man ein Raster eingestellt, an dem sich die Elemente auf dem Panel ausrichten, gibt es im Zweifelsfall immer noch die Möglichkeit, ein nicht gut passendes Objekt in kleineren Schritten zu verschieben. Dazu das Objekt mit der Pfeiltaste der Werkzeugpalette anklicken und anschließend mit Hilfe der Pfeiltasten der PC-Tastatur verschieben. Drückt man gleichzeitig die Umschalttaste am PC, verschiebt man das Objekt schrittweise im Gitterabstand.

2.1.5 Wiederherstellungen

Unter 'Werkzeuge' – 'Optionen...' – 'Umgebung' ist unter 'Allgemein' die Zahl der Wiederherstellungen als 'Zulässige Rückgängigsschritte pro VI' auf 99 voreingestellt. Bedeutung: Bemerkt der Programmierer einen Fehler bei der Entwicklung eines VIs, kann er maximal 99 Programmierschritte mit 'Bearbeiten' – 'Rückgängig ...' bzw. mit dem Shortcut <Strg>+<Z> zurückgehen.

Auf zusätzliche Möglichkeiten, Parameter mit 'Datei' – 'VI-Einstellungen...' zu ändern, werden wir später eingehen.

2.2 Paletten

In Kapitel 1 wurden bereits die wichtigsten Paletten genannt, die man zum Programmieren in LabVIEW braucht. Sie sollen nun ausführlicher besprochen werden.

2.2.1 Werkzeugpalette (Tools Palette)

Beim Starten von LabVIEW erscheint die Werkzeugpalette automatisch, falls sie beim vorigen Aufruf bereits geöffnet war. Ist das nicht der Fall, holt man sie mit dem Aufruf der Menüs 'Ansicht' – 'Werkzeugpalette' vom Frontpanel oder vom Blockdiagramm. Bild 2.6 zeigt nochmals diese Palette, die bereits in Kapitel 1 erwähnt wurde.



Bild 2.6 Werkzeugpalette

Die Werkzeugpalette enthält 11 Felder. Das oberste dient zur Aktivierung/Deaktivierung der automatischen Werkzeugwahl mit der Anzeige Grün/Schwarz.

Bei automatischer Wahl versucht das System, ein geeignetes Werkzeug zu finden, sobald man mit dem Mauszeiger über die Objekte im Panel oder Diagramm fährt. Es schaltet z.B. auf die Drahtrolle um (zweite Zeile unter dem Automatiksymbol, links), wenn man sich Terminals oder Funktionen nähert, die man verbinden könnte. Doch ist mancher Programmierer mit der speziellen Arbeitsweise der Automatik nicht einverstanden und verzichtet deshalb auf die Bequemlichkeit der automatischen Werkzeugwahl.

Will man ohne Automatik arbeiten, muss man die Schaltflächen darunter mit der linken Maustaste anklicken. Im Einzelnen haben sie folgende Bedeutung:

- Erste Zeile links (Hand mit gestrecktem Zeigefinger, Bezeichnung: 'Wert einstellen'): Dateneingabe, etwa durch Anklicken der Pfeile an einem Eingabesymbol oder durch Anklicken des Datenfeldes und Eingeben auf der Tastatur.
- Erste Zeile Mitte (Pfeil nach links oben, Bezeichnung: 'Position/Größe/Auswahl'): Auswählen von Objekten zur Positionierung oder Vergrößerung, auch zum anschließenden Löschen mit <Entf>. Beim Positionieren muss man in die Mitte des Objektes zeigen, beim Verändern der Größe auf irgendeine Ecke, dann linke Maustaste drücken und ziehen. In Bild 2.6 ist dieses Werkzeug gerade angewählt und deshalb grau gefärbt.
- Erste Zeile rechts (Buchstabe A): Texteingabe. Anklicken einer beliebigen Stelle im Panel oder Diagramm erzeugt ein rechteckiges Feld, in das man Text eingeben kann, z.B. als Kommentar. Damit lassen sich aber auch Namen und Wert eines Bedien- oder Anzeigeelements ändern.
- Zweite Zeile links (Drahtrolle): Verbinden von Objekten im Diagramm mit Datenpfaden.
- Zweite Zeile Mitte (Pfeil mit Schubkasten): Ermöglicht Öffnen des Kontext-Menüs des angeklickten Objekts mit der **linken** statt mit der rechten Maustaste.
- Zweite Zeile rechts (Hand mit ausgestreckten 5 Fingern): Bewegen aller Objekte in einem Fenster.
- Dritte Zeile links (rote Stopptaste): Setzen/Löschen von Haltepunkten ('Breakpoints') bei der Fehlersuche, dem so genannten Debugging.

- Dritte Zeile Mitte (gelber Kreis mit P): Anzeigen von Probedaten ('Probes') an Drähten zum Debuggen. Bewirkt das Erscheinen eines 'Sondenüberwachungsfensters' mit Werten der übertragenen Daten. Wirkt auch im Normalbetrieb, nicht nur im Debug-Modus.
- Dritte Zeile rechts (Pipette) und ganze vierte Zeile: Kolorieren, d.h. Farbe setzen und Farbe übernehmen. Damit kann man Panel und Diagramm nach eigenem Wunsch umfärben oder auch einzelne Elemente darauf wie LEDs, Tankfüllung usw.

Benutzt man die Automatik der Werkzeugpalette **nicht** (Anzeige oben schwarz), gelten folgende nützliche Hinweise:

- Leertaste (Space Bar) bewirkt **Werkzeugwechsel** zwischen 'Position/Größe/Auswahl' und 'Wert einstellen' (Hand mit gestrecktem Zeigefinger).
- <Tab> wechselt zwischen 'Wert einstellen', 'Position/Größe/Auswahl', 'Text bearbeiten' und 'Anschlüsse verbinden' (bzw. 'Set Color', falls sich noch keine Controls auf dem Frontpanel des VIs befunden haben)
- <Shift>+<Tab> stellt um auf Automatik.

Weitere wichtige Hinweise dienen dem **Positionieren** und **Kopieren** von Objekten:

- <Strg> + Bewegung mit links gedrückter Maustaste schiebt alle umliegenden Objekte auseinander, je nach Bewegungsorientierung in horizontaler oder in vertikaler Richtung oder auch gleichzeitig in beiden Richtungen.
- Cursortaste bewegt markiertes Objekt (gekennzeichnet durch gestrichelten Rand) um jeweils einen Pixel.
- <Shift> + Cursortaste bewegt markiertes Objekt um einen Gitterabstand.
- Markieren und Bewegen des Objektes mit der Maus bei gedrückter <Strg>-Taste kopiert das Objekt.
- <Shift> + Maus bewegt ein markiertes Element (oder mehrere markierte Elemente) nur senkrecht oder nur waagerecht, wobei die Richtung durch die zuerst registrierte Mausbewegung festgelegt wird.

2.2.2 Eingabe-/Ausgabe-Elemente

Wie bereits in Abschnitt 1.4 erwähnt, ist beim Aufruf von LabVIEW 2014 die Palette für die Bedienelemente voreingestellt. Wir können das aber entsprechend Bild 2.7 ändern, indem wir auf 'Anpassen' – 'Sichtbare Paletten ändern...' klicken und diejenige Palette **als einzige** markieren, die wir auf dem Frontpanel sehen wollen.

Wir sehen hier eine Fülle von kleinen Symbolen, die man für die Ein- und Ausgabe auf das Panel ziehen kann. Dazu muss man aber zunächst eines der 12 Symbole anklicken, um das zugeordnete Untermenü bzw. Unter-Untermenü zu öffnen.

Rezept: Bei angepinnter Palette und **n i c h t** gedrückter Maustaste so lange durch die sich öffnenden Menüs gehen, bis das gewünschte Element gefunden ist. Dann linke Maustaste drücken. Nun erscheint eine Hand, die das Element hält. Dieses an die vorgesehene Stelle im Panel ziehen und dort loslassen.

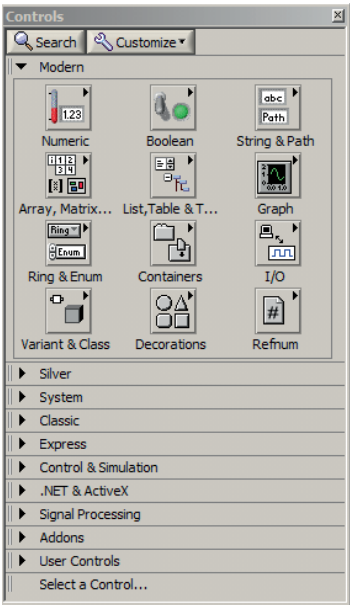


Bild 2.7 Palette 'Elemente'. Diese Elemente verwendet der Programmierer zur Gestaltung des Frontpanels. Hier wurden alle verfügbaren Palettenkategorien ausgewählt

Es ist nicht nötig, jedes Mal den ganzen Weg von der obersten Palette durch alle Unterpaletten zu verfolgen, bis man das gewünschte Element gefunden hat. Will man mehrere Elemente aus einer Unterpalette ins Panel ziehen, fixiert man diese mit dem kleinen Reißnagel links oben. Holt man die Palette nicht mit der rechten Maustaste, sondern über 'Ansicht' – 'Elementpalette', ist die Fixierung bereits automatisch erfolgt.

