

## Geleitwort

Aktuelle Entwicklungen führen es uns deutlich vor Augen: Mit rasantem Tempo stellen neue Technologien unsere Welt geradezu auf den Kopf. Ob künstliche Intelligenz, Big Analog Data, Cloudtechnologien, 5G oder die wachsende Vernetzung elektronischer Geräte – Technik verändert nicht nur unsere wirtschaftlichen Strukturen, sondern beeinflusst auch unsere Arbeit, unsere Art zu leben und zu konsumieren. Doch die Technologien allein sind noch nicht einmal das Ausschlaggebende. Vielmehr ist es deren Kombination, die uns ungeahnte neue Möglichkeiten beschert und in vielen Industriezweigen für völlig neuartige Produkte, Dienstleistungen und Geschäftsmodelle sorgen wird. Nicht zuletzt befeuert dies disruptive Trends wie das Industrielle Internet der Dinge (IIoT), das wiederum den Motor für die digitale Transformation darstellt.

An der digitalen Transformation kommt mittlerweile wohl kaum jemand mehr vorbei. Ihr Sinn und Zweck ist es, Markteinführungszeiten zu verringern, neue, differenzierte Produkte auf den Markt zu bringen, die betriebliche Effizienz zu steigern und neue Geschäftsmodelle zu erschaffen. Unternehmen erhoffen sich durch sie in vielfacher Hinsicht einen Vorteil. Bei der Umsetzung allerdings scheiden sich die Geister. Nicht anders ist es im Bereich des Messens und Testens bzw. der Mess- und Testdaten. Dabei sind diese Daten und auch die softwarebasierte Automatisierung ein Katalysator für die digitale Transformation.

Denn Test- und Messdaten sind eine reiche Informationsquelle und liefern z. B. Daten zur Leistung des Produkts, zur Leistung des Anlagenteils, zu Qualität und sogar zur Leistung im Betrieb. Durch den «Digital Thread», also die Zusammenführung unterschiedlicher IT-Systeme, mit dem Ziel, daraus neue Erkenntnisse zur Optimierung des Produktionsprozesses zu gewinnen, können wir nun wesentlich mehr Informationen über ein Produkt erfassen und verstehen als je zuvor – und das über den gesamten Produktlebenszyklus hinweg.

Dabei sind die Daten nicht nur auf den Entwicklungszyklus innerhalb einer bestimmten Firma limitiert, sondern können sogar die gesamte Lieferkette einschließen. Man denke an Unternehmen, die ihre Produktion auslagern und davon profitieren könnten, würden die Daten über die Firmengrenzen hinweg ausgetauscht, um nur ein Beispiel zu nennen. Doch in vielen Unternehmen wird diese Goldgrube an aufschlussreichen Daten schlichtweg nicht genutzt.

Um das Potential der verfügbaren Informationen vollends ausschöpfen zu können, bedarf es allerdings eines fundamentalen Umdenkens von einem hardwareorientierten zu einem softwarezentrischen Plattformansatz wie ihn National Instruments bereits seit Jahrzehnten verfolgt. Nur so lassen sich Systeme schnell und kostengünstig an sich rapide ändernde Anforderungen anpassen und neue kommerzielle Technologien integrieren. Mit einer Plattform, die über ein robustes Ökosystem verfügt und zu jeder Zeit die passenden Werkzeuge bereitstellt, sind Anwender sowohl für heutige Herausforderungen als auch für die Anforderungen zukünftiger Technologien bestens gerüstet.

Eine Plattform, die all dies ermöglicht, muss jedoch nicht nur einen Einblick in die Daten gewähren, sondern auch einen logischen Programmablauf aufweisen. Mit dem ihr zugrundeliegenden Konzept des Datenflusses schafft die Systemdesignsoftware LabVIEW – das Herzstück von NIs Plattform – den Spagat zwischen der

Sichtbarkeit von Daten und der logischen Programmausführung. Mehr noch: Das Konzept des Datenflusses entspricht sogar in idealer Weise den menschlichen Fähigkeiten hinsichtlich der kognitiven Erfassung und Informationsverarbeitung. Zahlreiche wissenschaftliche Erkenntnisse belegen, dass der Mensch sich visuelle Zusammenhänge einfacher erschließen kann. Zudem fällt es dem Gehirn wesentlich leichter, bildhafte Darstellungen und Eindrücke zu erfassen als abstrakte hardwareorientierte Kodierung. Kinder reagieren lange bevor sie lesen und schreiben lernen auf visuelle Eindrücke und können bereits räumliche Informationen verarbeiten. Auch bei naturwissenschaftlichen Versuchsaufbauten werden oft visuelle Anschauungsmittel eingesetzt, da das Experiment so besser begreifbar wird. Nicht zuletzt deshalb orientierte sich National Instruments bei der Entwicklung der Programmierumgebung LabVIEW nicht etwa an der zugrunde liegenden Hardware, sondern vielmehr an den kognitiven Fähigkeiten des Menschen.

Dem Autor des vorliegenden Buchs, Herrn Kurt Reim, gilt daher mein ganz besonderer Dank für seinen kontinuierlichen Einsatz, LabVIEW im Hochschulbereich zu fördern. Auch für die dritte Auflage dieses Werks wurden die Funktionen der neueren LabVIEW-Version genauer unter die Lupe genommen, mit dem Ziel, insbesondere Lernenden an Technikerschulen den Einstieg in die Struktur dieser Programmiersprache nahezubringen. Wie gewohnt, wartet auch die hier vorgestellte Version 2018 mit neuen, mächtigen Funktionen auf.

Anhand der wie immer zahlreichen Übungsbeispiele und dazugehörigen Musterlösungen kann der Leser das Wissen rasch in die Praxis umsetzen. Somit gibt K. Reim dem Lernenden ebenso wie dem erfahreneren LabVIEW-Anwender ein Nachschlagewerk an die Hand, das Theorie und Praxis miteinander verquickt und den Anwender bei der alltäglichen Nutzung von LabVIEW unterstützen soll.

München, im Januar 2020

Rahman Jamal

Business & Technology Fellow, National Instruments  
Visiting Professor, Chinesisch-Deutsche Hochschule für Angewandte  
Wissenschaften der Tongji-Universität Shanghai (China)

# Vorwort

## *Was ist LabVIEW?*

LabVIEW ist eine Entwicklungsumgebung, mit deren Hilfe Programme für mess-, steuer- und regelungstechnische Anwendungen erstellt werden können. Verwendet wird dazu die grafische Programmiersprache G, die zur Abbildung technischer Prozesse entwickelt wurde. Sie unterscheidet sich von einer textorientierten Sprache dadurch, dass Funktionsblöcke dargestellt werden, die – entsprechend miteinander verbunden – den Datenfluss eines Programms abbilden.

