Wolfgang Georgi
Philipp Hohl

Einführung in LabVIEW

HANSER

Georgi/Hohl
Einführung in LabVIEW

Wolfgang Georgi • Philipp Hohl

Einführung in LabVIEW

6., erweiterte Auflage

Mit 1018 Bildern und 163 Aufgaben

Fachbuchverlag Leipzig
im Carl Hanser Verlag

Prof. Dipl.-Math. Wolfgang Georgi
Hochschule Ravensburg-Weingarten für Technik, Wirtschaft und Sozialwesen
M.Eng. Philipp Hohl
Hochschule Ravensburg-Weingarten für Technik, Wirtschaft und Sozialwesen

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

ISBN: 978-3-446-44272-6
E-Book-ISBN: 978-3-446-44407-2

Fachbuchverlag Leipzig im Carl Hanser Verlag

Internet: http://www.hanser-fachbuch.de

Lektorat: Franziska Jacob, M. A.
Herstellung: Dipl.-Ing. (FH) Franziska Kaufmann
Satz: Kösel Media GmbH, Krugzell
Covergestaltung: Marc Müller-Bremer, www.rebranding.de, München
Coverrealisierung: Stephan Rönigk
Druck und Bindung: Pustet, Regensburg
Printed in Germany

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 W. Georgi
Wolfegg P. Hohl

Inhalt

Teil I: Grundlagen des Programmierens in LabVIEW

Systematische Einführung in die wichtigsten Konzepte von LabVIEW. Das umfasst:

Installation und Aufruf von LabVIEW, grundlegende Arbeitsmittel wie Frontpanel, Diagramm, Paletten für Eingabe/Ausgabe und ihre Anpassung an Benutzerwünsche, ferner Bedienelemente und Funktionen, Datentypen, das Erstellen von Unterprogrammen, Visualisierungstechniken, Umgang mit Referenzen sowie das Schreiben und Lesen von Daten auf bzw. von der Festplatte. Kapitel 9 gibt eine Kurzübersicht über den Aufbau von LabVIEW und zusätzliche Hilfen zum Erlernen dieser grafischen Programmiersprache.

1 Was ist LabVIEW?

Lernziel

Der Leser soll anhand eines sehr einfachen Beispiels einen ersten Eindruck von LabVIEW, von der Idee der Datenflussprogrammierung und den wichtigsten Programmierkonzepten gewinnen. Er kann einfache VIs von Beginn an entwickeln.

1.1 Entwicklungsstufen

Software wurde und wird unter verschiedenen Aspekten geschaffen. Bekannt sind Begriffe wie 'strukturierte Programmierung', 'objektorientierte Programmierung' usw. In jüngerer Zeit spielt auch die **Prozessvisualisierung** eine zunehmende Rolle.

Ursache ist die ständig komplizierter werdende Technik. Sie verlangt bessere Darstellungsmöglichkeiten, damit der Anwender den Überblick nicht verliert. Man beschränkt sich heute bei der Abbildung technischer Prozesse nicht mehr allein auf konventionelle Anzeigeinstrumente, sondern stellt auch den Prozessablauf selbst auf dem Bildschirm eines Rechners grafisch dar. Es geht um Anschaulichkeit. Die Füllstandsanzeige eines Behälters wird z.B. nicht mehr nur durch ein analoges Manometer auf dem Bildschirm verkörpert, sondern durch die Zeichnung des Kessels selbst, in dem bunt gefärbt die Flüssigkeit auf- und absteigt. So kann auch der Laie erahnen, in welchem Zustand sich ein technischer Prozess gerade befindet. Um die Programmierung solcher grafischen Oberflächen mit anschaulichen, teilweise auch bewegten Bildern zu unterstützen, sind verschiedene Programmierwerkzeuge entwickelt worden.

Eines davon ist das Softwarepaket **LabVIEW** von National Instruments. LabVIEW ist die Abkürzung von Laboratory Virtual Instrument Engineering Workbench. Es ist **Entwicklungsumgebung** und **grafische Programmiersprache** zugleich.

Grafische Hilfsmittel in Papierform wie Programmablaufpläne oder Flussdiagramme gibt es schon seit langem, doch 'Zeichnen am Computer' wurde erst möglich, als die Rechner hinreichend leistungsfähig und Bildschirme als Ausgabe- und Eingabegeräte mit hoher Auflösung verfügbar wurden. Das geschah gegen Ende der 70er-Jahre. Zu der Zeit hatten zwei der Gründer der Messtechnikfirma National Instruments, Jim Truchard und Jeff Kodosky, die Idee, die Software zum Testen ihrer Messgeräte ähnlich wie diese selbst zu strukturieren. Sie nannten einzelne Bausteine deshalb virtuelle Instrumente oder **VI**s, eine Bezeichnung, die sich bis heute als Datei-Erweiterung eines jeden LabVIEW-Programms erhalten hat. Eine andere Idee bestand darin, die Programmierung nicht, wie bisher üblich, zeilenweise in Form von Anweisungen, genannt 'Statements', niederzuschreiben, sondern Funktionsblöcke in einem Blockdiagramm auf dem Bildschirm darzustellen. Dies gestaltet sich ganz so, wie man das auch früher schon mit Bleistift und Papier gemacht hatte.