Bild 2.8 zeigt die Unterpalette, die zum oben links stehenden Symbol 'Numerisch' gehört. Sie wird angezeigt, wenn man unter 'Werkzeuge' – 'Optionen' – 'Elemente- und Funktionenpalette' – 'Formatierung' als Palette 'Kategorie (Symbol und Text)' eingetragen hat.

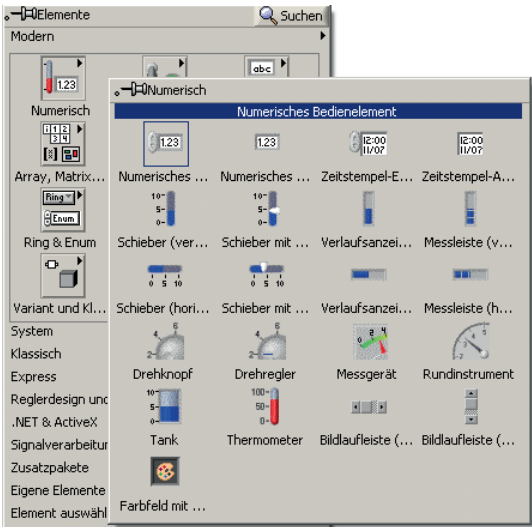


Bild 2.8 Unterpalette 'Numerisch'

Man sieht dann in der Unterpalette Elemente für die Ein- und Ausgabe von Zahlen (oben links, erstes und zweites Element), daneben entsprechend für Zeiteingaben und -ausgaben. Die zweite Zeile enthält verschiedene Formen vertikaler Schieberegler, die dritte horizontale Schieberegler. In der vierten Zeile findet man manometerartige runde Anzeigeelemente, die aber auch zur Eingabe verwendet werden können. In der fünften Zeile sind ein Tank (links), ein Thermometer und Bildlaufleisten angebracht, in der sechsten Zeile ein Farbfeld.

In ähnlicher Weise findet man auch unter den anderen Symbolen der Palette in Bild 2.7 Eingabe- und Ausgabeelemente. Ausprobieren! Zusammenfassend kann man die Bedeutung der Symbole in 'Elemente' – 'Modern' bei einer Anordnung 4 x 3 wie folgt beschreiben:



'Numerisch', **erste** Zeile links: Elemente zur Ein- und Ausgabe numerischer Zahlentypen.



'Boolesch', **erste** Zeile Mitte: Elemente, die nur zwei Zustände annehmen können, wie Schalter oder LEDs, OK-Schaltfläche oder der Stopp-Knopf.



'String & Pfad', **erste** Zeile rechts: Ein- und Ausgabe von Text, wobei die einzelnen Zeichen im ASCII-Code verschlüsselt werden. LabVIEW unterscheidet die Datentypen 'String' und 'Pfad'.



'Array, Matrix & Cluster', **zweite** Zeile links: Ein- und Ausgabe von Vektoren, Matrizen oder höherdimensionalen Feldern. Die einzelnen Elemente müssen vom gleichen Datentyp sein. Ferner Cluster, entsprechend den Strukturen in C. Hier dürfen die einzelnen Elemente auch von unterschiedlichem Datentyp sein.



'Liste & Tabelle & Baumstruktur', **zweite** Zeile Mitte.



'Graph', **zweite** Zeile rechts: grafische Darstellung von Funktionen und Relationen.



'Ring & Enum', **dritte** Zeile links: zyklische Bedienelemente zur Steuerung von Fallunterscheidungen.



'Container', **dritte** Zeile Mitte: zum Beispiel Registerkarte oder Trennbalken Frontpanel.



'I/O', **dritte** Zeile rechts: Eingabe/Ausgabe (Input/Output) von Signalen.

Index

A

Ability-VI 554

- AdaptToInputs.vi 563
- Bounds.vi 564
- GenerateCode.vi 566
- GenerateCode.vi mit SubVI 568
- GetTerms3.vi 565
- Image.vi 564
- Initialize.vi 568
- state.ctl 560, 562
- wie finden 561

Ablauf

- ablaufinvariant (reentrant) 96, **104**
- parallel oder sequentiell 260

abschließende Nullen ausblenden 50

absolutes Zeitformat 64, 137

Abstraktion, technologische 190

Abtastfrequenz 143

Abtastinformation **143**, 227

Access, Microsoft 408, 432

ActiveX 44

- ActiveX-Container 411
- ActiveX-Elemente (controls) 411
- Beispiel einer Homepage 413
- Datenaustausch mit Excel 414
- LabVIEW als Server oder Client 411

Addition

- Beispiel 25
- von Matrizen 76
- von numerischen Strings 106

Adresse, MAC- und IP-Adresse 408, 437

Aktualisierungsmodus 128

aktuelle Werte als Standard **52**, 72

aktuelles Datum 68

Aliasing 222, **231**

Alternative 48, **53**

Alternative als Case-Struktur 53

Amplitudengang 224

Amplitudenspektrum 222

Analogausgabe 330

Analogrechnerprinzip 245

Analogwert

- Ausgabe über USB 296
- erfassen mit NI USB-6251 324
- lesen 323

Animationen, Programmierung von 46

Anschluss 31

- Aufteilung Anschlussfeld 99
- erforderlich, empfohlen, optional 101
- LabVIEW-SubVI 96
- polymorphes VI 106
- trennen 101
- von Eingängen und Ausgängen 101
- welches Feld gehört zu welchem

Panelement 101

Anwendungssteuerung, Unterpalette 46

Anzeigeelement 27

- ausblenden 52
- erstellen 50

Anzeigeformat 64

Anzeigeformat von Datentypen 63

Application Builder 192

Approximation durch Fourierreihe 233

arithmetisches Mittel als Tiefpass 235

Array 71

- Array von Clustern 140
- Array-Konstante, Beispiel 367
- Aufziehen eines Arrays 72
- Beispiele 2-dimensionalen Arrays 74
- Bildung, Definition, Dimension 71
- Dimension **73**, 78
- einzelne Elemente 79
- Funktion zum Initialisieren 73
- Funktionenunterpalette 45

- höherdimensionales 73
- Index 71
- Indexfenster bei Arrays aufziehen 73
- initialisieren 71
- kein Array von Arrays 88
- komplexer Zahlen, Darstellung 228
- lesen/schreiben auf Festplatte 172
- programmgesteuerte Initialis. 73
- rechnen mit Arrays 75
- Typ boolesch, numerisch, String 72
- Verknüpfung (Verkettung) 75
- von Clustern 88
- Zweistufiger Prozess zur Array-
Erzeugung 71
- Array & Cluster, Unterpalette 43
- ASCII-Code 291
- Asynchron 291
- atomar 264
- Audio-Ausgabe 183
- Aufbau von LabVIEW 185
- Aufruf
 - erster Aufruf einer Funktion 248
 - rekursiver SubVIs 96, **104**
 - von LabVIEW 22
- Aufschlüsseln
 - einer Waveform 141
 - eines Clusters 85
- Aufwärtskompatibilität der LabVIEW-
Versionen 21
- Aufzählung *Siehe* Enum
- Ausführung
 - ablaufinvariant (reentrant) VIs 103
 - mit Synchronisierung von Threads 271
 - quasi-parallele von VIs 261
- Ausgabeelement *Siehe* Anzeigeelement
- Ausgang aus Schleifen 61
- Ausrichtung
 - Gitter 39
 - von Symbolen 39
- ausschneiden Objekt 34
- Automat und Konfigurationsdatei 182
- automatische
 - Erzeugung von Eingängen und
Ausgängen für Funktionen 79
 - Namensvergabe beim Kopieren 76
 - Verdrahtung 38, 39
 - Werkzeugwahl 40

B

- Balkendiagramm 130
- Bandpass, Bandstopp 234
- Bargraph *Siehe* Balkendiagramm
- Basis 2, 8 oder 16 zur Zahlendarstellung
64
- Basisklasse, Eigenschaften 154
- Baud 292
- Bedienelement **23**, 25
 - Beispiel Case-Struktur 54
 - eigenes Bedienelement 202
- Bedienungsanleitung zu LabVIEW 201
- Benutzerdefinierte Bedien-/
Anzeigeelemente 115
- Benutzerschnittstelle 189
- Bessel-Filter 240
- Betriebsmittel *Siehe* Ressource
- Betriebssystem, Zuweisung von
Prozessorzeit 262
- Bewegen aller Objekte 40
- Bibliothek
 - eigene LLB anlegen 202
 - Funktionsbibliotheken 188
 - konvertieren in, von LLB 202
 - LLB in LabVIEW 200
 - XML-Bibliothek 556
- Bild-Ring, Erzeugung 89
- Binärzahl 64
- Bindung
 - dynamische 108, 109
 - statische 108
- Bitmitte 291
- Bit-serielle Ein-/Ausgabe 290
- Blank_XNode.xnode 557
- Blockdiagramm **22**, 36
 - gekoppeltes Feder-Masse-System
254
 - im MATLAB®-Stil 255
 - mit XNodes 555
- Bode-Diagramm 235
- Bookmark 121
- Boolesch 53
 - Bedien- und Anzeigeelemente 66
 - Datentyp, Funktionen 66
 - Konstanten TRUE, FALSE 53, 67
 - Unterpalette **43**, 46
- Breakpoint 40

