

Vorwort

LabVIEW hat sich in den Ingenieur- und Naturwissenschaften längst als Standard etabliert, insbesondere in den Bereichen Mess-, Steuer- und Regelungstechnik, der Simulation sowie der Datenerfassung, Datenanalyse und Datenvisualisierung. Dementsprechend vielfältig sind die Einsatzbereiche in der Fertigung, Forschung und Entwicklung sowie in der Lehre und dies weitgehend unabhängig vom jeweiligen Fachgebiet, wie z. B. Physik, Chemie, Medizin oder den Ingenieurwissenschaften.

Der fundamentale Unterschied zu textbasierten Programmiersprachen wie z. B. C++ besteht darin, dass LabVIEW eine visuelle Programmiersprache ist, bei der der Quellcode mit Hilfe graphischer Elemente erzeugt wird. Dieser Ansatz basiert auf der Erkenntnis, dass Menschen Bilder schneller verarbeiten können als Texte. Bilder werden im menschlichen Gehirn in der rechten Hälfte parallel verarbeitet werden, während textbasierte Informationen in der linken Hälfte sequenziell und damit deutlich langsamer verarbeitet werden. Aus diesem Grund erfolgt ja auch in vielen Bereichen des täglichen Lebens die Kommunikation auf der Grundlage von Graphiken, seien es Verkehrsschilder, Wetterkarten, Stadtpläne, Musiknoten oder der *Desktop* eines PC.

► Anmerkung

In der Informatik wird für die Software-Entwicklung mit Hilfe graphischer Elemente der Begriff „visuelle Programmierung“ verwendet. Mit „graphischer Programmierung“ wird dagegen die Darstellung und Manipulation (geometrische Darstellung, Rasterung) von graphischen Objekten auf einer Ausgabeeinheit (z. B. Monitor, Drucker) bezeichnet. In den Ingenieurwissenschaften hat sich jedoch der Begriff „graphische Programmierung“ für das Erstellen von Programmen mittels graphischer Elemente eingebürgert. Da sich diese beiden Bezeichnungen im allgemeinen Sprachgebrauch vermischen, werden sie hier synonym verwendet. Um Missverständnisse zu vermeiden, ist aber der Begriff „visuelle Programmierung“ vorzuziehen. ◀

In den letzten Jahren hat sich die Informatik immer stärker zu einer Querschnittsdisziplin mit großem Einfluss auf alle ingenieur- und naturwissenschaftlichen Bereiche entwickelt. Damit werden Naturwissenschaftler und Ingenieure, aber auch Techniker in ihrer beruflichen Praxis mit Aufgaben konfrontiert, deren Komplexität stetig zunimmt und die einen ausgeprägten Bezug zur Software-Technik aufweisen. Traditionell erhalten diese Berufsgruppen jedoch keine angemessene Ausbildung und sind bei der softwaretechnischen Umsetzung ihrer Aufgaben häufig auf sich allein gestellt. Weiterhin soll qualitativ hochwertige Software in immer kürzerer Zeit produziert werden. Diese beiden Anforderungen, hohe Software-Qualität einerseits und kurze Entwicklungszeiten andererseits, können aber prinzipiell nicht gleichzeitig erfüllt werden.

Nur weil die Produktzyklen und dementsprechend die Entwicklungszeiten immer kürzer werden, können die am Projekt beteiligten Menschen nicht automatisch schneller denken. Daher ist in der Informatik eine Vielzahl von Methoden und Hilfsmitteln entwickelt worden, um zumindest die Umsetzung der Aufgabenstellung in Software zu vereinfachen und zu beschleunigen. Praktisch alle Methoden und Konzepte für die Systemanalyse verwenden Graphiken für die Darstellung komplexer Sachverhalte.

Die einzige Ausnahme stellt die Programmierung selbst dar, bei der ein graphischer Entwurf anschließend mit Hilfe einer textbasierten Programmiersprache umgesetzt wird. Hilfreich wäre es, wenn der graphische Entwurf gleichzeitig das ausführbare Programm darstellen würde. Dann würde bei der Software-Entwicklung ein wesentlicher Entwicklungsschritt, die zeitraubende und fehlerträchtige Übersetzung des Entwurfs in eine textbasierte Programmiersprache, vollständig entfallen (beispielsweise wird dafür zurzeit *MDA (Model Driven Architecture)* mit *executable UML (Unified Modeling Language)* entwickelt).

Mit dem Konzept der strukturierten Datenflussprogrammierung, welches LabVIEW zugrunde liegt, wird diese Anforderung bereits in idealer Weise erfüllt. Vermutlich besteht der größte Nachteil der Programmiersprache LabVIEW darin, dass sie leicht zu erlernen ist und auch die Vermarktung als Messtechnik-Tool (die u. a. dazu führt, dass LabVIEW als „Laborsoftware“ wahrgenommen wird) verspricht schnelle Erfolge. Dabei wird häufig außer Acht gelassen, dass LabVIEW, genau wie C++ oder Java, eine *general purpose language* ist und dass für das Erlernen die entsprechende Zeit benötigt wird.