Die Entwicklung der ersten lauffähigen LabVIEW-Version war eng geknüpft an das Aufkommen leistungsfähiger Personalcomputer, speziell des Macintosh von Apple. Anfang der 80er-Jahre bot nur dieser PC die grafischen Voraussetzungen zur Realisierung der mit LabVIEW verfolgten Ideen. Die erste LabVIEW-Version erschien 1986. Seitdem gab es folgende Entwicklungsstufen:

1986: LabVIEW 1.0 für den Macintosh II
1988: LabVIEW 1.2
1990: LabVIEW 2.0
1992: LabVIEW 2.5
1993: LabVIEW 3.0
1996: LabVIEW 4.0 (erst ab 1995 war Microsoft Windows so weit verbessert, dass LabVIEW auch unter diesem Betriebssystem lauffähig wurde)
1998: LabVIEW 5.0
2000: LabVIEW 6.0 (auch 'LabVIEW 6i' genannt)
2001: LabVIEW 6.1
2003: LabVIEW 7.0 (auch 'LabVIEW 7 Express' genannt)
2004: LabVIEW 7.1
2005: LabVIEW 8.0
2006: LabVIEW 8.2 (eigentlich LabVIEW 8.20 wegen damals 20 Jahren LabVIEW)
2007: LabVIEW 8.5
2008: LabVIEW 8.6
2009: LabVIEW 2009 (von da an wurde das Jahr zur Versionsnummer)
.
.
2013: LabVIEW 2013
2014: LabVIEW 2014

LabVIEW hat sich in den letzten Jahren stark verbreitet. Gleichzeitig hat es sich von einer anfangs messtechnisch orientierten zu einer universellen Programmiersprache entwickelt. National Instruments bietet LabVIEW inzwischen längst nicht mehr nur für das Betriebssystem MacOS von Apple-Rechnern an, sondern für eine Fülle anderer Systeme, von denen hier nur genannt seien:

Microsoft Windows 7 (32 Bit, 64 Bit)
Betriebssystem von Sun-Workstations (Solaris)
Linux und andere UNIX-Varianten

Die Aufzählung der verschiedenen Vorteile von LabVIEW sprengt den Rahmen dieses Abschnitts. Nur so viel sei einleitend erwähnt:

LabVIEW erlaubt die Anbindung an gängige Programmiersprachen. Damit wird unter LabVIEW z.B. die Nutzung von C-Code möglich. Von LabVIEW 8.2 an wurde die objektorientierte Programmierung erleichtert.

Heutzutage ist das Internet aus unserem Leben nicht mehr wegzudenken. Konsequenterweise verfügen die heutigen Versionen von LabVIEW über Module, welche die Anbindung an das Internet erleichtern und so z.B. die **Fernüberwachung** von Maschinen und Anlagen erlauben.

1.2 Was will dieses Lehrbuch?

Das vorliegende Lehrbuch führt in die Programmierung mit LabVIEW ein. Es setzt voraus, dass der Leser die Beispiele und Übungen am PC durcharbeitet.

Die neueste Ausgabe von LabVIEW ist die Version 2014 (zum Zeitpunkt der Drucklegung dieses Buches). Die Bilder in diesem Buch und die Beispiele sind durchgängig auf LabVIEW 2014 unter Windows 7 abgestellt.

Man kann davon ausgehen, dass derzeit noch viele Leser Zugang zu einem PC mit den älteren Versionen haben, z.B. wenn sie Mitarbeiter einer Firma sind und sich in die Behandlung messtechnischer Probleme mit LabVIEW einarbeiten müssen. Auch Besitzer von älteren LabVIEW-Versionen können dieses Buch zu Rate ziehen, denn die Unterschiede bei den einführenden Beispielen sind nicht sehr groß. Programme, die mit den Versionen 8.2, 8.6, 2009 erstellt wurden, laufen auch unter der Version 2014.