Ein wesentliches Merkmal von LabVIEW ist die Möglichkeit, physikalische Signale in das Programm einzulesen, zu verarbeiten und visuell darzustellen. Dadurch kann ein LabVIEW-Programm nicht nur ein Messgerät ersetzen, sondern es bietet dem Benutzer die Möglichkeit, sich sein eigenes, auf die Messaufgabe zugeschnittenes Messgerät zu entwerfen. Ein LabVIEW-Programm bezeichnet man deshalb als VI (Virtuelles Instrument).

Die Abbildungen dieses Buches wurden an die LabVIEW-Version 2018 angepasst.

## *Hinweise zum Buch*

Je länger man mit LabVIEW arbeitet, desto mächtiger und umfangreicher erscheint es. Paletten und Menüs liefern eine Vielzahl von Bedien- und Anzeigeelementen und eine nicht enden wollende Anzahl von Bearbeitungsfunktionen. Einsteiger sind davon oft irritiert, wenn nicht sogar ratlos.

Dieser vorliegende Kurs soll allen Einsteigern kurz und knapp die ersten Schritte in die LabVIEW-Welt erleichtern. Er basiert auf einem Skript zu der Lehrveranstaltung «Computergestützte Messtechnik» an der Rudolf-Diesel-Fachschule in Nürnberg. Ziel war es, den Schülern einen Überblick über die Struktur von LabVIEW zu verschaffen und sie dann Schritt für Schritt mit den wesentlichen Bausteinen der Programmiersprache bekannt zu machen. Jedes Kapitel liefert zuerst eine kurze theoretische Zusammenfassung der wichtigsten Funktionen und Eigenschaften eines Themas, danach bieten darauf abgestimmte Übungsaufgaben die Möglichkeit, das Erlernte zu sichern und zu festigen. Nach dem Durcharbeiten des Buches besitzt der Schüler eine solide Grundlage, um sich selbstständig in weitere Themenkreise von LabVIEW einzuarbeiten.

Bestimmt kann das Buch auch fortgeschrittenen Anwendern hilfreich sein, da es im Theorieteil wie eine «Formelsammlung» viele Eigenschaften und Funktionen von LabVIEW zusammenfasst, ordnet und übersichtlich präsentiert.

Objekte von LabVIEW enthalten sehr viele Funktionalitäten, so dass es unmöglich ist, diese alle zu beschreiben. Das ist in einem Buch auch nicht notwendig, denn die beste Beschreibung liefert LabVIEW selbst in seiner **Hilfe-Funktion**. Da dies die offizielle Referenz von LabVIEW ist, wird bei der Erklärung von Aufgaben auch häufig darauf verwiesen.

Obwohl die Beschreibungen in der Hilfe-Funktion oft sehr allgemein und dadurch etwas abstrakt formuliert sind, sollte man sie unbedingt häufig nutzen, da zu vielen Themen auch Verweise auf Beispielprogramme enthalten sind.

In diesem Buch beschränken sich die Erklärungen daher nur auf die für Bearbeitung und Lösung der Aufgaben notwendigen Kenntnisse, d.h., aus Menüs werden nur konkret benötigte Punkte und der Umgang damit besprochen. Auch die dargestell-

ten Bildschirmkopien (Screenshots) zeigen nicht immer die vollständigen Menüs an, sondern beschränken sich aus Platzgründen auf die im Text besprochenen Punkte. Die Übungsaufgaben sind so ausgewählt, dass damit ein möglichst breites Spektrum an unterschiedlichen Themen, Methoden und Eigenschaften von LabVIEW abgedeckt wird.

Einige der vorgestellten Beispiele sind anwendungstechnisch gesehen vielleicht etwas zu einfach, dadurch sind aber die zugrunde liegenden Funktionen besser zu verstehen.

Ein wichtiger Punkt dieses Buches ist nicht nur die Bereitstellung einer Anzahl von Übungsaufgaben, sondern auch die für einen Lernerfolg unbedingt notwendigen Lösungen dazu. Wie aber jeder Programmierer weiß, bieten sich bei einem Programm-entwurf immer mehrere Lösungen an. Die vorgestellte Lösung ist also nicht «die richtige», sondern eine mögliche. Weiterhin findet jeder geübte Nutzer von LabVIEW in vielen vorgestellten Programmen Möglichkeiten zur Verbesserung oder Erweiterung. Aus Platzgründen wurde hier darauf verzichtet. Ebenso wird die sehr umfangreiche Fehlerbehandlung von LabVIEW nur in einem Kapitel prinzipiell angedeutet. Abschließend möchte ich mich noch bei allen Schülern bedanken, die durch ihre rege Mitarbeit am Zustandekommen des Buches beteiligt waren. Der Vogel Communications Group danke ich für die gute Zusammenarbeit. Anregungen und Verbesserungsvorschläge aus dem Leser- und Benutzerkreis nimmt der Verlag gerne entgegen.

Alle Aufgaben des Buches stehen dem Leser zur Ausführung als VI zur Verfügung. Die Programmfunktion kann getestet werden, das Blockdiagramm ist nicht gesperrt und kann verändert werden. National Instruments bietet privaten Anwendern ab 2020 eine kostenfreie LabVIEW Community Edition für nichtkommerzielle und nichtakademische Zwecke. Mit dem Programm können die VIs zum Buch direkt ausgeführt werden. Der Downloadlink wird unmittelbar nach Veröffentlichung der Community Edition über den Onlineservice InfoClick bereitgestellt.

Anfragen können gerne an folgende E-Mail-Adresse gerichtet werden:  
[labview@vogel-fachbuch.de](mailto:labview@vogel-fachbuch.de)

# Inhaltsverzeichnis

Vorwort .....	7
<b>1 Entwicklungsumgebung .....</b>	<b>15</b>
1.1 Entwurf eines Virtuellen Instruments (VIs) .....	15
1.1.1 Startfenster .....	15
1.1.2 Frontpanel und Blockdiagramm .....	15
1.2 Arbeiten in den Fenstern .....	17
1.2.1 Bearbeitung Frontpanel .....	18
1.2.2 Bearbeitung Blockdiagramm .....	19
1.3 Ausführen eines VI .....	21
1.4 Werkzeugpalette .....	23
1.5 Kontextmenü .....	24
1.5.1 Kontextmenü von Bedien- und Anzeigeelementen .....	24
1.5.2 Kontextmenü von Funktionsobjekten im Blockdiagramm .....	25
1.5.3 Kontextmenü einer Verbindungsleitung .....	26
1.6 Hilfsmittel .....	26
1.6.1 Beispiele suchen .....	26
1.6.2 LabVIEW-Hilfe .....	27
1.6.3 Kontexthilfe .....	27
1.6.4 Hilfsmittel im Bearbeitungsmodus .....	28
1.6.5 Hilfsmittel im Ausführungsmodus .....	29
1.7 Umgang mit Objekten .....	30
1.7.1 Markieren von Objekten .....	30
1.7.2 Objektgröße ändern .....	31
1.8 Express-VI .....	32
1.9 Diverses .....	33
1.9.1 Passwortschutz .....	33
1.9.2 EXE-Programm .....	33
1.9.3 Blockdiagramm neu zeichnen .....	33
1.9.4 VI speichern .....	33
1.9.5 VI ausdrucken .....	34
1.9.6 Tastenkombinationen (Shortcuts) .....	34
<b>2 Numerische Funktionen .....</b>	<b>35</b>
2.1 Bedienelemente zur Eingabe numerischer Daten (Datenquellen) .....	35
2.2 Anzeigeelemente zur Ausgabe numerischer Daten (Datensenken) .....	35
2.3 Formelknoten .....	39
2.4 Ausdrucksknoten .....	40
<i>Aufgaben</i> .....	42
<b>3 Boolesche Funktionen .....</b>	<b>51</b>
3.1 Vergleichen numerischer Werte .....	52
3.2 Logische Verknüpfungen .....	53
<i>Aufgaben</i> .....	57