Bündeln

- einer Waveform 141
- zur Clustererzeugung, 83

Butterworth-Filter 238

C

Call by value 150

CAN-Bus

- als Feld-Bus 299
- CANopen-Protokoll 302
- Identifier 300
- Interface USB-to-CAN 301
- Normung 299
- Prioritätssteuerung 300
- Protokoll 299

CANopen 302

Case-Struktur 45, **53**

- Ausgabetunnel 57
- boolesche Selektorsteuerung 55
- Enum-Selektorsteuerung 362
- Groß-/Kleinschreibung ignorieren 361
- Integer-Selektorsteuerung 363
- String-Selektorsteuerung 360
- Variable herausführen 56
- Voreinstellung 56

C-Code einbinden 204

Chart 123

- Historie 127
- Legende 130
- löschen 128
- mehrere Kurven in einem Chart 129
- Signalverlaufs-Diagramm (Chart) 127
- Skalierung der Ordinate 131
- sweep, scroll, scope 128

CLF-Knoten

- konfigurieren 209
- Parameter 211

Client 437

Cluster 43, **81**

- Änderung einzelner Elemente 83
- aufschlüsseln **45, 85**
- automatisch Rahmen skalieren 82
- bündeln beim Chart 129
- bündeln, umwandeln 45
- Clusterkonstante, Beispiel 365
- Elemente umordnen 87
- Erzeugung durch 'Bündeln' 83

- Erzeugung in zwei Stufen 82
- Funktionenunterpalette 45
- nach Namen aufschlüsseln 85
- nach Namen bündeln 84
- Programmgesteuerte Erzeugung 86
- Reihenfolge der Elemente 82
- Steuerung durch oberen Anschluss 83
- XY-Graph als Cluster-Typ 139
- zur Fehlerbehandlung 81

CMD/CSA-Prinzip 299

Code, wieder verwendbarer 123

COM-Schnittstelle

- COM1, COM2 292
- virtuelle 292

Conrad Elektronik, DLL K8055D.dll 295

CPU-Auslastung 262

cRIO

- Anwendung läuft autonom 497
- Aufbau des cRIO-9014 465
- Aufruf Xilinx-Compiler 475
- Beispiel 1 autonom nur mit FPGA 497
- Beispiel 2 autonom aber mit FPGA und cRIO-RT-Computer 499
- Beispiel Digitalausgabe 472
- Beispiel Frequenzgang 479
- Beispiel Zähler 477
- Definition 464
- DMA, FIFO 484
- eingeschränkte Funktionspalette 474
- Einstellung Sinusgenerator 484
- erhöhte Taktfrequenz 477
- Finden logischer Fehler im VI 489
- 'FPGA Target'-Modus 489
- installieren Hardware/Software 466
- IP-Adresse 467
- Modul NI 9263 für Analogausgabe 464
- Modul NI cRIO 9215 für .Analogeingabe 464
- Modul NI cRIO 9401 für Digital-Ein-/Ausgabe 464
- nur ein 'Haupt-VI' 477
- Programmtypen 471
- rtexe-Datei 465
- Scan Interface oder LabVIEW FPGA Interface 473
- Schreiben in FIFO 486

- Simulationsmodus 489
- Sinusfunktion 484
- Software-Installation 470
- System arbeitet ohne Host 472, **497**
- TCP/IP-Verbindung zum PC 465, 468, **470**
- Träger cRIO-9103 464
- Umgebungsvariable 490
- Umstellung von einem Standalone-Projekt zum nächsten 502
- vermeiden Xilinx-Übersetzung durch Scan-Interface 473
- zeitgesteuerte Schleife mit nur 1 Taktzyklus 486
- Ctl 202
- Klassenbildung 384
- Cwdss.exe, automatischer Start 442

D

- DAQ-Assistent 323
 - NI-DAQmx-Code 332
 - Programmierung mit 331
- DAQmx-Funktionen 326
- Darstellung 50
 - Genauigkeit 63
 - im Bode-Diagramm 236
 - logarithmische Darstellung 236
 - numerischer Datentypen 63
 - Parameterdarstellung eines Kreises 138
 - Speicher sparend bei Gleitkommazahlen 63
- Data Acquisition Board *Siehe* Datenerfassungskarte
- DataSocket
 - Konzept und cwdss.exe 442
 - Lesefunktion 443
 - Programmieren mit 441
 - Schreibfunktion 443
 - Verfügbarkeit 441
 - Vorbereitung der Verbindungen 442
- Datei
 - aktueller Pfad zum VI und zur EXE-Datei 175
 - CSV-Datei 408
 - Dateioperationen 164
 - Dateiposition festlegen 167
 - Erweiterung 'txt' 165

- EXE-Datei 162
- fortgeschrittene Funktionen 165
- 'High Level'-Funktionen 162
- Konfiguration eines Automaten 182
- Konfigurationsdateien 181
- Lesezeiger, Schreibzeiger 166
- 'Low Level'-Funktionen 162
- LVM-Datei 176
- Messwerte in Datei schreiben 176
- öffnen, erstellen, ersetzen 165
- öffnen, schließen 165
- Protokolldatei 167
- Schlüssel einer Konfigurationsdatei 182
- TDM- und TDMS-Dateien 176
- überschreiben ohne Warnung 167
- verketteten Schreib-/Lesefunktionen 172
- Zeichen in Datei lesen, schreiben 165
- Datei-I/O, Unterpalette 46
- Daten
 - Austausch zwischen Prozessen 260
 - Datenabhängigkeit und Sequenz 187
 - Datenabhängigkeit zweier While-Schleifen 260
 - Datenaustausch mit globalen Variablen 249
 - Datenpfad komplett sichtbar 356
 - Typ DBL, SGL 172
 - Verbindung mit Datenpfad 40
- Datenabhängigkeit erzwingt Sequenz 80
- Datenbank 432
 - Abfrage 436
 - Datenquelle 433
 - DSN (Data Source Name) 433
 - Inhalt ändern 436
 - Sprache SQL 433
 - Verbindung mit Datenquelle 433
 - Verbindung mit LabVIEW 432
 - VI-Bibliothek 436
- Datenbits bei RS-232 291
- Dateneingabe 40
- Datenerfassung
 - Aufbau des VI 325
 - simulierte **326**, 328
- Datenerfassungskarte
 - andere Anbieter 317
 - NI USB-6251 318
 - NIDAQ PCI-MIO-16E-4 317

Datenfluss

- als LabVIEW-Prinzip 22
- LabVIEW als datenflussorientierte Programmiersprache 187

Datenoperation 'Element löschen' 78

Datenpaket bei serieller Übertragung 291

Datentyp

- bei Schieberegister 59
- boolesch 63, **66**
- 'Enum' 56
- 'Menüring' 56
- numerisch 63
- Pfad 63
- String 63, **68**
- 'Textring' 56
- Variant 92, 155

Datenvergleich 46

Datum

- aktuelles 68
- Datum/Zeit in Sekunden ermitteln 69
- Datum/Zeit-String lesen, formatieren 69
- Format, Betriebssystemeinstellung 69

dB, Definition 236

DBL (Datentyp) 26

Deaktivierung

- bedingte, unbedingte 197
- von Programmteilen 197
- Vorteile 199

Debuggen

- Deaktivierungsstrukturen 197
- Einstellen Deaktivierung im Projekt 199
- Fehlersuche 40
- großer Programme 199

Defaultwert *Siehe* Standard, *Siehe* Standard

Dezimalpunkt, lokaler 36

DFT (Diskrete Fouriertransformation) 229

Diadem 179

Diagramm

- anzeigen, Panel 33
- gestalten 94

Dialogfunktionen 44

Differenzialgleichung 243

- Anwendung auf Umweltproblem **258**
- Beispiel eines Systems 257
- Beispiel Tourismus – Umweltqualität 259
- Lösen mit Analogrechnerprinzip 245

- Lösen mit ODE-Funktionen 243, 254
- MATLAB®-Blockdiagramm 245
- Schrittweite 250
- System in MATLAB-Simulink® 257
- Systeme von Dgl. **253**
- Umwandlung in System von Dgl. 243

Digital-I/O 331

Dijkstra 263

Dimension

- eines Arrays 78
- eines Vektors 78

Direct Memory Access *Siehe* cRIO DMA

DLL

- einbinden mit CLF-Knoten 204
- K8055D.dll (Conrad Elektr.) 295

doppelte Genauigkeit 63

Drähte, Verbindungslinien **31, 44**Drahtrolle als Werkzeug **40, 99**

Dreiecksberechnung 97

Dreiecksberechnung im Web 457

dynamische Bindung von SubVIs 97

dynamischer Dispatcheingang bei OOP 386

E

Echtzeit, Beschriftung der X-Achse 145

Editieren 39

Editor, grafischer 186

Effizienz

- eines VIs 376
- Werkzeuge zur Bestimmung 378

eigene Elemente 362

Eigenschaft

- allgemein nach spezifisch 154
- Basisklasse 154
- Vererbung von Eigenschaften 152