Doch trifft das Umgekehrte naturgemäß nicht zu, weil jede neue Version auch neue Möglichkeiten bietet. Braucht man diese allerdings nicht, kann man zu einer früheren Version zurückgehen, indem man z.B. bei einem unter Version 2014 entwickeltes Programm 'Für vorige Version speichern...' anklickt und anschließend eine der Versionen 2013 bis 8.2 aussucht. Das Programm lässt sich dann mit der gewählten Version aufrufen.

1.3 Installation

Die Installation der LabVIEW-Version 2014 von DVD ist selbsterklärend. Als Betriebssysteme sind z.B. Microsoft Windows XP, Microsoft Windows 7, ein Apple-Betriebssystem oder Linux geeignet. Die Hardware des PC muss den Anforderungen des jeweiligen Betriebssystems entsprechen. Alle Beispiele in diesem Buch wurden unter den Betriebssystemen Windows 7 (32 Bit) und, soweit möglich, unter Linux getestet.

1.4 Einführendes Beispiel

Angenommen, Sie wollen die Summe

$$c = a + b$$

berechnen. Programmiersprachen wie C, C++, C# und ihre Vorläufer sind so konzipiert, dass sie genau diese Art von Aufgaben perfekt lösen können. Man schreibt einfach:

```
c = a + b;
```

und muss also anscheinend nur das Semikolon hinzufügen.

Doch übersieht man dabei Eingabe und Ausgabe. Der Anwender will Werte für *a* und *b* eingeben und am Bildschirm das Ergebnis ablesen. Fügt man die entsprechenden Programmteile hinzu, ist ein C-Programm längst nicht mehr so übersichtlich.

LabVIEW verringert diese Schwierigkeiten mit zwei Methoden:

- **grafische** Programmierung nach dem **Datenflussprinzip,**
- Verwendung umfangreicher **Funktionsbibliotheken** für Ein- und Ausgabe.

Bild 1.1 macht deutlich, dass Eingabe, Ausgabe und mathematische Operationen nach dem Datenflussprinzip organisiert sind. Das ist hier am Beispiel $c = a + b$ erklärt. Die folgende Skizze in Bild 1.2 reduziert dieses Prinzip auf seinen Kern.

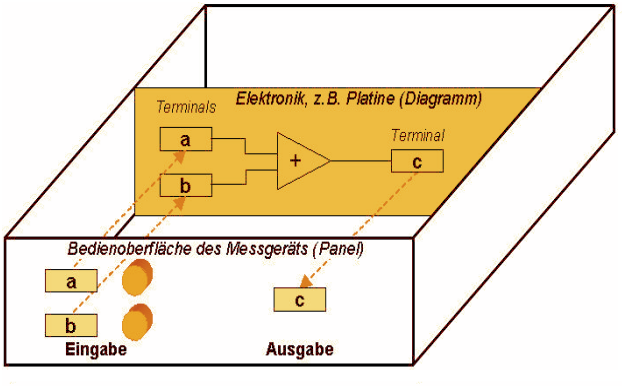

Bild 1.1 Idee der Simulation eines realen Messgeräts mit Hilfe von (Front-)Panel und (Block-)Diagramm in LabVIEW

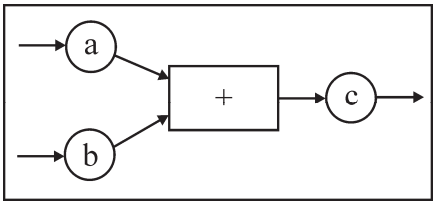

Bild 1.2 Idee der Datenflussprogrammierung am Beispiel von $c = a + b$. Links die Eingabe, rechts die Ausgabe

Aufruf von LabVIEW

Ein linker Mausdoppelklick auf das LabVIEW-Icon auf dem Desktop öffnet eine Startseite, im Beispiel dargestellt für die Version 2014. Man kann das aber auch mit 'Programme' – 'National Instruments' – 'LabVIEW 2014' im Windows-Startmenü erreichen. Dann öffnet sich nach einigen Sekunden das in Bild 1.3 dargestellte Fenster.

Dort erlaubt das Feld 'Neu' – 'Leeres VI' die Anfertigung eines LabVIEW-Programms. Mit 'Öffnen' kann man ein bereits existierendes VI von der Festplatte laden, z.B. um es zu modifizieren. 'Beispiele' – 'Beispiele suchen…' ermöglicht das Erlernen von LabVIEW anhand von Beispielen. Hilfreich sind auch die Rubriken 'Hilfe' – 'Erste Schritte mit LabVIEW'. Für Fortgeschrittene, die sich bereits mit einer älteren Version auskennen, sind 'Hilfe' – 'Liste aller neuen Funktionen' und 'Online-Unterstützung' nützlich.