<b>4</b>	<b>Schleifen</b> .....	63
	4.1 While-Schleife .....	64
	4.2 For-Schleife .....	65
	4.3 Ablaufreihenfolge .....	70
	<i>Aufgaben</i> .....	72
<b>5</b>	<b>Graphische Anzeigen</b> .....	81
	5.1 Signalverlaufdiagramm .....	82
	5.2 Signalverlaufgraph .....	84
	5.3 XY-Graph .....	86
	<i>Aufgaben</i> .....	89
<b>6</b>	<b>Case</b> .....	95
	6.1 Boolesche Case-Struktur .....	95
	6.2 Numerische Case-Struktur .....	96
	6.3 Ringelement .....	97
	6.4 Enum .....	98
	<i>Aufgaben</i> .....	100
<b>7</b>	<b>Sequenz und lokale Variable</b> .....	107
	7.1 Sequenz .....	107
	7.2 Gestapelte Sequenz .....	109
	7.3 Lokale Variable .....	110
	<i>Aufgaben</i> .....	112
<b>8</b>	<b>SubVI</b> .....	115
	8.1 Erstellen eines SubVI .....	115
	8.2 Benutzung des SubVI .....	116
	8.3 Automatische Erstellung eines SubVI .....	117
	<i>Aufgaben</i> .....	119
<b>9</b>	<b>Strings</b> .....	123
	<i>Aufgaben</i> .....	128
<b>10</b>	<b>Cluster</b> .....	133
	10.1 Erstellen eines Clusters .....	133
	10.2 Reihenfolge der Cluster-Elemente .....	134
	10.3 Fehler-Cluster .....	136
	<i>Aufgaben</i> .....	138
<b>11</b>	<b>Array</b> .....	143
	11.1 Erstellen eines Arrays .....	143
	11.2 Lottozahlengenerator .....	144
	11.3 Autoindizierung .....	146
	<i>Aufgaben</i> .....	147

<b>12</b>	<b>Datei-I/O</b> .....	153
	12.1 Speichern von Daten .....	153
	12.2 Pfadangabe .....	157
	12.3 Lesen von Dateien .....	159
	<i>Aufgaben</i> .....	161
<b>13</b>	<b>Messen mit LabVIEW</b> .....	167
	13.1 Der Befehlssatz SCPI .....	168
	13.2 Measurement & Automation Explorer .....	170
<b>14</b>	<b>Serielle Schnittstelle</b> .....	173
	14.1 Anschluss eines Messgeräts .....	173
	14.2 Datenübertragung .....	173
	14.3 Ablaufsteuerung (Flow Control) .....	174
	14.4 Betriebsarten .....	175
	14.5 RS-232 mit LabVIEW .....	176
	<i>Aufgaben</i> .....	178
<b>15</b>	<b>GPIB-Schnittstelle</b> .....	183
	15.1 Aufbau eines GPIB-Messsystems .....	183
	15.2 Aufbau des IEC-Busses .....	184
	15.3 Allgemeine Daten eines IEC-Bus-Systems .....	185
	15.4 GPIB mit LabVIEW .....	186
	<i>Aufgaben</i> .....	187
<b>16</b>	<b>NI myDAQ</b> .....	193
	16.1 Anschlussmöglichkeiten .....	193
	16.2 Spannungsmessung mit Eingang AI0 und DAQ-Assistent .....	194
	16.3 Datentyp «Dynamisch» .....	196
	16.4 Spannungsmessung mit AI0 und DAQmx .....	196
	16.5 Datentyp Signalverlauf .....	199
	16.6 Spannungsausgabe .....	200
	16.7 Digitale Ein- und Ausgabe mit DAQ-Assistent .....	200
	16.8 Digitale Ein- und Ausgabe mit DAQmx .....	201
	<i>Aufgaben</i> .....	205
	<b>Lösungen</b> .....	213
	<b>ASCII-Code</b> .....	276
	<b>Steuerzeichen</b> .....	277
	<b>Literaturverzeichnis</b> .....	278
	<b>Stichwortverzeichnis</b> .....	279

# Aufgaben

## Kapitel 2 Numerische Funktionen

Aufgabe 2.01	Temperaturumrechnung . . . . .	42
Aufgabe 2.02	Gesamtwiderstand . . . . .	43
Aufgabe 2.03	Kugel / Kegel . . . . .	44
Aufgabe 2.04	Höhenmessung / Sonnenstand . . . . .	45
Aufgabe 2.05	Zylinderdichte . . . . .	46
Aufgabe 2.06	Leitungslänge . . . . .	47
Aufgabe 2.07	Geschwindigkeitsmessung . . . . .	48
Aufgabe 2.08	Zeit . . . . .	49
Aufgabe 2.09	Farbsynthese . . . . .	50

## Kapitel 3 Boolesche Funktionen

Aufgabe 3.01	Soll-Ist-Vergleich . . . . .	57
Aufgabe 3.02	Safe Schloss . . . . .	58
Aufgabe 3.03	Bargraph . . . . .	58
Aufgabe 3.04	Füllstand . . . . .	60
Aufgabe 3.05	DIP-Schalter . . . . .	61

## Kapitel 4 Schleifen

Aufgabe 4.01	Würfel1 . . . . .	72
Aufgabe 4.02	Würfel2 . . . . .	73
Aufgabe 4.03	Sägezahn . . . . .	74
Aufgabe 4.04	Netzteil . . . . .	75
Aufgabe 4.05	Blinklicht . . . . .	76
Aufgabe 4.06	Schleifenfunktionen1 . . . . .	77
Aufgabe 4.07	Schleifenfunktionen2 . . . . .	78
Aufgabe 4.08	Schleifenfunktionen3 . . . . .	79
Aufgabe 4.09	Schleifenfunktionen4 . . . . .	80

## Kapitel 5 Graphische Anzeigen

Aufgabe 5.01	Frequenzänderung / Phasenverschiebung . . . . .	89
Aufgabe 5.02	Augenblicksleistung . . . . .	90
Aufgabe 5.03	Blindwiderstand . . . . .	91
Aufgabe 5.04	Lissajous-Figur . . . . .	92
Aufgabe 5.05	Flimmer-Box . . . . .	93