Eigenschaftsknoten 123

- Aktualisierungsmodus 128
- Beispiel 123
- Darstellung im Frequenzbereich 229
- finden 123
- implizite, explizite Darstellung 156
- Kontextmenü 124
- Reihenfolge der Verarbeitung 126
- 'sichtbar' 124
- StringsUndWerte[] 367
- Sweep-, Scope- und Scroll-Modus 128

- 'TastenFokus' 125
- von Element entfernen 156
- eindimensionales Array oder Vektor 71
- Einfügen Objekte 34
- Eingabe abschließen, quittieren 29, **126**
- Eingabeelemente 41
- Einheitsmatrix erzeugen und initialisieren 79
- Einstellungen
 - Benutzerdefiniert 102
 - 'Datei' – 'VI-Einstellungen' 35
 - Fenstererscheinungsbild 102
 - von VIs 102
 - 'Werkzeuge' – 'Optionen' 35
- Element **23**
 - Beispiel 54
 - erstellen 25
 - gleich Anzeigeelement 27
 - gleich Bedienelement 25
 - löschen 27
- Endlosschleife 67
- Entfernen gestrichelte Linien 34
- Entwicklungsumgebung 185
- Enum 43, **54**, 88
- EOF-Zeichen 162
- EOL-Zeichen 163
- Ereignis
 - Datenknoten **280**
 - durch Wertänderung 284
 - Filtern von Ereignissen 285
 - Quelle ist Applikation 281
 - Quelle ist Frontpanel-Element 279
 - Quelle ist Timeout 280, 284
 - Quelle ist Wertänderung 284
 - Stopp 283
 - Wertänderung generieren 284
- Ereignis oder Event **279**
- Ereignisstruktur im Zustandsautomaten 372
- Eselsohr, Gestaltungselement der For-Schleife 58
- Ethernet
 - CSMA/CD-Protokoll 437
 - Datenübertragungsrate 437
 - Prinzip 437
- Event *Siehe* Ereignis
- Excel 46
 - als ActiveX Automation Server oder Client 411
 - Datenaustausch mit LabVIEW 408, 414
 - Datenerfassung, Schreiben parallel 431
 - Grafik erstellen 421
 - Grafik unter LabVIEW erstellen 428
 - höhere Geschwindigkeit beim Datenverkehr Excel – LabVIEW 420
 - Macro für höhere Geschwindigkeit 425
 - Macros erstellen durch Aufzeichnen 421
 - Makros in Visual Basic aufrufen 425
 - Mappe (Workbook) anzeigen 416
 - NI-Beispiel zum Daten eintragen 419
 - Nutzung Makros in 'Projekt.xls' 425
 - öffnen, schließen unter LabVIEW 415
 - schreiben mehrerer Dateien mit modifizierten Namen 428
 - 'Set Cell Value.vi' 420
 - Visual Basic-Programm 424
- EXE-Datei 192
 - aktueller Pfad zu einer 173
 - Erstellung 173, **192**
 - ohne Diagramm 194
 - Runtime-Bibliothek 192
- Experimentierplatine Velleman 295
- Exponentialdarstellung von Gleitkommazahlen 64
- Express-VI **146**, 176, 215
- F**
- FALSE 55
- Farbe
 - einer Kurve 123
 - Farbpalette 130
 - setzen 41
 - übernehmen 41
- Farbgebung 54
- Feder-Masse-System
 - Differenzialgleichung 243
 - gekoppeltes System 253
- Feedbackknoten *Siehe* Rückkopplungsknoten
- Fehler
 - 'Hornissen gestreifte' Fehlerleitung 158
 - Bedien- und Anzeigeelemente 157
 - Dreiecksbeispiel 98
 - einfacher Fehlerbehandler 82, 157

- Erzeugung der Standardanzeige 159
- Erzeugung Fehler-Cluster 82
- Fehler-Cluster 81, 244, 247
- Funktionen ohne oder mit vereinfachter Fehlerleitung 158
- Meldung mit und ohne Dialog 156
- suchen (debuggen) 29, 40
- Fehlerliste öffnen 34
- Feinheit des Gitters 39
- Feld, Array 43, **71**
- Feld-Bus 299
- Fenster
 - aktives 103
 - Fenstererscheinungsbild anpassen 103
 - Größe ändern, minimieren 103
 - schließen 103
 - Typ (frei verschiebbar, Standard, modal) 103
- Fernüberwachung 20
- Festplatte, lesen/schreiben 162
- FFT (Fast (schnelle) Fouriertransformation) 229
- FGV 277
 - Grundstruktur 277
 - mit zusätzlicher Inkrementierung 278
- Field Programmable Gate Array *Siehe* FPGA
- FIFO als Prinzip bei Queues 265
- FIFO beim cRIO 484
- Filter
 - analoge Filter **235**
 - arithmetisches Mittel als Tiefpass 235
 - Beispiele realer Filter 239
 - Bessel-Filter 240
 - Butterworth-Filter 238
 - digitale Filter **235**
 - Filter vom Typ FIR, IIR 238
 - filtern im Frequenzbereich 240
 - filtern im Zeitbereich 235
 - filtern von Ereignissen 285
 - Frequenzgang, Grenzfrequenz 234, **239**
 - idealer Bandpass, Bandstopp 235
 - idealer Filter 234
 - idealer Hochpass, Tiefpass, 235
 - LabVIEW-Filterfunktionen 234, **238**
 - Ordnung von Filtern 238
 - Tschebyscheff-Filter 238
 - Überschwingen beim Tschebyscheff-Filter 240
- Filtertyp 239
- Finite State Machine *Siehe* Zustandsautomat
- FIR-Filter 238
- Fixieren einer Palette 42
- Folge-Struktur *Siehe* Sequenz
- Formelknoten 190
- For-Schleife 48, 51, **57**
 - als Kartenstapel 58, 59
 - 'Eselsohr' 58
 - Indizierung Datenpfad 78
 - i-Terminal 57
 - N-Eingang 58
 - Realisierung eines Zählers 57
 - Steuerung durch Arrays 77
 - Zahl der Durchläufe 61, 77, **79**
- Fourier
 - DFT (Diskrete Transformation) 229
 - Eindeutigkeit 224
 - einseitige, zweiseitige Transformation 222, **225**
 - endliche Fourierreihe 224
 - FFT (Fast (schnelle) Fourier Transformation) 229
 - inverse Fouriertransformation 240
 - Koeffizienten in LabVIEW 228
 - komplexe Fouriertransformation 225
 - komplexer Ausgangsvektor in LabVIEW 228
 - Nyquisttheorem 230
 - reelle Fouriertransformation 223
 - Reihe als optimale Näherung 224
 - Satz von Fourier 223
 - Sinus-/Kosinus-Reihe 224
 - Transformation in LabVIEW 226
 - unendliche Fourierreihe 223
- FPGA
 - Chip von Xilinx® 465
 - Definition und Bedeutung 464
 - Finden logischer Fehler im VI 489
 - im cRIO-9014 465
 - VHDL-Compiler 465
- Frequenz
 - Balkendiagramm Frequenzbereich 225
 - bei serieller Übertragung 291

- Frequenzauflösung 222, **232**
- Frequenzbereich 222
- Frontpanel **22**, 36
 - beim Aufruf anhalten 109
 - beim Aufruf anzeigen 108, 462
 - beim Laden öffnen 108
 - Erscheinungsbild 102
 - Events oder Ereignisse 279
 - globale Einstellungen 102
 - lokale Einstellungen 108
- FSM *Siehe* Zustandsautomat
- Funktion
 - Bibliothek 22, 188
 - boolesche 67
 - eigene Funktionen für user.lib 201
 - erster Aufruf einer Funktion 248
 - Funktionen zur Prozesskommunikation 265
 - grafische Darstellung 43
 - im Gegensatz zur Relation 138
 - logische Funktionen 67
 - periodische 222
 - zur Bildung des Matrizenprodukts 77

funktionale globale Variable *Siehe* FGV

G

- G *Siehe* Grafische Programmiersprache
- Gate Array, Erklärung 464
- Gaußsche Zahlenebene 225
- Genauigkeit
 - der Zahlendarstellung 64
 - doppelte 50
 - numerischer Verfahren 243, **250**
 - Vergleich bei verschiedenen Schrittweiten 250
 - Vergleich in grafischer Darstellung 251
 - Vergleich Runge-Kutta und Euler 250
 - VI zur Ermittlung der Genauigkeit 252
- Geschwindigkeit VI und EXE 192
- Gestaltungselemente 54
- Glättung beim Filtern mit Tiefpass 236
- Gleitkommazahlen 63
 - Darstellung 'automatisch' 64
 - Darstellung 'Fließkomma' 64
 - SI-Darstellung 64
 - wissenschaftlich 64