Wählt man nun 'Neu…' – 'Leeres VI', so erscheinen zwei Fenster: das eine mit dem Titel 'Unbenannt 1 Frontpanel', das andere mit dem Titel 'Unbenannt 1 Blockdiagramm'. Sie sind in Bild 1.4 und Bild 1.5 dargestellt.

Bild 1.3 Startfenster, das sich beim Aufruf von LabVIEW 2014 öffnet

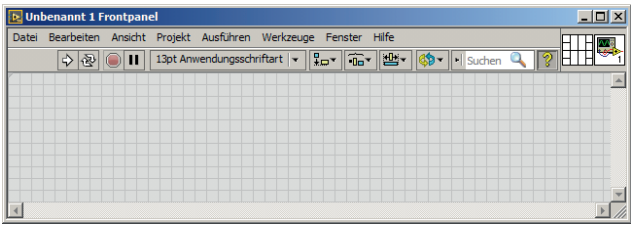

Bild 1.4 Frontpanel oder kürzer 'Panel', für die Benutzeroberfläche des späteren Programms

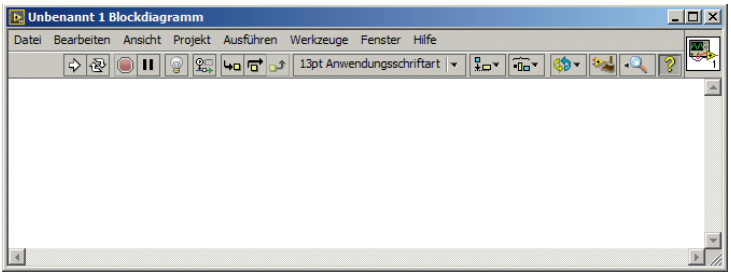

Bild 1.5 Blockdiagramm oder kürzer 'Diagramm', zur Programmierung und grafischen Darstellung des Programms

Wichtig für die Programmierung ist die 'Werkzeugpalette' nach Bild 1.6. Sie erscheint bei entsprechender Voreinstellung automatisch. Wenn nicht, kann man sie vom Panel oder vom Diagramm aus mit 'Ansicht' – 'Werkzeugpalette' holen. Zur Programmerstellung brauchen Sie ferner die Elementepalette gemäß Bild 1.7 und die Funktionenpalette nach Bild 1.8.

Bild 1.6 Werkzeugpalette

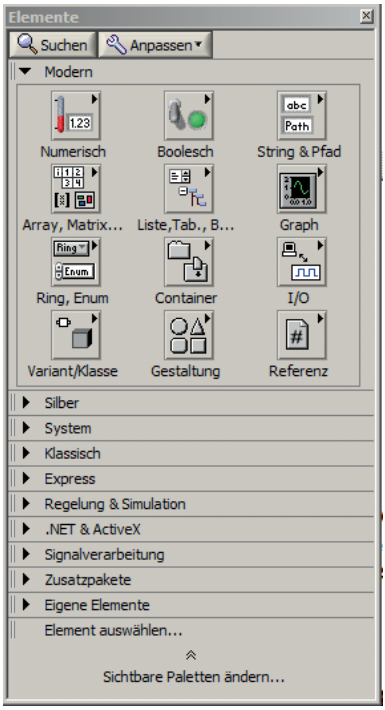

Bild 1.7 Palette mit 'Elementen', die zur Gestaltung des Panels genutzt werden. Hier wird die Palette aus der Kategorie 'Modern' gezeigt. Voreingestellt ist allerdings die Kategorie 'Express'. Doch lässt sich leicht von einer Einstellung zur anderen wechseln.

Man kann auch statt der voreingestellten Palettenkategorie oder zusätzlich zu ihr weitere Kategorien wählen und öffnen wie z.B. 'Silber', 'System' oder 'Klassisch'.

Man erhält die Paletten aus Bild 1.7, indem man im **Frontpanel** (Bild 1.4) mit der **rechten** Maustaste auf die freie Fläche klickt. Mit der **linken** Maustaste kann man diese Palette dauerhaft sichtbar machen, indem man sie 'anpinnt', d.h. die kleine Reißzwecke links oben im zugehörigen Loch versenkt. Danach kann man mit dem Doppelpfeil unten weitere Kategorien anzeigen und gegebenenfalls zu einer von ihnen durch Linksklick wechseln. Erneuter Linksklick lässt die Elemente einer geöffneten Kategorie verschwinden. Auf diese Weise kann man die Ansicht von Bild 1.7 gewinnen.

Die Palette der Funktionen in Bild 1.8 erhält man, indem man im **Diagramm** (Bild 1.5) mit der **rechten** Maustaste auf die freie Fläche klickt.