## Kapitel 6 Case

Aufgabe 6.01	Taschenrechner . . . . .	100
Aufgabe 6.02	Obstpreise . . . . .	102
Aufgabe 6.03	Winkelfunktionen . . . . .	102
Aufgabe 6.04	Würfelanzeige . . . . .	103
Aufgabe 6.05	Stern-Dreieck-Umrechnung . . . . .	104
Aufgabe 6.06	Durchflussmessung . . . . .	105



**Kapitel 7 Sequenz und lokale Variable**

Aufgabe 7.01	Wartezeitmessung / Eieruhr	112
Aufgabe 7.02	Summe $1/x$	113
Aufgabe 7.03	Würfeltreffer	113

**Kapitel 8 SubVI**

Aufgabe 8.01	Rechenoperationen	119
Aufgabe 8.02	SubVI Tiefpass / Hochpass	120
Aufgabe 8.03	Übertragungsfunktion	121
Aufgabe 8.04	Rechteckschwingung	122

**Kapitel 9 Strings**

Aufgabe 9.01	Stringumwandlungen	128
Aufgabe 9.02	Stringfunktionen1	129
Aufgabe 9.03	Stringfunktionen2	130
Aufgabe 9.04	ASCII-Generator	131
Aufgabe 9.05	Datum/Zeitangabe als String	132

**Kapitel 10 Cluster**

Aufgabe 10.01	Buchkarteikarte	138
Aufgabe 10.02	Prüfprotokoll	139
Aufgabe 10.03	Zeitstempel	140
Aufgabe 10.04	Messstellen	141
Aufgabe 10.05	Fehler-Cluster	142

**Kapitel 11 Array**

Aufgabe 11.01	Arrayfunktionen	147
Aufgabe 11.02	Zehn Lottospiele	148
Aufgabe 11.03	Jahrestabelle	149
Aufgabe 11.04	Trefferwahrscheinlichkeit	150
Aufgabe 11.05	Messwertsortierer	151
Aufgabe 11.06	Phrasengenerator	152

**Kapitel 12 Datei-I/O**

Aufgabe 12.01	Bandpassdaten speichern	161
Aufgabe 12.02	Bandpassdaten lesen	162
Aufgabe 12.03	Lottozahlen-Datei	163
Aufgabe 12.04	Niederschlagsmessung	164

**Kapitel 14 Serielle Schnittstelle**

Aufgabe 14.01	Nur-Senden-Modus	178
Aufgabe 14.02	Senden und Empfangen	179
Aufgabe 14.03	Messung konfigurieren	180

**Kapitel 15 GPIB-Schnittstelle**

Aufgabe 15.01 Messgerät bedienen . . . . .	187
Aufgabe 15.02 Frequenzgenerator wobbeln . . . . .	189
Aufgabe 15.03 Tiefpass-Messung . . . . .	190
Aufgabe 15.04 Diodenkennlinie . . . . .	191

**Kapitel 16 NI myDAQ**

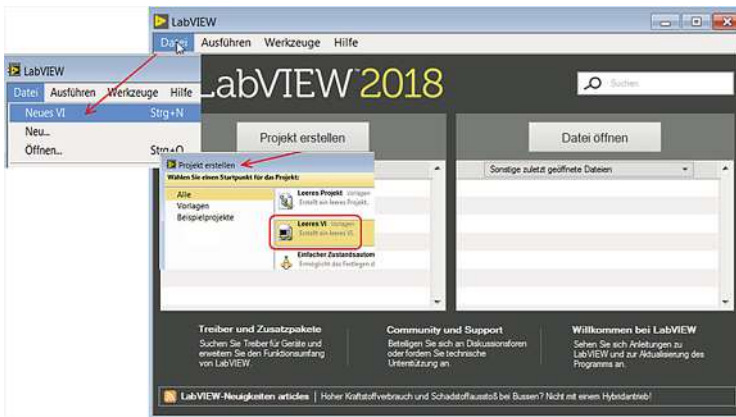
Aufgabe 16.01 Pt100-Messung . . . . .	205
Aufgabe 16.02 Sägezahnspannung . . . . .	206
Aufgabe 16.03 Kennlinie Solarmodul . . . . .	207
Aufgabe 16.04 Kennlinien darstellen . . . . .	208
Aufgabe 16.05 Klangkontrolle . . . . .	209
Aufgabe 16.06 Signalverlauf . . . . .	210

# 1 Entwicklungsumgebung

## 1.1 Entwurf eines Virtuellen Instruments (VIs)

### 1.1.1 Startfenster

Nach dem Aufruf von LabVIEW erscheint das Eröffnungsfenster.



Der Start für den Entwurf eines neuen LabVIEW-Programms (VI) erfolgt entweder – mit *Datei* → *Neues VI* oder

– unter *Projekt erstellen* und einem Doppelklick auf → *Leeres VI*.

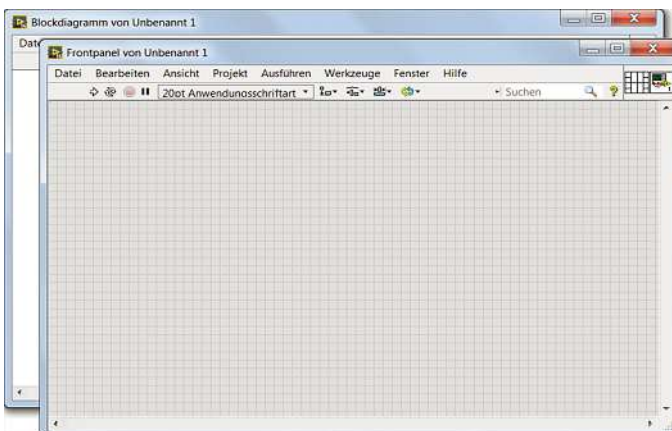
Der Aufruf von vorhandenen VIs erfolgt mit dem Auswahlpunkt *Datei öffnen*.

Startet man mit *Neues/Leeres VI*, öffnen sich zwei hintereinander liegende, leere Fenster.

Man befindet sich im *Bearbeitungsmodus*.

### 1.1.2 Frontpanel und Blockdiagramm

Ein Klick in das jeweilige Fenster bringt es in den Vordergrund.





**Strg+T** öffnet beide Fenster auf dem Bildschirm. Der Menüpunkt *Fenster* erlaubt die Auswahl, ob sie nebeneinander oder untereinander dargestellt werden.

### Hauptmenü

Beide Fenster besitzen am oberen Rand ein Menü mit 8 Menüpunkten.



Das Hauptmenü bietet die Grundfunktionen für den Umgang mit einem VI, wie Speichern, Drucken, Start und Stopp, Zugriff auf Bearbeitungs- und Hilfsfunktionen sowie auf Zusatzprogramme.

Viele Funktionen des Hauptmenüs sind auch mit Tastenkombinationen (Shortcuts, z.B. **Strg+T**) aufrufbar. Nichtausführbare Menüpunkte sind ausgegraut.

Häufig gebrauchte Menüpunkte des Hauptmenüs findet man als Icon in den beiden Symbolleisten der jeweiligen Fenster.