- globale Variable 45
 - funktionale globale Variable 277
 - Laden wie ein Unterprogramm 275
 - mehrere globale Variablen 275
 - Rezept zur Nutzung 274
- GND, 'Ground' oder 'Erde' 290
- GObjekt beim VI-Server 152
- GPB-Bus 315
- Grafik & Klänge, Unterpalette 46
- Grafik, Beispiel 30
- Grafik, unter LabVIEW erstellt in Excel 428
- Grafik-Ring 88
- grafische Ausgabe 123, **127**
- grafische Programmiersprache 185
- grafische Programmierung 22
- Graph 43, 133, 43
 - Abszisse mit relativer Zeit 137
 - Array von Datenpunkten 133
 - Bild löschen 134
 - einer Sinuskurve 133
 - Farbbänderung 133
 - Legende 129, 133
 - mehrere Kurven in einem Graph 134
 - Signalverlaufsgraph (Graph) 127
 - Skalieren der Abszisse 136
 - Unterpalette 43
 - Zeit statt Index anzeigen 136
- Grenzfrequenz
 - Definition beim Filter 239
 - obere, untere Grenzfrequenz 239
- Größenänderung von Objekten 40
- Gruppierung von Objekten 94

H

- Hardwareunterstützung 188
- Hauptdiagonale Matrix 80
- Hexadezimalzahl 64
- Hex-Anzeige 'String' 71
- Hierarchie bei Vererbung 152
- Hilfe
 - Beispiele suchen 214
 - Developer-Zone 213
 - Funktionen-/Anwendungshilfe 214
 - Kontexthilfe, LabVIEW-Hilfe 214
- Hintergrundfarbe 54

Hinweisstreifen und Beschriftung 38
Historie-Daten, Historie-Länge 128
Hochpass, idealer Hochpass 234, 235
HTTP 447

I

I/O, Unterpalette 43
I32, Integer mit 32 Bit 63
Icon *Siehe* Symbol
Indizierung Datenpfad aktivieren 78
Initialisieren
– Einheitsmatrix 79
– programmgesteuert initialisieren 72
– Schieberegister 58
Installation von EXE-Programmen 195
Installer **195**
Instanz
– einer Klasse 381
– einer Zahl 381
Instrumenten-I/O 290
Integration
– Eulerverfahren 247
– Laplace-Symbolik mit 's' 249
– mit einem einzelnen SubVI auch bei mehrfacher Integration 247
– verschiedene SubVIs bei mehrfacher Integration 247
Intelligente Sensoren 345
Internet 437
– Beispiele/Lösungen zu LabVIEW 21
– Fernüberwachung 20, 437
– Hilfe suchen im Internet 31
Interpolation 130
interpretieren oder kompilieren? 186
Inverse einer Matrix 79
Inverse Fouriertransformation 240
Invertierungsoperator 'NICHT' 61
IP gleich 'Internet Protocol', Aufgabe und Bedeutung 439
IP-Adresse 437
– Aufbau 438
– finden 438
– Localhost 438
– partielle Festlegung durch Systemadministrator 438
– Vorgaben NIC und DE-NIC 438

Isolierte Punkte 130
I-Typ, Integer mit Vorzeichen 63
IVI
– Beispiel-VI 351
– Definition 349
– Geräteklassen 350
– Installation NI USB-5133 350
– NI-Scope Express.vi 353
IXXAT, CAN-Produkte 301

J

JSON 452

K

Kalibrierung 183
Kanal
– Abtastfrequenz 325
– Einfachmessung 325
– kontinuierliche Messung 325
– Mehrfachmessung **325**, 327
– physikalischer Kanal 324
– virtueller Kanal 325
Kapseln 123
Klang *Siehe* Grafik & Klänge
Klasse 123
– allgemein nach spezifisch 155
– Bildung in LabVIEW 383
– GObject 152
– Hierarchie anzeigen 189, 405
– kapseln von Funktionen und Variablen 123, **381**
– Klasse 'Bedienelement' 154
– Klasse 'Digital' 155
– Klassenbezeichner 152
– Klassenbezeichner-Konstante 152
– LabVIEW-Objekt 391
– Methode 385
– private Daten 382
Klassische Elemente, Unterpalette 44
Knoten **31**, 50
kolorieren 41
Kommastellen 54
Kommentare 40
Kommentare zu
– Elementen 120
– Funktionen 121
– Leitungen 120

Kommunikation, Unterpalette

'Datenkommunikation' 46

komplexe Zahlen **63**, 66

Konfiguration

– CLF-Knoten 211

– CLF-Knoten für eine
Experimentierplatine 296

– Express-VIs 123

– Kalibrierungs-VIs 183

– Konfigurationsdateien 181

– konfigurieren statt programmieren 146

– NI USB-6251 320

Konstante

– numerische Konstante 63

– Pfadkonstante 169

– String-Konstante 69

Konstruktorknoten 207

Kontexthilfe ein-/ausschalten 34

Kontextmenü 37

– Anschluss eines VI 99

– für Datentyp 63

– öffnen mit linker Maustaste 40

Kontrast zum Hintergrund 39

Kontrollelement *Siehe* Ctl

kopieren Objekte **34**, 76

Kosinusfunktion 129

Kuh als XNode 561

Kurve

– Farbe, Linienstil 130

– Strichstärke 123

– wandernde 256

L

LabVIEW

– ActiveX-Container 411

– als ActiveX Automation Server und
Client 411

– als Entwicklungsumgebung 185

– Anbindung an C, C++, C# 20

– Aufbau von 185

– Aufruf von LabVIEW 22

– Aufwärts-/Abwärtskompatibilität 21

– Bedeutung 19

– Bedienungsanleitung 201

– bedingte Deaktivierungsstruktur 191,
197

– Beispiel zweistufige Vererbung 403

– Berichterzeugung 46, **147**

– Bibliotheken 200

– Dateioperationen 179

– Diagramm-Deaktivierungsstruktur
197

– Editor 185

– Erfinder 19

– Filterfunktionen 234, **238**

– Grafik in Excel erstellen 428

– Hardwareunterstützung 188

– Hilfen 213

– Klasse 'LabVIEW-Objekt' 391

– Laufzeit VI und EXE-Datei 196

– Makros in Visual Basic aufrufen 425

– Microsoft Web Browser 412

– mit Optionen speichern 21

– OOP ab LabVIEW 8.2 382

– Projekt-Explorer 200

– Projektverwaltung 190

– Richtung Signalfuss im Diagramm 259

– Server-Client-Paar 441

– Shortcuts 33

– Symbolleiste als XControl 526

– Systemfunktionen in vi.lib 201

– Template 201

– Testen großer Programme 199

– Verbindung mit Datenbank Microsoft
Access 432

– verschiedene Rechenmodelle 190

– Version 2014 20

– VHDL-Compiler 465

– 'Wrapper'-VI 416

– XControl 505

– Zustandsdiagramm 359

LabVIEW.ini 556

LabVIEW-Webdienst-Anfrage 450

LAN 291

Ländereinstellungen 36

Laplace-Symbolik mit 's' 249

Latch (verriegelt) 67

Laufdiagramm (Sweep-Modus) 128

Laufzeit VI und EXE-Datei 196

Laufzeitfehler 272, **277**

Laufzeitprobleme **275**

– bei globalen Variablen 278

– bei lokalen Variablen 275

lava xnode manager 557

- LED, Anzeigeelement 66
- leerer Vektor 128
- Legende *Siehe* Chart oder Graph
- Leitung, gekreuzt, nicht gekreuzt (RS-232) 291
- letzten Schritt wiederholen, zurücknehmen 34
- Linienbreite 130
- Linux 20
- Liste & Tabelle, Unterpalette 43
- LLB
 - Nutzen 204
 - umwandeln 202
- Localhost und IP-Adresse 438
- Logische Funktionen, Operatoren 46, **67**
- Lokale Variable 45
- Lokaler Dezimalpunkt 36
- Löschen
 - Element 27
 - Verbindungsleitung 28
- LVM-Datei 176, 177
- M**
- MAC-Adresse finden mit 'ipconfig/all' im DOS-Fenster 438
- Makro für höhere Geschwindigkeit beim Datenaustausch 427
- Marken beim Semaphor 269
- Markierung von Objekten 94
- Maschinencode, -programm 186
- MathScript-Knoten 213
- MATLAB® 243
- Matrix
 - Elementweises Multiplizieren 77
 - Hauptdiagonale 80
 - Inversion 79
 - Matrix oder 2-dimensionales Array 71, 73
 - Multiplikation **76**
 - Spalten-, Zeilenindex 73
 - transponierte **135**
- Mauszeiger fünffingrige Hand 41
- MAX (Measurement & Automation Explorer) 317
 - Aufruf 319
 - physikalischer, virtueller Kanal 324
 - Sprache Deutsch 319
- Mehrfachalternative 48, **53**
- Mehrfacharithmetik 85
- Melder *auch* Notification
 - Beispiel Produzent – Konsument 272
 - Daten übertragen mit Melder 272
 - Wann soll man sie einsetzen? 273
- Menüring 56, **88**
- Messdatenerfassung Überblick 290
- Mess-I/O 290
- Messsignal, Störung, Filterung 222
- Messwerte in Datei schreiben 176
- Methode
 - einer Klasse 385
 - verschiedene Methoden in einem Anwendungsfall 400
 - zur Manipulation privater Daten 386
- Methode der kleinsten Quadrate 233
- Methodenknoten 123, **133**, 154
- Metronom 58
- Moderne Elemente, Unterpalette 37
- Modulaustausch bei OOP 397
- Modus 'Wiederholt ausführen' 76
- Multiplikation von Matrizen 76
- Multiprocessing, -tasking, -threading 262
- Münzautomat
 - Zustandsautomat mit Queues 372
 - Zustandsdiagramm 363
- N**
- Nachkommastelle 64
- Namen
 - automatische, manuelle Vergabe 76
 - Cluster nach Namen aufschlüsseln 85
 - Cluster nach Namen bündeln 84
 - symbolische im MAX 323
- National Instruments (NI) 19
 - BNC-2096 347
 - PCI-4474 347
- N-Eingang For-Schleife 77
- NET 44
- NI USB-6251
 - Beispiel mit zwei Spannungen 336
 - Eigenschaften 334
 - Elemente in der Queue 342
 - Parallelverarbeitung 340
 - Streaming 338
 - triggern 337