Man kann diese Paletten auch unter 'Ansicht' – 'Elementepalette' im Panel bzw. unter 'Ansicht' – 'Funktionenpalette' im Diagramm finden. Manche Fenster lassen sich ebenso mit **Shortcuts** öffnen, etwa das Blockdiagramm vom Panel aus mit <Strg>+<E> oder das Panel vom Blockdiagramm aus mit denselben beiden Tasten, die gleichzeitig zu drücken sind.

In den folgenden Abschnitten werden wir uns meist auf die Teilpalette 'Programmierung' beziehen. Man kann sie **dauerhaft** zur voreingestellten Palette machen, indem man aus einem VI heraus die Elementepalette (bzw. Funktionspalette) aufruft und nach Anpinnen mit 'Anpassen' – 'Sichtbare Paletten ändern…' ein Fenster öffnet, in dem man eine **einzige** Kategorie markiert und alle anderen nicht. Beim Rechtsklick auf das Frontpanel erscheint dann später **nur diese** Palette ganz oben und bereits geöffnet. Entsprechendes gilt für die Funktionenpalette.

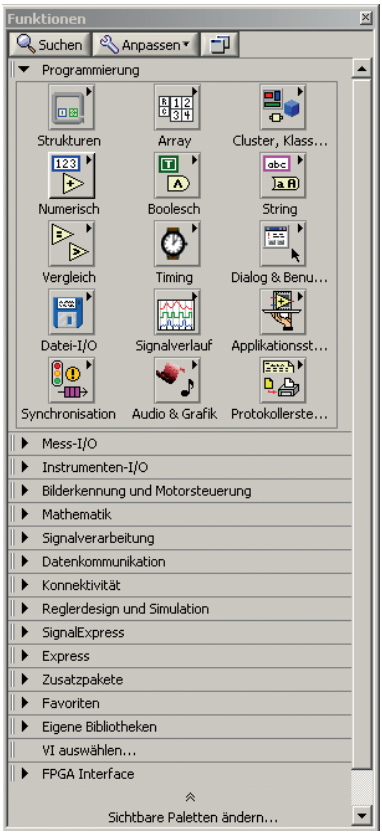

Bild 1.8 Palette mit Funktionen, mit deren Hilfe das Programm im Diagramm erstellt wird. Hier ist die Teilpalette 'Funktionen' – 'Programmierung' geöffnet. Die Voreinstellung bei Version 2014 ist wieder 'Express'. Die letzte Zeile wird nur angezeigt, wenn man vorher das Zusatzmodul 'FPGA Interface' mit installiert hat.

Der Anwender kann leicht von einer Palette auf die andere umschalten. Auch lassen sich zusätzlich andere Unterpaletten wie 'Mess-I/O', 'Mathematik', 'Express' usw. öffnen

1.4.1 Programmierung von c = a + b

Folgende Schritte sind auszuführen:

1. Kontrollieren, ob 'Bearbeitungsmodus' eingestellt ist. Das sieht man an den Symbolen (Icons) unter der Menüzeile im Frontpanel. Im Bearbeitungsmodus ist dort das Suchfenster zu sehen. Fehlt es, befindet man sich im 'Ausführungsmodus', der keine Programmierung erlaubt, sondern anzeigt, wie das Frontpanel während der Ausführung aussehen wird. Zur Umstellung 'Ausführen' – 'In Bearbeitungsmodus wechseln' anklicken oder <Strg>+<M> drücken.
2. Eingabefelder für die Variablen *a* und *b* anlegen. Zu diesem Zweck 'Elemente' gemäß Bild 1.7 holen, indem man dort Cursor zum Icon links oben bewegt. Es erscheint die Überschrift 'Numerisch'. Ein Mausklick führt zur nächsten Unterpalette. Wiederum das Icon links oben ('Numerisches Bedienelement') anklicken und aufs Panel ziehen. Das Ergebnis dieser Operation sieht man in Bild 1.9. Gleichzeitig verändert sich automatisch das Diagramm durch ein zugeordnetes 'Terminal', siehe Bild 1.10.