### Symbolleiste Frontpanel



zur Ausführung eines VI

zum Anordnen der Objekte

### Symbolleiste Blockdiagramm



zur Ausführung eines VI

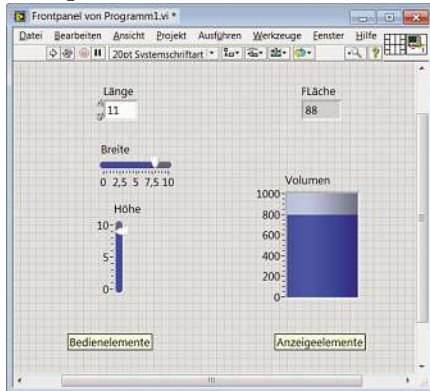
Wahl der Schriftart

zum Anordnen der Objekte

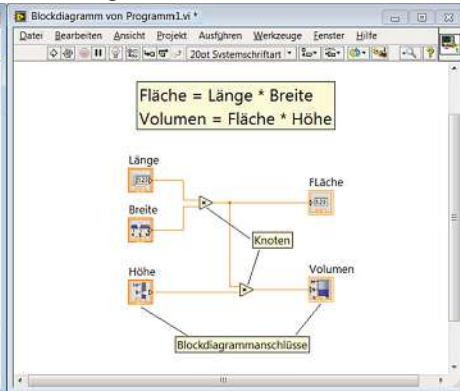
Nachfolgendes VI soll entworfen werden.

**Funktion:** Es besitzt die drei Eingabelemente *Länge*, *Breite* und *Höhe* und die beiden Ausgabelemente *Fläche* und *Volumen*. Aus den Eingabewerten werden die Ausgabewerte berechnet.

## Frontpanel



## Blockdiagramm



### Frontpanel

ist die Bedienoberfläche des Programms. Es dient zur Eingabe von Daten oder Aktionen und stellt Ergebnisse und Ausgaben dar. Es ist die Schnittstelle zum Benutzer. Alle Bedienelemente und alle Anzeigeelemente werden auf der grau karierten Oberfläche platziert.

*Bedienelement = Eingabeelement = Datenquelle  
Anzeigeelement = Ausgabeelement = Datensinke*

### Blockdiagramm

ist das eigentliche Programm. Darin wird der Programmablauf festgelegt, hier werden die Daten bearbeitet. Die auf dem Frontpanel befindlichen Bedien- und Anzeigeelemente findet man hier als Funktionsblöcke mit Anschlüssen. Man nennt sie **Blockdiagrammanschlüsse**.

Eine Verbindung dieser Blockdiagrammanschlüsse durch Leitungen erlaubt den Datentransport zwischen Ein- und Ausgabe. Durch Einfügen von zusätzlichen Funktionsobjekten (man nennt sie allgemein **Knoten**) mit unterschiedlichen Funktionen (Addieren, Multiplizieren) ist eine vielseitige Bearbeitung der Daten möglich.

## 1.2 Arbeiten in den Fenstern

Für das Arbeiten in den Fenstern stehen zwei Menütypen zur Verfügung:

- Im Hauptmenü oder einer Symbolleiste wird mit dem Mauszeiger und Linksklick ein Menüpunkt oder ein Icon ausgewählt.
- Der Mauszeiger zeigt auf ein beliebiges Objekt, und ein Rechtsklick der Maus öffnet **das Kontextmenü**: Es zeigt alles an, was an dieser Stelle mit dem Objekt möglich ist. Die Auswahl wird mit Linksklick bestätigt.

### Achtung:

Das Kontextmenü ändert sich bei einem Objekt – je nachdem, welche Stelle / Eingang / Ausgang man mit der Maus anwählt.



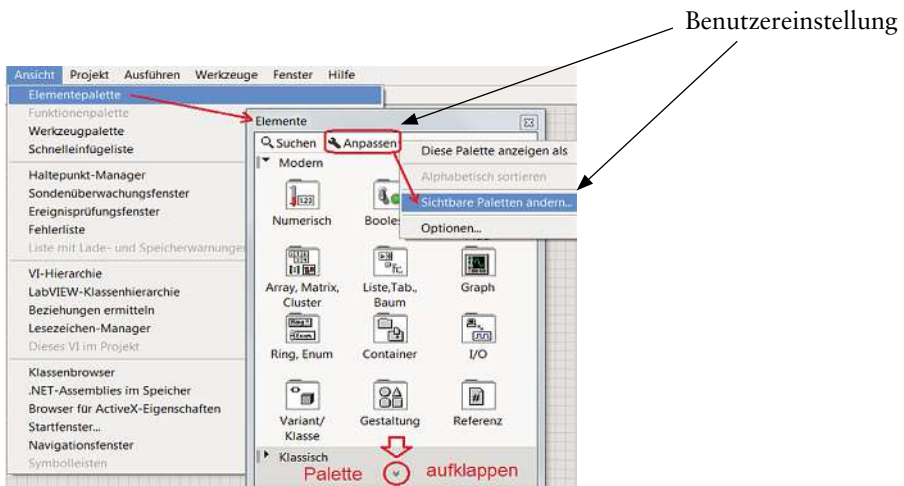
Woher kommen die Eingabe-/Ausgabelemente für das Frontpanel bzw. die Funktionsknoten des Blockdiagramms?

Sie stehen dem Benutzer auf sog. **Paletten** zur Verfügung. Sie sind über das Hauptmenü oder über das Kontextmenü der Arbeitsfläche (Rechtsklick) aufrufbar und enthalten alle Bauteile für die Bearbeitung der beiden Fenster. Da jedes Fenster unterschiedliche Aufgaben hat, besitzt jedes auch unterschiedliche Paletten.

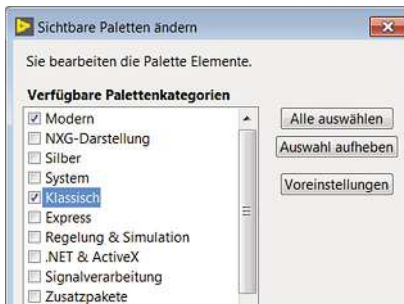
### 1.2.1 Bearbeitung Frontpanel

Auf dem Frontpanel werden Bedienelemente und Anzeigeelemente zur Datenein- und -ausgabe platziert. Dem Frontpanel zugeordnet ist die **Elementepalette**.

Man erhält sie im Frontpanelmenü unter **Ansicht → Elementepalette**.