Erfahrungsgemäß lassen sich mit LabVIEW innerhalb weniger Stunden erste Resultate bei der Programmentwicklung und Gerätesteuerung erzielen. Anschließend ist aber häufig zu beobachten, dass bei den in aller Regel folgenden Erweiterungen und zunehmender Komplexität der Aufgabenstellungen die zugrunde liegenden Konzepte der Software-Lösungen den gestiegenen Anforderungen nicht mehr standhalten können.

Deshalb soll hier versucht werden, einen Beitrag zur systematischen Software-Entwicklung mit LabVIEW zu leisten, der nicht die messtechnische Aufgabe oder eine Anwendung in den Vordergrund stellt, sondern die Software-Entwicklung vor allem aus der Sicht der Informatik berücksichtigt. Unter anderem werden Datentypen und Datenstrukturen sowie die Kontrollstrukturen der strukturierten Programmierung und endliche Automaten ausführlich behandelt, da diese für eine effiziente Implementierung von Algorithmen von grundlegender Bedeutung sind.

Dementsprechend richtet sich das Buch nicht nur an Studierende der Informatik, sondern vor allem an Studierende der Ingenieur- und Naturwissenschaften, aber auch an Wissenschaftler sowie Ingenieure und Techniker in der Praxis. Gemäß dem alten LabVIEW-Motto

It's OK to have fun!

habe ich die Hoffnung, mit diesem Buch etwas zum Vergnügen bei der Arbeit mit LabVIEW beizutragen, weil sich die Freude bei der Arbeit auch immer in der Qualität der Arbeit niederschlägt.

Dank

Die Erstellung des Manuskriptes war etwas aufwendiger als erwartet und wäre ohne vielfache Unterstützung sicher nicht möglich gewesen. Bedanken möchte ich mich daher bei allen, die zum Entstehen des Buches beigetragen haben. Insbesondere bei allen Freunden, die ich lange vernachlässigt habe, von denen ich aber trotz alledem viel Aufmunterung und Zuspruch erhalten habe.

Zu großem Dank bin ich dem Team der deutschen Niederlassung von National Instruments in München verpflichtet für die kontinuierliche Hilfe und Unterstützung über viele Jahre, beispielsweise durch Marc Backmeyer, Dipl.-Ing. Markus Solbach und besonders durch Dipl.-Ing. Philipp Krauss, der die gute Ausstattung des Buches mit der deutschen und englischen LabVIEW-Studentenversion besorgt hat und darüber hinaus die gesamte Entstehung des Buches sehr eng begleitet und tatkräftig unterstützt hat, was entscheidend zur Entstehung beigetragen hat.

Für die Unterstützung bei mathematischen Fragestellungen danke ich Prof. Dr. rer. nat. Hardy Moock (FH SWF, Iserlohn) und Prof. Dr. phil. nat. Peter Dörre (FH SWF) für ein innovatives Gleichheitszeichen. Die engagierte Betreuung von Lehrveranstaltungen durch Dipl.-Ing. Udo Reitz hat zu vielen Ideen geführt, die sich auch in einigen Beispielprogrammen im Buch niederschlagen. Weiterhin möchte ich mich bedanken bei Dipl.-Ing. Oliver Drölle, Dipl.-Ing. Matthias Faulstich und Dipl.-Ing. Georg Overmann vom Labor für Angewandte Informatik (FH SWF) für die stete Hilfsbereitschaft bei Fragen aller Art.

Für die Prüfung des Manuskriptes auf inhaltliche Korrektheit bin ich Thomas Sandrisser (National Instruments) und Prof. Dr.-Ing. Erhard Stein (FH Lausitz, Senftenberg) und für die mühsame Rechtschreibkorrektur des Manuskriptes Dipl.-Bibl. Sabine Schust (Staatsbibliothek zu Berlin) zu großem Dank verpflichtet. Besonders zu erwähnen ist an dieser Stelle Dr.-Ing. Friedrich Müller (Max-Planck-Institut für Astronomie, Heidelberg), der das gesamte Manuskript gleichfalls mit großer Sorgfalt auf formale und inhaltliche Fehler überprüft und darüber hinaus die Strapaze auf sich genommen hat, einen großen Teil der Korrekturen in das Manuskript einzuarbeiten.

Ebenfalls danke ich Frau Dipl.-Biol. Barbara Lühker und Dr. rer. nat. Andreas Rüdinger von Elsevier – Spektrum Akademischer Verlag für die außerordentliche Unterstützung und die gute Ausstattung des Buches sowie für die Geduld bei der Manuskripterstellung.