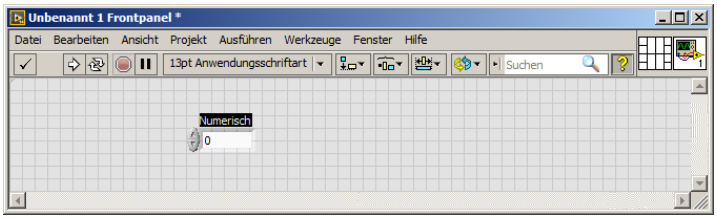

Bild 1.9 Panel mit numerischem (Bedien-)Element

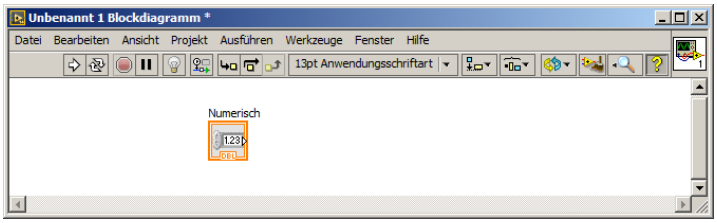

Bild 1.10 Diagramm mit 'Terminal', das dem (Bedien-)Element auf dem Panel entspricht

Das 'DBL' im Symbol von Bild 1.10 bedeutet, dass die Eingabedaten vom Typ 'Double Precision' sind (Gleitkommazahlen doppelter Genauigkeit). Voreingestellt ist auch die Darstellung des Symbols als Quadrat. Wir können jedoch auch auf eine platzsparende Darstellung umschalten, indem wir mit Rechtsklick auf dieses Symbol das Kontextmenü öffnen und dort die Markierung vor 'Als Symbol anzeigen' entfernen. Dann erhalten wir ein kleineres Rechteck. Diese Darstellung können wir auch dauerhaft einstellen, siehe dazu Kapitel 2, Bild 2.4. Wir verwenden für Terminals stets Rechtecksymbole.

3. Das Eingabefeld in Bild 1.9 dient zur Zahleneingabe über die Tastatur. Man kann die Zahlenwerte aber auch mit den Aufwärts-Abwärts-Pfeilen am linken Rand des Bedienelements ändern.
4. Wie schon erwähnt, wird das zugehörige Terminal im Diagramm automatisch gebildet. Man kann es als Darstellung der Durchführung auffassen, die in einem realen Messinstrument vom Gehäuse zur Platine führt und den vom Benutzer eingestellten Wert an die elektronische Schaltung weitergibt. Siehe dazu nochmals Bild 1.1.

Merke: **Die Wahl des Datentyps DBL ist voreingestellt. Wir werden später sehen, dass es viele andere Datentypen gibt.**

Wir können die Beschriftung 'Numerisches Element' für die Eingabevariable in Bild 1.9 durch einen eigenen sinnvollen Namen ersetzen, z.B. durch 'a'. Das kann entweder unmittelbar nach der Platzierung des Elements auf dem Panel erfolgen, wenn die Schrift noch schwarz unterlegt ist, siehe Bild 1.9. Oder man muss, wenn man die Beschriftung **später** ändern will, in der Werkzeugpalette das große **'A'** wählen, die Beschriftung mit der Maus schwarz einfärben und dann mit der gewünschten Bezeichnung überschreiben. Eine zweite Möglichkeit besteht im Links-Mausdoppelklick auf die Beschriftung, was der Wahl von 'A' entspricht. Das Ergebnis ist in Bild 1.11 dargestellt.

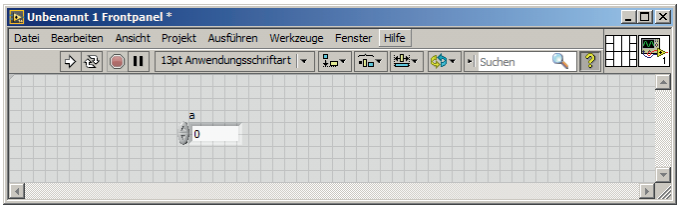

Bild 1.11 Panel mit Bedienelement für die Variable *a*

5. In gleicher Weise geht man mit der Variablen *b* um.
6. Dagegen muss man *c* als Ausgabevariable (Anzeigeelement) etwas anders behandeln, weil ihr Wert nicht vom Anwender gewählt, sondern vom LabVIEW-Programm errechnet wird. Man erhält sie in der Unterpalette der Palette von Bild 1.7 unter 'Numerisch' – 'Numerisches Anzeigeelement'. Bild 1.12 zeigt das Ergebnis.

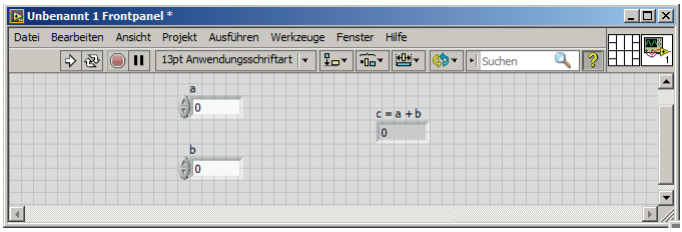

Bild 1.12 Panel mit zwei Eingabevariablen *a* und *b* sowie der Ausgabevariablen *c* = *a* + *b*