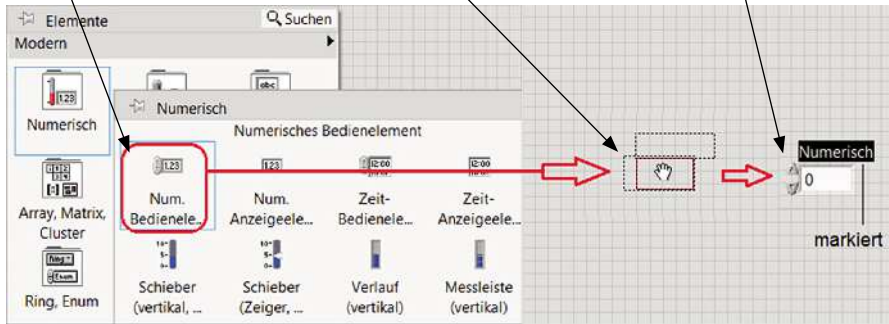
Klappt man die Elementepalette unten auf, sieht man, dass sie etliche Unterpalletten enthält, die zugeklappt (▶) oder aufgeklappt (▼) darstellbar sind.



Da in diesem Buch nur wenige Paletten gebraucht werden, kann man sich zur besseren Übersicht den Inhalt seiner «eigenen» Elementepalette zusammenstellen. Dazu auf **Anpassen → Sichtbare Paletten ändern** klicken, im erscheinenden Auswahlmenü zum Löschen der Voreinstellung auf **Auswahl aufheben** klicken und die beiden Punkte  
 → **Modern** und  
 → **Klassisch** auswählen.

Nach dieser Einstellung liefert ein Rechtsklick auf die leere Frontpaneloberfläche die Elementepalette, d.h., es ist eine Platzierung von Elementen möglich.

Von der Unterpalette *Modern* → *Numerisch* ein numerisches Bedienelement mit Linksklick auswählen, auf die Arbeitsfläche schieben und mit Linksklick ablegen.

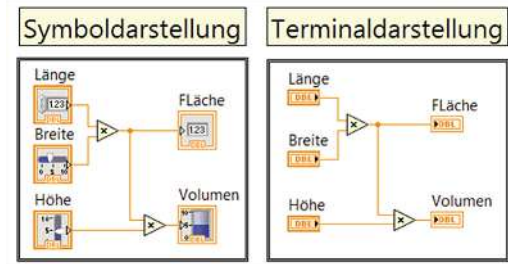


Direkt nach dem Ablegen des Objekts ist die voreingestellte Beschriftung schwarz markiert und kann sofort in die Bezeichnung Länge geändert werden.

**Aufgabe:** Platzieren Sie die noch fehlenden Ein- und Ausgabebestandteile auf das Frontpanel.

## 1.2.2 Bearbeitung Blockdiagramm

Im Blockdiagramm sind die auf dem Frontpanel platzierten Bedien- und Anzeigeelemente als Symbole oder als Terminals zu sehen. Vor dem Platzieren kann die Einstellung im Hauptmenü unter *Werkzeuge* → *Optionen* → *Blockdiagramm* → *Allgemein* → *Frontpanel-Elemente als Symbole darstellen* gewählt werden.



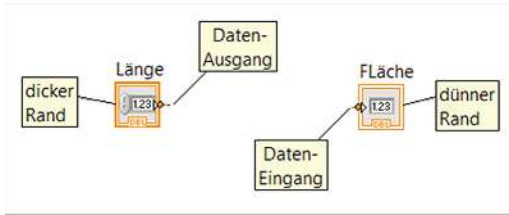
Es gibt keinen Funktionsunterschied zwischen den Darstellungen. Viele Symbole zeigen die Funktion der Elemente an, die Terminals benötigen weniger Platz. Eine Umschaltung der Darstellung ist im Kontextmenü möglich: *Als Symbol anzeigen* ein- oder ausschalten.

Im Blockdiagramm stehen die vom Bediener eingestellten Werte als Daten zur Verarbeitung bereit.

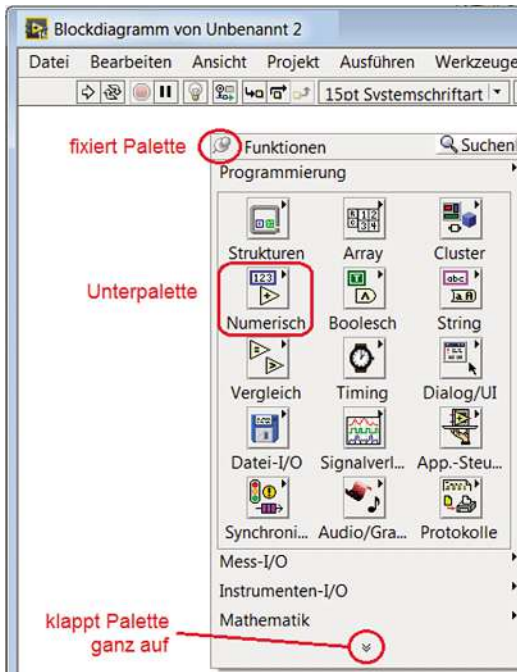
### Grundsatz

- Ein Bedienelement ist eine Datenquelle und besitzt rechts einen Ausgang.
- Ein Anzeigeelement ist eine Datensenke und besitzt links einen Eingang.





Die Bearbeitung der Daten erfolgt mit Objekten, die unterschiedliche **Funktionen** ausführen. Die dem Frontpanel zugeordnete Palette ist die *Funktionenpalette*.



Sie wird im Blockdiagramm mit einem rechten Mausklick aufgerufen.

Mit einer Pinnnadel links oben lässt sich die Funktionenpalette fixieren, mit dem Symbol  $\nabla$  unten ganz aufklappen.

Die Palette *Programmierung* ist aufgeklappt dargestellt und zeigt die am häufigsten gebrauchten Unterpaletten an.

Eine Anpassung an eigene Bedürfnisse wie bei der Elementpalette ist möglich, aber mit den voreingestellten Paletten

→ *Mess-I/O*

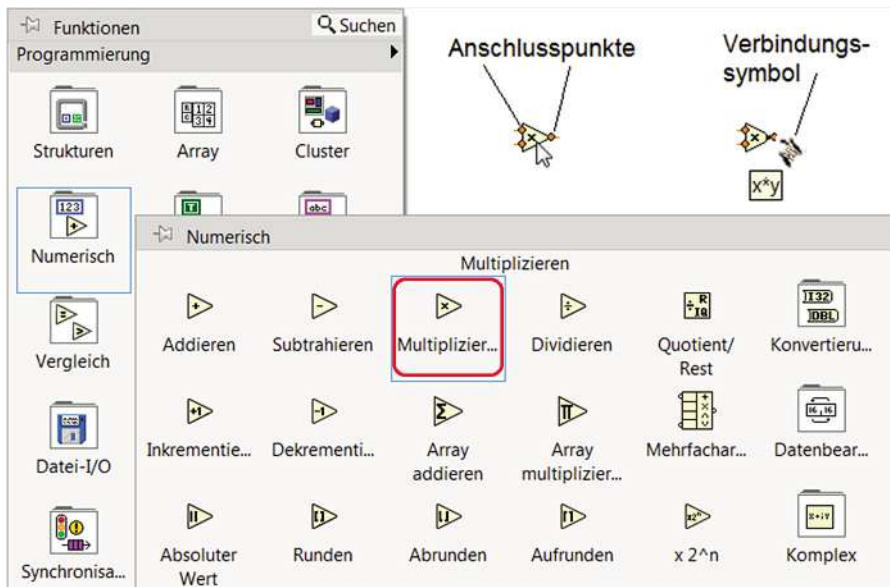
→ *Instrumenten-I/O* und

→ *Mathematik*

nicht nötig.