Normung RS-232 **291**

Notification *Siehe* Melder

Nullen nach dem Komma erhalten 63

Nullmodem 291

Numerisch

- Bedien- und Anzeigeelemente 66
- Datentyp 63
- Funktionen 66
- Konstante 63
- mit Clustern rechnen 85
- Unterpalette **43**, 46

Numerische Verfahren, Genauigkeit 243

Nyquisttheorem 230

O

Objekt, Objekte

- alle bewegen 40
- ausrichten, anordnen 94
- ausschneiden, kopieren, einfügen **34**, 55
- bearbeiten, Kontextmenü 57
- markieren 94
- mit Maus platzieren 41
- Objektgröße verändern 94
- suchen nächstes Objekt 33, 34

Objektorientierte Programmierung 123,
380

Occurrence **265**, 268

ODE-Verfahren 243

Offset einer Sinuskurve 131

Oktalzahl 64

OOP 123, 152, **380**

- abgeleitete Klasse 393
- abstrakte Funktionen 390, 394
- Basisklasse 393
- dynamische Dispatch-Anschlüsse 386
- Fehlersuche 396
- kapseln 380
- Klasse 380
- Klassenhierarchie 397, 404
- LabVIEW-Beispiel 383
- Modularität 382
- Modulaustausch 397
- öffentliche (public) Methoden 380
- rechnen mit arabischen und römischen Zahlen 400
- Schutz Klassenbibliothek 405
- überschreiben einer Basisklasse 395

– Vererbbarkeit 381

– Vorteile 381, **382**

– wieder verwendbarer Code 382

Operatoren für Addieren usw. 46

Oszilloskopdiagramm (Scope-Modus) 128

P

Package Manager 218

– Eigene Pakete 220

– Nutzen 218

– Zusatzpakete 218

Palette 39

– ActiveX 411

– fixieren 42

– synchronisieren von Prozessen 264

– Voreinstellung 24, **41**

– Zahlendarstellung 63

Panel

– gestalten 94

– Panel/Diagr. neben-/untereinander 34

– Platz schaffen auf dem Panel 79

Parallel

– Messdatenerfassung und -verarbeitung
341

– parallele Ausführung von While-Schleifen 260

– Parallelverarbeitung 50

– quasi-parallele Verarbeitung 187

Parameter

– eines CLF-Knotens 212

– mit dem Programm speichern 52

– Parameterdarstellung eines Kreises 138

– Parameterdarstellung Zykloide 139

Passwort-Anzeige eines Strings 71

Path *Siehe* Pfad

PCI-Bus 318

Pfad 43, **162**

– aktueller Pfad zum VI und zur EXE-Datei 175

– erstellen, zerlegen 168

– Pfad zu einer Datei 162

– Pfad zum aktuellen VI 46, 169

– Pfadkonstante Standardverz. 169

– Pfadname einer Textdatei 168

– relative Pfadangabe 169

– String-Konvertierung 69

Pfeiltasten PC-Tastatur 39

- Phasengang 224
- Ping-Kommando im Internet 439
- Plus-Operator
 - Matrixaddition 76
 - Polymorphie 75, **105**
- Polling 279
- Polling, Beispiel für völligen Verzicht auf 284
- Polnische Notation 186
- Polymorphie
 - des Plus-Operators 75
 - Matrixaddition 75
 - von SubVIs 96
- Port, TCP/IP 439
- Positionieren von Objekten 40
- Private Daten einer Klasse 384
- Probedaten, Probes 41
- Produzent und Konsument 269
- Profil – Leistung und Speicher 378
- Programm, Programmieren
 - zeitsparend 94
- Programmablaufplan
 - Äquivalenz zum Zustandsdiagramm 359
 - Strukturen 48
- Programmieren
 - grafisch mit LabVIEW 22
 - in G bzw. LabVIEW 187
 - mit DataSocket 441
 - mit TCP/IP 443
 - mit VISA 295
 - objektorientiert 380
 - Programmiersprache APL 187
 - Schnittstellen zu anderen Programmiersprachen 187
 - strukturiert programmieren **48**, 123
 - Testen großer Programme 199
 - Übersetzen eines Programms 186
- Programmiersprache, universelle 20
- Programmierstil
 - Einstellungen Frontpanel und Diagramm 35
 - guter **94**, 118
- Programmstrukturen, vorgefertigte Muster
 - Siehe* Templates
- Projekt
 - für OOP 401
 - XControl 506
- Protokoll
 - verbindungsloses (IP) 439
 - verbindungsorientiertes (TCP) 439
- Protokolldatei 167
- Prozess
 - benötigte Prozessorzeit 262
 - echt paralleler Ablauf auf FPGA 465
 - Synchronisierung mit Datenaustausch 272
 - Synchronisierung ohne Datenaustausch 268
- Prozessor
 - eines Computers 186
 - Verringerung der Auslastung durch Ereignissteuerung 283
- Public (öffentlich) 123
- Punkt
 - Form, Größe 130
 - isolierte Punkte 130
- Q**
- Queue 273
 - auch Warteschlange 265
 - Datenübertragung mit Queues statt mit Meldern 274
 - FIFO-Prinzip bei Datenspeicherung 265
 - neuen Wert in Queue schreiben 373
- Quick Drop 215
- Quickdrop
 - Quickdrop Plugins 216
 - Ein Plugin erstellen 550
- Quittieren Eingabe 126
- R**
- Radix im Kontextmenü 71
- Raster 39
- Rechnen
 - mit Arrays 75
 - mit Clustern 85
 - mit numerischen Strings 106
- Reentrant **96**, **104**
- Referenz
 - auf Bedien- und Anzeigeelemente 150
 - auf LabVIEW-Objekte 44
 - Bedeutung 150
 - Schließen ist notwendig 110

- Vorteile 151
- Zahlenwert einer Referenz 150
- RefNum als Zahlenstring sichtbar machen 109
- Reihe
 - als DLL in C# programmiert 205
 - als DLL in C++ programmiert 204
 - in LabVIEW (in G) programmiert 204
 - mit MathScript programmiert 213
- Reihe als LabVIEW-Beispiel 50
- Reihenfolge
 - bei Cluster-Elementen 82
 - der Bits bei RS-232-Übertragung 291
- Rekursiver Aufruf 112
- Relation
 - Darstellung im XY-Graph 138
 - Relation und Funktion 138
- Relatives Zeitformat 64, 137
- Rendezvous 265, **271**
 - Beispiel mit zwei Schleifen unbestimmter Ausführungsdauer 271
 - Synchronisieren von drei Schleifen 272
- Ressource
 - Eingänge und Ausgänge wichtiger Funktionen 266
 - erzeugen, nutzen 264
 - freigeben, schließen 264
 - Name, Typ einer Ressource 266
 - Schonung durch Zeitbegrenzung (Timeout) 268
 - zur Synchronisierung 264
- Rezepte zum Arbeiten mit LabVIEW 32
- Ring
 - Anzeige-/Bedienelement 89
 - Bild-Ring 89
 - Datentyp 43
 - Ring und Case-Struktur 88
 - Unterpalette 'Ring & Enum' 56
 - Verknüpfung Zahl <--> Text (Bild) 88
 - zyklische Anordnung 88
- Ring & Enum **43**, 88
- Ringpuffer 340
- RIO 464
- Römische Zahlen 397
- Router, Aufgabe im Internet 439
- RS-232
 - Bitfolge 292
 - Bitmitte 291
 - Datenbits 291
 - Normung 291
 - Nullmodem 291
 - Startbit, Stoppbit 291
- Rückkopplungsknoten, Funktion als Schieberegister 247
- Rundungsfehler 251
- Runge-Kutta 244
- Runtime Engine 195
- Runtime-Bibliothek, Zweck 192
- S**
 - Scanfrequenz 227
 - größer als nach Shannon erlaubt 230
 - im Gegensatz zur Frequenz der abgetasteten Funktion 227
- Schaltverhalten, 6 Typen 66, 68
- Schieberegister **58**, 78
 - bei For-Schleife 58
 - Initialisierung 59
 - Speicherung interner Zustand 104
- Schleife, Struktur 45, 48, **57**
- Schlüssel einer Konfigurationsdatei 182
- Schnelleinfügeleiste 189, 215
 - Quick Drop 215
 - Shortcuts 216
- Schnittstelle, serielle 290
- Schutz Klassenbibliothek durch Passworteingabe 405
- Schwingung
 - Beispiel erzwungene Schw. 245
 - Feder-Masse-System 243
- Scope *Siehe* Oszilloskopdiagramm
- Scroll *Siehe* Streifendiagramm
- Sedezimalzahl *Siehe* Hexadezimalzahl
- Selektor
 - für polymorphes VI 106
 - mehrere Fälle zusammenfassen 56
- Selektor, Beschriftung 56
- Semaphor 263, 265, **269**
 - Ausführung verhindert, wenn Marke fehlt 269
 - Beispiel Produzent – Konsument 269
- Sensor
 - Temperatursensor TSR-CAN-03 304