Letztendlich war der Motor für die Erstellung des Buches die Zusammenarbeit mit engagierten Studierenden, die mir nach wie vor viel Freude bereitet. Stellvertretend für viele möchte ich an dieser Stelle Dr.-Ing. Martin Skambraks (Bundesamt für Wehrtechnik und Beschaffung, Koblenz), Dipl.-Ing. Stefan Tauche (Infineon, München) und Christoph Keßler (FH SWF) erwähnen, die auch nach ihrem Studium bzw. außerhalb von Lehrveranstaltungen jederzeit bereit waren, mir mit Rat und Tat zur Seite zu stehen.

Software-Version

Die dem Buch beiliegende DVD enthält die „LabVIEW Studentenversion“ in deutscher und englischer Sprache für Windows-PCs, die sich vom *LabVIEW Full Development System* lediglich durch die Einblendung eines Wasserzeichens unterschei-

det; der Funktionsumfang ist vollständig identisch. Da PCs mit einem Windows-Betriebssystem zurzeit in der Praxis deutlich überwiegen, orientiert sich der Text im Wesentlichen an der Windows-Version von LabVIEW. Soweit erforderlich, werden aber auch die entsprechenden Hinweise für Mac- und Linux-Versionen gegeben.

Verwendet wird hier die Version 8.0 von LabVIEW, mit der auch alle Beispiele im Buch realisiert worden sind. Dabei wurde darauf geachtet, dass die Beispiele auch für Nutzer der Versionen 6 und 7 weitestgehend lesbar bleiben. Ausgehend von der Version 7.1 werden neue Funktionen in der Marginalienspalte durch „LV8.0“ gekennzeichnet.

Mittlerweile ist LabVIEW lokalisiert worden, d.h. die Entwicklungsumgebung steht auch in deutscher, französischer, japanischer, koreanischer und chinesischer Sprache zur Verfügung. Der vorliegende Text – obwohl in deutscher Sprache – bezieht sich auf die englische Version von LabVIEW. Auf den ersten Blick mag dieser Ansatz ungewöhnlich und unter Umständen auch ein wenig abschreckend wirken und soll daher im Folgenden begründet werden.

- Jede Übersetzung – so gut sie auch sein mag – muss an einer präzisen Übertragung scheitern, da nicht für alle Wörter einer Sprache exakte Entsprechungen in einer anderen Sprache zur Verfügung stehen, so dass sich allein durch die Übersetzung Verständnisprobleme ergeben können. Beispielhaft soll dies der erste Satz aus „La Divina Comedia“ (Die Göttliche Komödie) von Dante Alighieri veranschaulichen:

*Auf halbem Weg des Menschenlebens fand
ich mich in einen finstern Wald verschlagen.
Weil ich vom rechten Weg mich abgewandt.
Übersetzung: Karl Streckfuß*

*Dem Höhepunkt des Lebens war ich nahe,
da mich ein dunkler Wald umfing und ich,
verirrt, den rechten Weg nicht wieder fand.
Übersetzung: Karl Vossler*

„Auf halbem Weg“ und „Auf dem Höhepunkt“ sind sicher nicht bedeutungsgleich. In der Literatur wird eine Fehlinterpretation allenfalls den ästhetischen Genuss mindern, während in der Technik Fehlinterpretationen zu Fehlfunktionen und den damit verbundenen wirtschaftlichen Folgen führen können.

- In aller Regel stehen Updates für lokalisierte Software-Versionen erst einige Monate später zur Verfügung. Daher können Fehler durch die Verwendung der Originalversion frühzeitig vermieden werden.
- Die effektive Nutzung von LabVIEW basiert auch auf dem umfangreichen Hilfesystem, in dem unter anderem anhand einer Vielzahl von Beispielen das Verständnis für nahezu alle Funktionen gefördert wird. Diese Programmbeispiele stehen ausschließlich in englischer Sprache zur Verfügung, so dass eine Einführung in LabVIEW in deutscher Sprache letztendlich erfordern würde, sich das Verständnis für die Entwicklungsumgebung sowohl in deutscher als auch in englischer Sprache zu erarbeiten.
- In gleicher Weise gilt dies für Geräte-Treiber, *Application Notes* und besonders für Beispielprogramme, die auf der Homepage von National Instruments (www.ni.com/downloads) in großem Umfang zur Verfügung stehen, da sich LabVIEW-Entwickler in hohem Maße als Teil einer Gemeinschaft (*Community*) verstehen, bei der der Austausch von Informationen und Lösungen zum Nutzen aller zu einem selbstverständlichen Teil der täglichen Arbeit gehört.
- Die umfangreichen Erweiterungen der Entwicklungsumgebung LabVIEW durch Module und *Toolkits* stehen ausschließlich in englischer Sprache zur Verfügung;

für diese ist keine Lokalisierung vorgesehen. Eine Erweiterung der Entwicklungsumgebung hat dann automatisch zur Folge, dass die Grundfunktionen in deutscher und die neuen Funktionen in englischer Sprache verwendet werden müssen.