7. Haben Sie versehentlich ein falsches Element platziert, müssen Sie es löschen. Dazu das rechteckige Feld in der ersten Zeile der Werkzeugpalette anklicken und auf die Farbe Grün stellen, was Automatikbetrieb bedeutet. Je nach Anwendung wählt nun LabVIEW automatisch das der jeweiligen Aufgabe entsprechende Werkzeug. In diesem Fall das falsch gesetzte Symbol mit ständig gedrückter linker Maustaste umfahren. Es bildet sich ein gestricheltes Rechteck, dessen Ränder blinken. Nun mit der <Entf>-Taste löschen. Allgemein gilt: Fehler lassen sich mit 'Bearbeiten' – 'Rückgängig…' oder mit <Shift>+<Z> korrigieren. Zurück zum alten Zustand mit 'Wiederherstellen…' oder mit <Strg>+<Shift>+<Z>.
8. Verknüpfen der Eingabe- mit der Ausgabevariablen über die gewünschte Funktion 'Addieren'. Das geschieht im Diagramm. Zunächst wählen Sie in der Funktionenpalette von Bild 1.8 das Symbol 'Numerisch' (Zahlen 123 und Pluszeichen). Die Unterpalette zeigt Symbole, wie man sie von elektrischen Schaltplänen her kennt. Hier ist der Plusoperator links oben anzuklicken und ins Diagramm zu ziehen, siehe Bild 1.13.

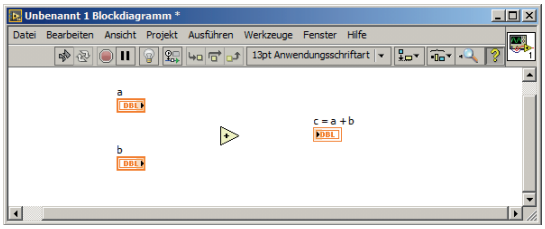

Bild 1.13 Diagramm mit Additionsfunktion in der Mitte

9. Nun sind die Symbole noch mit Drähten zu verbinden, wie schon in Bild 1.2 angedeutet wurde. Dazu dient die Drahtrolle in der Werkzeugpalette. Entweder wählt man diese

Rolle durch Mausklick oder man verlässt sich auf den Automatik-Modus der Werkzeugpalette. Die Drahtrolle wird wirksam, wenn man sich entweder einem Terminal oder einer Funktion nähert. Jedes Icon streckt dann kleine Fühler aus, die Anschlüsse der Funktion. Berührt man mit dem Mauszeiger einen dieser Anschlüsse, verändert er seine Form und wird zu einer kleinen Drahtrolle. Drückt man nun dort die linke Maustaste und bewegt die Maus, so zieht man eine gestrichelte Linie hinter dem Mauszeiger her, mit der sich die Icons verbinden lassen. Die Verbindung ist hergestellt, sobald man an einem der Fühler des zweiten Icons die Maustaste loslässt. Die Wegeführung des Drahtes ist ebenfalls zu beeinflussen, indem man an beliebigen Zwischenpunkten die Maus kurz loslässt. Dieser Punkt ist dann fixiert, und der Programmierer kann die Drahtrichtung ändern. Das Endergebnis ist in Bild 1.14 zu sehen.

10. Falsche Verbindungsleitungen kann man löschen, indem man sie hinreichend weit von den Anschlüssen entfernt anklickt und dann auf <Entf> drückt. Dabei wird meist nur ein Teil der Verbindungslinie gelöscht. Alle restlichen Teile beseitigt man mit dem Shortcut <Strg>+<B>. Das ist einfacher, als alle Teillinien einzeln mit <Entf> zu löschen.

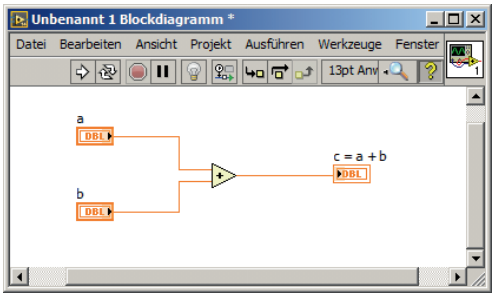

Bild 1.14 Diagramm des fertigen Programms zur Berechnung von $c = a + b$

1.4.2 Speicherung als Programm Add.vi

Das fertige Programm sollte jetzt unter einem einprägsamen Namen auf der Festplatte gespeichert werden. Zum Beispiel könnte man unser Programm mit 'Datei' – 'Speichern unter...' mit dem Namen 'Add.vi' ablegen. Die Datei-Erweiterung 'vi' ist erforderlich, wenn man das Programm später aus einem Ordner heraus mit Mausdoppelklick aufrufen möchte. Sie wird automatisch angehängt und muss nicht mit eingegeben werden. Hier speichern wir das Programm der besseren Auffindbarkeit wegen als '0115-Add.vi'.