Von der Unterpalette *Programmierung* → *Numerisch* ein Objekt mit Multiplizierfunktion auswählen, auf die Arbeitsfläche schieben und mit Linksklick ablegen.





Die Maus über dem Objekt zeigt die Anschlusspunkte für Leitungen. Über einen Anschlusspunkt verwandelt sich der Mauszeiger in eine Drahtrolle. Ein Linksklick befestigt daran eine Leitung, die zu einem anderen Anschluss oder Leitung gezogen und dort befestigt werden kann.

Im vorliegenden Beispiel wird der Ausgang der Multiplizierfunktion mit dem Eingang des Anzeigeelements Fläche verbunden.



**Aufgabe:** Fügen Sie noch eine weitere Multiplizierfunktion ein und erstellen Sie die fehlenden Verbindungen.

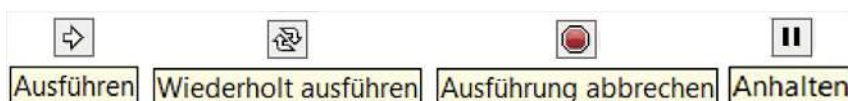
Ist das VI fertig, speichern Sie es mit **Strg+S** unter dem Namen *Programm1*.

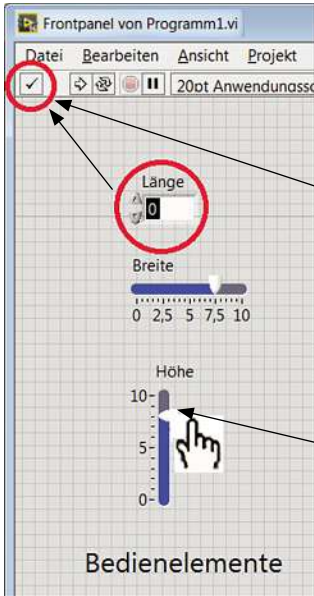
### 1.3 Ausführen eines VI



In der Symbolleiste ganz links findet man den Startknopf zur Ausführung. Wird er betätigt, befindet sich das VI im *Ausführungsmodus*: Das Programm läuft, das Karo im Frontpanel verschwindet.

Die Icons zum Start und Stopp des VI sind in den Symbolleisten beider Fenster vorhanden.






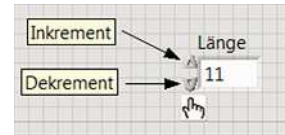
Sinnvolle Ergebnisse erhält man, wenn alle 3 Eingabewerte ungleich null sind. Dazu müssen Werte eingegeben bzw. eingestellt werden. Dies kann **vor** oder **während** der Programmausführung erfolgen. Ein Doppelklick auf das Eingabefeld von **Länge** markiert die enthaltene Zahl schwarz.

Sie kann durch eine nachfolgend eingegebene Zahl ersetzt werden.


Gleichzeitig erscheint am linken Rand der Symbolleiste die Schaltfläche **Texteingabe**.

Diese zeigt den aktiven Eingabemodus an. Er wird beendet durch die Enter-Taste oder einen Mausklick auf die Arbeitsfläche.

Wenn der Mauszeiger sich in eine Bedienhand  verwandelt, kann ein Eingabeelement mit der Maus bedient werden.

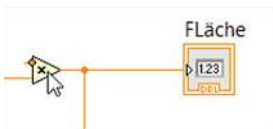


Im vorliegenden Fall ist die Ausführungsdauer sehr kurz, deshalb ist es sinnvoll,

das VI mit  **Wiederholt ausführen** zu starten und eine Änderung der Eingabewerte während des Programmlaufs vorzunehmen. Das Ergebnis wird sehr anschaulich dargestellt.

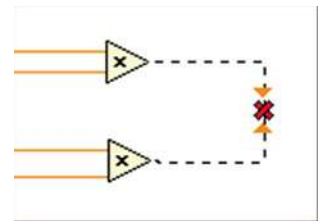
Das VI läuft so lange, bis es mit **Ausführung abbrechen** gestoppt wird.

Ist der **Ausführen**-Pfeil gebrochen , kann das VI nicht gestartet werden. Dies kommt vor, wenn sich Fehler im Programm befinden, wie z.B.



offener Eingang

oder



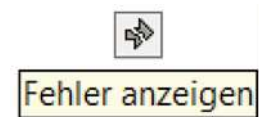
ungültige Verbindung (zwei Ausgänge).

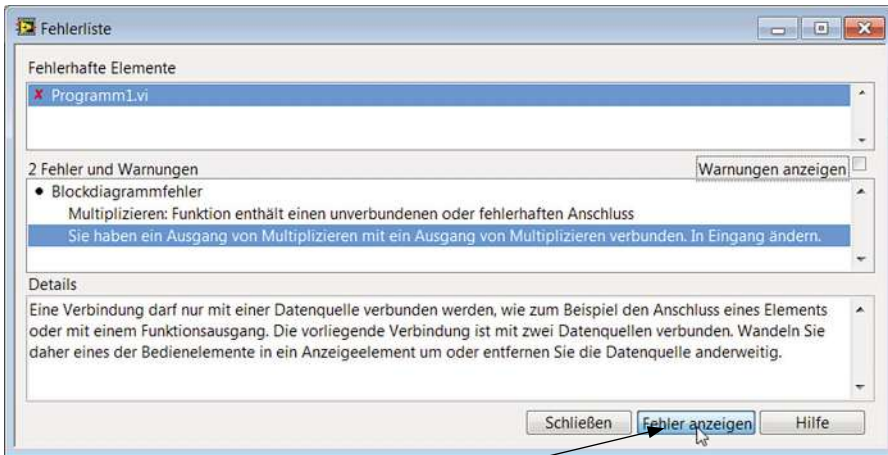


### Merken:

Löschen aller ungültigen Verbindungen mit **Strg+B**.

Klickt man trotzdem auf den gebrochenen **Ausführen**-Pfeil, so wird eine Fehlerliste angezeigt, die eine Beschreibung der vorhandenen Fehler enthält.