- von Brüel & Kjaer für Beschleunigung 346
- Sequenz 48, **50**
 - Ausführung 52
 - Datenabhängigkeit 80
 - Erzwingung sequentieller Ablauf 161
 - flache 48
 - flache und gestapelte 357
 - zeitgesteuerte Sequenz 260
 - Sequenzstruktur
 - Nachteil 357
 - zum Aufbau eines VI 356
- seriell, Bit-serielle Ein-/Ausgabe 290
- serielle Schnittstelle
 - Öffnen und Schließen 293
 - Vorteil 290
- Server 437
 - LabVIEW-Beispiel mit TCP/IP 444
 - Server-Client auf einem oder auf verschiedenen PCs mit TCP/IP 443
 - Server-Client-Paar zum Nachführen eines Drehinstruments 441
- Shannon, Satz von 222, **229**
- Shared Variable *Siehe* Umgebungsvariable
- Shortcut
 - allgemein 24, **33**
 - Quick Drop Plugins 217
- Sichtbare Objekte bei Bedien- oder Anzeigeelementen 70
- SI-Darstellung von Gleitkommazahlen 64
- Signal
 - als Summe von Sinussignalen 222
 - Ein/Ausgabe, Unterpalette 43
 - periodisches 222
 - Richtung Signalfluss in LabVIEW-Diagrammen 259
- Signalverlauf *Siehe* auch Waveform
 - Signalverlaufs-Diagramm *Siehe* Chart
 - Signalverlaufs-Graph *Siehe* Graph
 - Unterpalette 46
- signifikante Stellen, Ziffern **54**, 63, 65
- Simulation, Geräteauswahl 329
- Simulink®
 - Beispiel eines Ergebnisdiagramms 258
 - Darstellung eines Differenzialgleichungssystems 257
- Sinus
 - Funktion 98, **128**
 - Sinuskurve 123
 - wandernde Sinuskurve 143
- Sinusgenerator *Siehe* cRIO-System
- Skala ausblenden 54
- Skalarprodukt
 - n-dimensionaler Vektoren 79
 - von Vektoren, einfaches Beispiel 77
- Skalieren
 - grafische Ausgabe 123
 - X-Achse 136
 - X-Achse mit Echtzeitwerten 144
 - Y-Achse 132
- Skripting
 - 9 Beispiele 536
 - modifizieren 'Leeres VI' 546
 - weiterführende Informationen 544
- Smartphone im LabVIEW-Webdienst 462
- Solaris 20
- Sondenüberwachungsfenster 41
- Spaltenindex einer Matrix 73
- Spannungsmessung
 - differentiell 334
 - NRSE-Verfahren 334
 - RSE-Verfahren 335
 - Wahl des Verfahrens 335
- Speicherersparnis bei Gleitkommazahlen 64
- Speichern eines VI 28
- Spektrum
 - Amplitudenspektrum 222
 - Phasenspektrum 224
- SQL (Structured Query Language)
 - Anweisungen 433, 435
 - Kurzübersicht 435
- Standard
 - aktuelle Werte als Standard 52
 - für Variablenwerte festlegen 72
 - Standarddatenverzeichnis 171
 - Standardeinstellung numerischer Datentypen 63
 - Standardverzeichnis ändern 170
- Standardsymbol eines VI 99
- Startbit bei RS-232-Schnittstelle 291
- Startseite von LabVIEW 22

statische Bindung

- Nachteile 108
- SubVIs 97

Steuerung des Case-Selektors mit Enum 55

Stopp 283

- in einer Ereignisstruktur 283
- Stoppbit bei RS-232 291

Störung

- durch weißes Rauschen 240
- eines Messsignals 222
- zufällige Störung 240

Streifendiagramm (Scroll-Modus) 128

Strikte Typdefinition 364

String 43, 46

- Anzeigetypen Normal-, '\'-Code-, Passwort-, Hex-Anzeige 70
- Konstante 69
- Pfadkonvertierung 69
- String & Pfad 68
- String-Länge 69
- Teil-String 69
- verknüpfen, verketteten 46, **69**
- Zahl-Konvertierung 69

String & Pfad, Unterpalette 43

Struktogramm 48

Strukturen, Unterpalette 45

Strukturierte Programmierung **48**, 123

SubVI

- ablaufinvariantes 112
- Anordnung der Anschlüsse 100
- Austausch während der Laufzeit 111
- Dummy-Anschlüsse 100
- dynamisch einbinden 97
- Erstellung 96
- Erstellung eines polymorphen VI 105
- Erzeugung aus bestehendem VI 102
- kompatible SubVIs 110
- platzieren 100
- statisch einbinden 97
- Verhalten SubVI-Fenster (frei verschiebbar, Standard, modal) 103
- verschiedene SubVIs mit gleichem Anschlussfeld 110
- Zahl der Anschlüsse 100

Suchen

- in der NI Developer Zone 215
- NI-Suchmaschine 214

Suchfenster öffnen 33

Suchfunktionen 188

Sweep *Siehe* Laufdiagramm

Symbol 35

- 'Eigenschaft für VI-Klasse' – 'Frontpanel' 153
- eines VI 99
- für ein VI gestalten 100
- Symbole der Werkzeugpalette 40

Symbole *Siehe* Icon

Symbolleiste als XControl 526

Synchronisierung

- mit Datenübertragung, Melder, Queues 264
- ohne Datenübertragung, Semaphor, Rendezvous, Occurrence 264
- Unterpalette Synchronisierungsfunktionen 265
- von Prozessen 260
- von Tasks und Threads 272

Systemzeit, Beschriftung der X-Achse 145

T

Tabelle, Liste & Tabelle, Unterpalette 43

Tabellenskalkulation mit CSV-Datei 408

Taktfrequenz cRIO-FPGA 465

Task

- bei simulierten Geräten 329
- Beispiele zur Simulation 330
- Definition 325
- erstellen 321
- programmatische Erstellung 333
- Task-Manager Microsoft Windows 262

TCP/IP

- auf Listener warten 445
- Aufgabe und Bedeutung von 439
- Lese-, Schreibfunktion 445
- Listener erzeugen **445**
- Prinzip 437

TDM-Datei 176, 177

TDMS-Datei, Palette 177

Technologische Abstraktion 190

TEDS

- Bedeutung, Vorteile 345
- Beispiel-VI 349
- Datenblatt, Konfiguration bei NI 346, 347

Templates 201, 376
Terminal *Siehe* Anschluss
Testen ablaufinvarianter VIs 113
Testen von Programmen *Siehe* Debuggen
Text- & Grafikring 88
Textdatei 162
Textring 56, **88**
Textsuche 33
Thread **262**
Tiefpass 234
– arithmetisches Mittel als Tiefpass 235
– Beispiel der Auswirkung auf ein gestörtes Signal 236
– Filtern im Frequenzbereich 241
– glättende Wirkung 236
– idealer Tiefpass 235
– im Frequenzbereich 241
– realer Tiefpass 236
Timerwert auslesen 50
Tipstrip 28
Toolchain 217
Tools Palette *Siehe* Werkzeugpalette
Top-Down-Methode 96, **358**
transparente Beschriftungsfelder 38
Transponieren einer Matrix **135**
Treiber **318**
– IVI-Treiber 350
– NI-DAQmx 318
– VISA-Treiber 318
Treppenfunktion 130
Triggern 337
TRUE 55
Tschebyscheff-Filter 238
Typ
– einer Variablen, Auffinden 43
– einer Variablen, numerisch 42
– Strikte Typdefinition 364
– U16 bei Ringen 88
Typdefinition
– als 'Bedienelement.' 115
– als 'Strikte Typ-Def.' 115
– als 'Typ-Def.' 115
– Beispiel Registerkarte 117
Typumwandlung 150

U
Überschreiben einer Datei 167
Übersetzen eines Programms 186
Uhrzeit und Datum 58
Umfärben 41
Umgebungsvariable
– beim cRIO 495
– Bibliothek **490**
– 'Data Binding' 494
– Datenaustausch zwischen Computern **492**
– Einstellungen 490
– Typ Einzelprozess 492
– Typ Netzwerk 493
Umordnen Clusterelemente 87
Umwandeln
– Anzeigeelem. <--> Bedienelem. 76
– Cluster <--> Array und umgekehrt 45
– Datentyp --> Variant 92
– String <--> Pfad 46
– Variant --> Datentyp 92
– von Daten in Strings 171
– von Format & Genauigkeit 66
– von ganzen Zahlen in Hex-Zahlen 64
Umweltbeispiel 257
Undo 34
Unterabtastung, Beispiel 231
Unterpalette 44
Unterprogramm *Siehe* SubVI
URL 447
USB
– Bedeutung **295**
– Experimentierplatine von Conrad Elektronik (Velleman) 295
– Geschwindigkeit Version 1.1 und 2.0 295
– in LabVIEW 290
– NI USB-6251 334
U-Typ, Integer ohne Vorzeichen 63