1.4.3 Starten und Stoppen von Add.vi

Das Programm kann in den Hauptspeicher geladen werden:

- durch Doppelklick auf seinen Namen in dem Ordner, in dem es gespeichert ist,
- vom LabVIEW-Startfenster aus mit 'Öffnen…' und Pfadwahl,
- von einem anderen, bereits geöffneten VI aus mit 'Datei' – 'Öffnen…' und Pfadwahl.

Sobald Add.vi geladen ist, kann man es vom Panel aus auf dreierlei Wegen starten:

- Durch Anklicken des in der Symbolleiste ganz links stehenden Pfeils. Dann läuft das Programm genau einmal und stoppt dann.
- Durch Anklicken von 'Ausführen' – 'Starten'. Gleiche Wirkung wie oben.

- Durch Anklicken des zweiten Icons links oben mit der Bezeichnung 'Wiederholt ausführen', das zwei verschlungene Pfeile zeigt. Jetzt wird das Programm ständig ausgeführt, und zwar so lange, bis es der Anwender mit Hilfe des dritten Icons mit dem roten Stoppzeichen anhält.

Bild 1.15 zeigt das Programm Add.vi im Modus 'Wiederholt ausführen', wobei der Anwender als Eingabedaten die Werte $a = 7$ und $b = -3$ eingestellt hatte. Er kann diese Werte während des Programmlaufs beliebig verändern. Dazu klickt man das gewünschte Eingabefeld an und ändert den Variablenwert entweder mit den Aufwärts-Abwärts-Pfeilen oder man klickt direkt ins Eingabefeld und gibt den Wert über die Tastatur ein.

Hinweis: Der Modus 'Wiederholt ausführen' sollte möglichst vermieden werden. Besser lässt man das VI in einer Schleife ablaufen. Näheres siehe Abschnitt 3.4, While-Schleife

Solange im letzteren Fall die Eingabe noch nicht abgeschlossen ist, erscheint in der Symbolleiste links vom Startsymbol ein weiteres Icon mit einem kleinen Haken. Sobald man diesen Haken mit der Maus anklickt, betrachtet LabVIEW die Eingabe als beendet und rechnet von dem Moment an mit dem neuen Variablenwert. Statt den Haken anzuklicken, kann man aber einfacher mit der Maus unmittelbar neben das Eingabefeld klicken oder die Returntaste betätigen. Auch in diesem Fall verschwindet das Icon mit dem Haken.

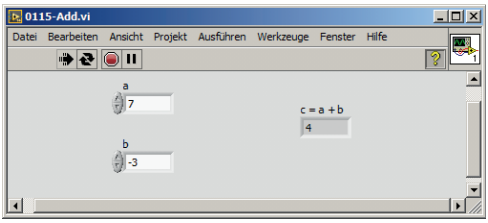

Bild 1.15 Panel im Modus 'Wiederholt ausführen'

1.4.4 Fehlersuche in Add.vi (Debugging)

Die Fehlersuche in diesem VI erscheint unnötig. Das Programm ist zu einfach. Doch lässt sich das Prinzip gut erklären: Zum Debuggen, d.h. Fehler suchen, geht man ins Diagramm und klickt auf das Icon mit der Glühlampe (fünftes Symbol von links, Bezeichnung 'Highlight-Funktion'). Die Lampe färbt sich dann gelb. Ein erneuter Mausklick macht sie wieder weiß. Im gelben Zustand verzögert die Lampe den Programmlauf, so dass man die Datenströme mit bloßem Auge verfolgen kann. Dazu zeigt Bild 1.16 einen während des Debuggens aufgenommenen Schnappschuss. Um den Ablauf noch genauer verfolgen zu können, klickt man auf das Pause-Icon in der Symbolleiste. Damit wird eine Pause erzwungen.

Danach kann man das Programm mit der Pfeiltaste zwei Kästchen rechts von der Glühlampe in Einzelschritten ausführen. Die zwei zusätzlichen Pfeiltasten dienen zum Überspringen von Unterprogrammen bzw. zum Verlassen des VI. Im Beispiel von Bild 1.16 sieht man zwei kleine Kugeln, die an den Terminals für a und b starten und nach rechts laufen. Sie repräsentieren die Datenströme. Im Moment befinden sie sich innerhalb des Additionssymbols. Die Momentanwerte von a und b werden ebenfalls angezeigt. Das Ergebnis $c = 4{,}00$ erscheint einen Moment später rechts vom Additionssymbol. **Ausprobieren!**