Durch den Einsatz der englischen Version können all diese Nachteile vermieden werden. Im Text wird aber zu allen englischen Fachbegriffen in unmittelbarem Zusammenhang auch die deutsche Übersetzung angegeben. Einfachere Begriffe wie *Help* = Hilfe oder *File* = Datei werden dabei jedoch als bekannt vorausgesetzt. In Zweifelsfällen orientiert sich die Übersetzung an den Begriffen der deutschen LabVIEW-Version und nicht an einem Fachwörterbuch, damit der Text auch bei einer Installation der deutschen LabVIEW-Version lesbar und nutzbar bleibt. Im Stichwortverzeichnis werden alle Begriffe sowohl in englischer als auch in deutscher Sprache aufgeführt.

Schreibung

Einige wenige Abweichungen der Standard-Schriftart und Schreibung werden zur Hervorhebung von Fachbegriffen, Programmnamen etc. verwendet.

- Englische Fachbegriffe sind grundsätzlich *kursiv* gesetzt.
- Programmnamen sind in **Schreibmaschinenschrift (Courier)** gesetzt.
- Hinweise und ergänzende Tipps werden durch ► ... ◄ geklammert.
- Die Verzweigung in Unterpaleetten oder Untermenüs wird durch einen Doppelpfeil » gekennzeichnet.
- Beschreibungen zur Bedienung einer Computer-Maus sind durch den Wunsch nach einer kurzen und prägnanten Schreibung geprägt, die zu sprachlichen Ungetümen wie dem „rechten Mausklick“ führt, womit das Betätigen der rechten Taste einer Computer-Maus gemeint ist. Dafür wird hier – in Anlehnung an das Betriebssystem MAC OS – der Begriff Kommandoklick verwendet.

Dezimaltrennzeichen

Eine wichtige Anmerkung zur Schreibung betrifft das Dezimaltrennzeichen. Dem englischen Sprachgebrauch folgend, wird hier durchgehend ein Punkt und nicht ein Komma als Dezimaltrennzeichen benutzt. Spätestens bei der Kommunikation zwischen einzelnen Geräten können so von vornherein schwer aufzufindende Fehler in Programmen vermieden werden, da nahezu alle Kommunikationsprotokolle mit dem Punkt als Dezimaltrennzeichen arbeiten.

Gliederung des Buches

Nach der Einleitung werden in einem ersten Teil zunächst in einer kurzen Zusammenfassung wesentliche Grundlagen der Informatik behandelt, in Kapitel 2 die Darstellung von Zahlen und weiterer Datentypen in einem Rechner, in Kapitel 3 die boolesche Algebra und Schaltnetze. Kapitel 4 gibt einen Überblick über einige Konzepte der Software-Technik, soweit diese im Zusammenhang mit der Entwicklungsumgebung LabVIEW stehen, insbesondere Datenflussdiagramme, Kontrollstrukturen und endlichen Automaten.

Bei vorhandenen Vorkenntnissen oder dem in der Praxis häufig zu beobachteten Zeitdruck bei der Projektarbeit ist es ohne Einschränkungen möglich, direkt mit den Kapiteln 5 und 6 zu beginnen, in denen die Entwicklungsumgebung LabVIEW eingeführt und die Programmentwicklung an einem ersten, einfachen Beispiel erläutert wird. Fehlende Informationen können gegebenenfalls jederzeit in den Kapiteln 2 bis 4 nachgelesen werden.

Danach werden in Kapitel 7 Kontrollstrukturen, in Kapitel 8 Datentypen und in Kapitel 9 Datenstrukturen im Zusammenhang mit der strukturierten Datenflussprogrammierung in LabVIEW behandelt. Anschließend werden in Kapitel 10 Funktionen zur Dateieingabe und -ausgabe eingeführt. Unabhängig von der algorithmischen Sicht bei der Software-Entwicklung werden dann in Kapitel 11 die umfangreichen Möglichkeiten zur Gestaltung der Bedienoberfläche vorgestellt.

Mit dem Ziel, ein tragfähiges Konzept für komplexere Aufgabenstellungen zur Verfügung zu stellen, wird abschließend in Kapitel 12 anhand von drei Beispielen ausführlich gezeigt, wie endliche Automaten in in LabVIEW realisiert werden können.

Bernward Mütterlein
Iserlohn, im Dezember 2006

Handbuch für die Programmierung mit LabVIEW
mit Studentenversion LabVIEW 2009

Mütterlein, B.

2009, XVIII, 517 S., Softcover

ISBN: 978-3-8274-2337-5