V
Variable
– globale 45, 274
– lokale 45, 269
– Namen ändern 55
– vertauschen 149

- Vektor 81
 - als 1-dimensionales Array 71
 - Dimension 71
 - leeren Vektor erzeugen 128
- Verbindung cRIO-Computer – FPGA 472
- Verbindungslinien, Drähte **31, 44**
 - doppelt ausgezogen 76
 - konstituierend für LabVIEW 186
- Verbindungspunkte an Kreuzungen 38
- verdeckter Code 356
- Verdrahtung, automatische 38, 39
- Vererbung 123, **152**
 - Beispiel 389
 - Hierarchie 391
 - in LabVIEW festlegen 390
 - zweistufige Vererbung 403
- Vergleich von DBL-Zahlen 183
- Vergleichen, Unterpalette 46
- Verkettung 75
- Verkettung von Arrays 75
- Verkettung von Strings 69
- Versionen von LabVIEW 20
- Vertauschen
 - mit Referenzen auf Variable 151
 - von Clustern, Arrays 151
 - von zwei Variablen 149
- Verzeichnis, Unterverzeichnis für 'FPGA Bitfiles' beim cRIO-System 489
- VI
 - als Unterprogramm oder als Hauptprogramm 100
 - dynamischer Aufruf 110
 - Einstellungen 102
 - ersten Aufruf anders behandeln 105
 - LabVIEW-Umgebung 192
 - laden 28
 - mit internem Zustand 104
 - öffnen, schließen 34
 - Ping.vi 440
 - polymorphes VI als Container 106
 - Programmbeispiel polymorphes VI 107
 - Referenz auf ein VI 109
 - speichern 28, 34
 - starten, stoppen 28
 - Steuerung von VI's 46
 - VI-Server 152, **189**
 - 'Wrapper'-VI 416
 - zur seriellen RS-232 DÜ 292
- VI-Analyzer 545
- VIPM 218
- VISA 292
- VI-Skripte 534
 - Funktionen 536
 - Funktionen – Abstand des neuen VI-Objekts von Referenzobjekt 539
 - Funktionen – GObject-Beschriftung abfragen 542
 - Funktionen – GObjects suchen 541
 - Funktionen – Klassenhierarchie mittels Klassennamen ermitteln 542, 544
 - Funktionen – Neues VI 536
 - Funktionen – Neues VI-Objekt 537
 - Funktionen – VI-Objektreferenz öffnen 539
- VI-Skripting 568
- Visual Basic, Erstellen Makro durch Aufzeichnen 424
- Vordergrundfarbe 54
- W**
- Warteschlange *Siehe* Queue
- Wartezeit
 - Enlastung des Prozessors 262
 - nutzbar in Schleifen 261
 - While-Schleife ohne 262
- Waveform 141
 - als spezieller Datentyp 46
 - Darstellung in einem Graphen 142
 - Elemente t0, dt, Y 141
 - Funktionen zum Bündeln und Aufschlüsseln 141
 - mehrere Kurven, ein Fenster 143
 - Parameter 143
- Web Browser und ActiveX 411
- Webdienst 447, 448
 - Abrufen auf Smartphone 457
 - Beispiele 448, 453, 456, 457
 - Darstellung als Text, HTML, JSON 451
 - in LabVIEW 447
 - Redirektion in LabVIEW 454
 - Schreiben in Browser 456, 457
 - Schreiben in Datei 455

- Starten 450
- Struktur 448
- Umstellungen auf Text, JSON, HTML 452
- Veröffentlichen 450
- Webserver 447
 - firmeninternes Netz 462
 - Internet 463
- Werkzeug zum Verbinden 99
- Werkzeugpalette **23**, 40
- Werkzeugtaste 'A' 76
- Wert (signalisierend) *Siehe* Ereignis
- Wert einstellen 40
- Wertebereich prüfen und erzwingen 183
- While-Schleife 48, **57**, 60
 - Ausgang 60
 - Bedingungsterminal 61
 - Fehlerausgang 160
 - im Zustandsdiagramm 359
 - Speicherung interner Zustand 104
 - Zahl der Durchläufe 61
- wieder verwendbarer Code 123, **382**
- Wiederherstellung früherer Programmierschritte 39
- Wiederholt ausführen (Modus) 29
- Windows 7, Windows 8 20
- WLAN 291
- Wrapper für Excelfunktion 416, 417
- X**
- XControl
 - als gekapselte OOP-Software 507
 - Beispiel Erstellung einer Symbolleiste 525
 - Beispiel Steuerung Flugbahn 504
 - benötigt ein aufrufendes VI, den 'Container' 511
 - Daten, Zustand, Fassade, Initialisieren 506
 - 'Daten_Pfeil.cti' 509
 - Eigenschaften 514
 - erstellen, allgemeines Rezept 506
 - 'Fassade_Pfeil.cti' 509
 - keine gleichzeitige Programmierung von XControl und Container 525
 - Leistungsmerkmale 507
 - Methoden 527
 - optionales Leistungsmerkmal 'Delnit' 508
 - optionales Leistungsmerkmal 'Status für Speichern umwandeln' **508**, 532
 - Projekt 506
 - Rahmen 'Anzeigezustandsänderung' 520
 - Rahmen 'Ausführungszustandsänderung' 518
 - Rahmen 'Datenänderung' 521
 - Rahmen 'Richtungsänderung' 519
 - sollte wieder verwendbar sein 513
 - Standardrahmen, ihre Bedeutung 517
 - 'Stein.vi' als Container 512
 - Unterschied zum Ctl 504
 - 'XControl_Pfeil.xcti' 505
 - Zeichnen eines 'echten' Pfeils als Aufgabe 523
 - 'Zustand_Pfeil.cti' 509
 - zwei LabVIEW-Instanzen 506
- Xilinx®-Compiler 465
- XML-Bibliothek
 - Aufbau 570
- XNET
 - arbeitet mit Datenbankformat FIBEX 305
 - Beispiel mit zwei Sensoren bei verschiedenen Abtastraten 314
 - CAN-Bus 304
 - FlexRay-System 304
 - Funktionen finden 304
 - LIN als Lowcost-System 304
 - National Instruments 304
 - NI cDAQ-9174 mit Einschub NI 8962 304
- XNode Template.vi 562
- XNodes
 - Ability-VIs 556
 - als XML-Bibliothek **556**
 - Beispiel Gestalt einer Kuh 553
 - Eintragung im VI-Server 556
 - Eintragung in LabVIEW.ini 556
 - Entwicklung von 557
 - Erscheinungsbild im Diagramm 553
 - Erweiterung von Funktionen im Diagramm 553
 - in Funktionspalette aufnehmen 571

- lava xnode manager 557
- 'leerer' XNode 557
- Nutzen 553
- Offizielle Unterstützung durch NI 554
- Programmierung einer Kuh 561
- Regelungstechnische Anwendungen 554
- Umbenennung von
Blank_XNode.xnode 557
- Vereinfachung Diagramm 555
- Verwendung bei NI 553
- wie ändert man sie? 571
- XML-Datei 559

XY-Graph 123, **138**

- Darstellung einer Zykloide 139
- mehrere Relationen in einem XY-
Graphen 140

Z

Zahl

- Basis 2, 8 oder 16 zur Darstellung 64
- Darstellung an Größe anpassen 155
- ganze Zahl, I-Typ und U-Typ 63
- Gleitkommazahl, DBL und SGL 63
- Integer vorzeichenbehaftet,
vorzeichenlos 66
- komplexe 63
- String-Konvertierung 69
- Zahl der Nachkommastellen 64
- Zahlensystem, Basis 64

Zahlenanzeige 70

Zähler mit For-Schleife 57

Zeichenkette 68

Zeilenindex einer Matrix 73

Zeit

- absolutes, relatives Zeitformat 64, **137**
- exaktere Durchlaufzeit 287
- momentane Uhrzeit und Datum 46, **69**
- Timing 69
- Zeit & Dialog 46
- Zeitbereich 222

- Zeitformat einer Zahl 64
- Zeitfunktionen 46
- Zeitgesteuerte Strukturen 287
- Zeitmessung 50
- Zeitschleife 287, 477
- Zeitstempel 145

ZILA Elektronik GmbH

- Sensor für CANopen 303
- Temperatursensor TSR-CAN-03 304

ZIP-Dateien 179

Zustand

- Beschreibung mit 'String' oder 'Enum'
359
- Zustandsauswahl mit 'Enum' 362
- Zustandsauswahl mit 'String' 361

Zustandsautomat

- Darstellung durch gerichteten Graphen
358
- Darstellung im Zustandsdiagramm 359
- mit Ereignisstruktur 372
- mit Queues 372
- Polling 371
- unter LabVIEW 357
- Verwendung von Konfigurationsdateien
368
- Voreinstellung 364
- Vorteil 358

Zustandsdiagramm

- Äquivalenz zum Programmablaufplan
359
- Folgezustand 368
- Leerzustand 371
- Münzautomat 363
- Umsetzung in LabVIEW 359
- Zähler 358

zweidimensionales Array oder Matrix 71

Zwischencode 186

zyklische Anordnung bei Ringen 88

Zykloide, Darstellung als XY-Graph 139