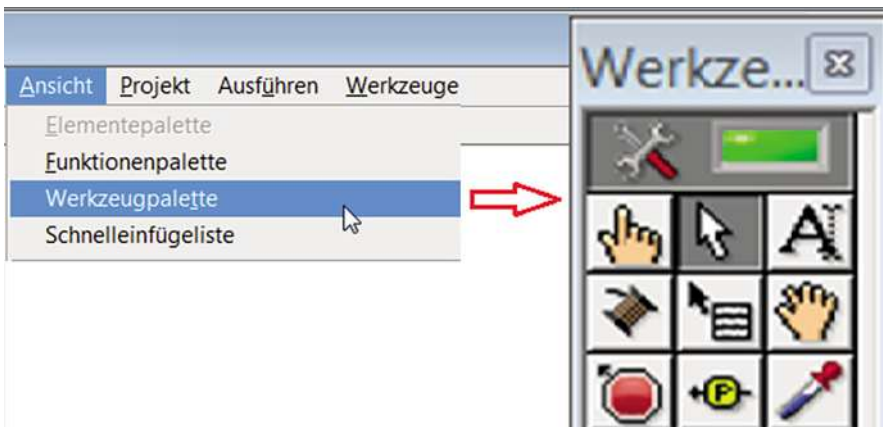


Ein Klick auf die Schaltfläche Fehler anzeigen markiert die Fehlerstelle im Blockdiagramm kurzzeitig mit einem schwarzen Quadrat.

## 1.4 Werkzeugpalette

Beim Bearbeiten eines VI bemerkt man, dass der Mauszeiger an verschiedenen Stellen über einem Objekt verschiedene Formen annimmt. Diese Mauszeigerformen nennt man **Werkzeuge**.

Oft ist es bei der Arbeit einfacher, den automatischen Werkzeugwechsel abzuschalten und ein festes Werkzeug zu wählen. Das ermöglicht die Werkzeugpalette. Der Aufruf der Werkzeugpalette erfolgt im Menü des Frontpanels oder des Blockdiagramms unter *Ansicht* → *Werkzeugpalette* oder *Shift* + Rechtsklick.



Linksklick auf die grüne LED schaltet die automatische Werkzeugauswahl ab (LED erlischt). Das gewählte Werkzeug ändert sich jetzt nicht mehr ortsabhängig. Ein Linksklick auf die graue LED schaltet die automatische Werkzeugauswahl wieder ein (LED grün).

Bedeutung der einzelnen Icons:



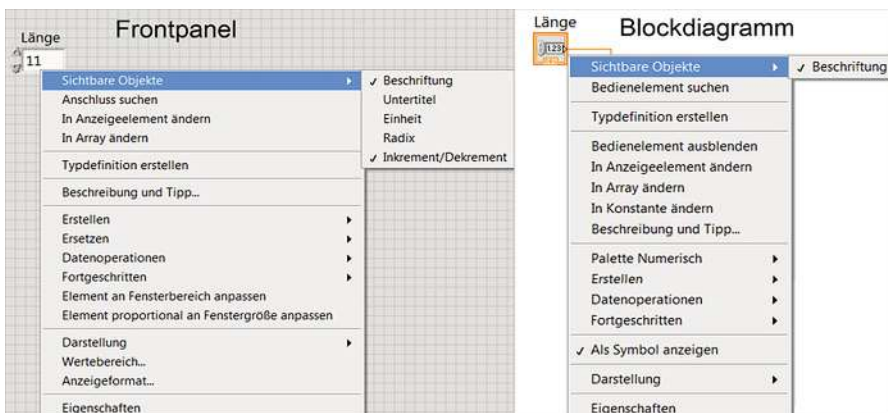
## 1.5 Kontextmenü

Das Kontextmenü von Objekten ist das mächtigste Hilfsmittel bei der Arbeit mit LabVIEW. Es wird mit der **rechten Maustaste** aufgerufen. Dabei ist es wichtig zu wissen, dass jedes Teil sein eigenes Kontextmenü besitzt. In größeren Objekten gibt es viele verschiedene Kontextmenüs, die gewünschte Stelle im Objekt muss mit der Maus genau ausgewählt werden.

Hier einige grundsätzliche Eigenschaften, auf weitere Punkte dieses Menüs wird im Einzelnen noch bei ihrem Gebrauch in den Aufgaben eingegangen.

### 1.5.1 Kontextmenü von Bedien- und Anzeigeelementen

In der nachfolgenden Darstellung wird im Frontpanel und im Blockdiagramm das Kontextmenü des Bedienelements Länge aufgerufen.



#### *Sichtbare Objekte:*

Bezieht sich auf darstellbare Teile von Bedien-/Anzeigeelementen, wie Beschriftungen, Titel, Legenden, Achsen und zusätzliche Anzeigen.

Ein Haken  schaltet ein, kein Haken  schaltet aus.

**Anschluss suchen:**

Ein Linksklick markiert den zugehörigen Anschluss im Blockdiagramm.

**Bedienelement suchen:**

Ein Linksklick markiert das zugehörige Bedienelement im Frontpanel.

**In Anzeigeelement ändern:**

Jedes Bedienelement lässt sich in ein Anzeigeelement, jedes Anzeigeelement in ein Bedienelement umwandeln.

Vorhandene, angeschlossene Leitungen zerbrechen dann, d.h. sind ungültig.

**In Konstante ändern:**

Im Blockdiagramm kann jedes Anzeigeelement und jedes Bedienelement in eine Konstante umgewandelt werden.

Konstanten sind nur im Blockdiagramm sichtbar!

**Eigenschaften:**

Hier kann das Erscheinungsbild von Elementen wie Farbe, Wertebereich, Anzeigeformat, Skalenart eingestellt werden. Für Bedienelemente kann eine Tastatursteuerung eingestellt werden.

**Erstellen:**

Im Blockdiagramm kann LabVIEW einen Anschluss automatisch beschalten. Ein Ausgang erhält ein Anzeigeelement, ein Eingang ein Bedienelement oder eine Konstante. Sehr hilfreiche Funktion!

## 1.5.2 Kontextmenü von Funktionsobjekten im Blockdiagramm

Kontextmenü des Funktionsobjekts Multiplizieren

**Multiplizieren (Funktion)**

Übergeordnete Palette: [Numerische Funktionen](#)

Erfordert: Base Package

Gibt das Produkt der Eingangswerte aus.

**Diagramm:** Ein Blockdiagramm zeigt zwei Eingangsanschlüsse 'x' und 'y' in gelben Kreisen, die in einen Multiplikationsblock (ein Quadrat mit einem 'x' in der Mitte) führen. Ein Ausgangsanschluss 'x\*y' in einem gelben Kreis führt von dem Block nach rechts. Ein Fehlerausgang 'Fehler (Ausgang)' in einem roten Kreis führt von dem Block nach unten. Ein Text 'Fehler (Eingang, kein Fehler)' ist mit einem roten Pfeil an den Eingang 'y' gebunden.

Zum Blockdiagramm hinzufügen    Auf Palette suchen

---

**DBL N** **x** kann zum Beispiel eine skalare Zahl, ein Array oder Cluster mit numerischen Werten oder ein Array aus Clustern mit numerischen Werten sein.

**DBL N** **y** kann zum Beispiel eine skalare Zahl, ein Array oder Cluster mit numerischen Werten oder ein Array aus Clustern mit numerischen Werten sein.

**DBL N** **x\*y** ist das Produkt von **x** multipliziert mit **y**.

**Hilfe:**

Alle Funktionsobjekte besitzen im Kontextmenü eine leistungsfähige Hilfefunktion, die eine genaue Beschreibung des Funktionsobjekts liefert.

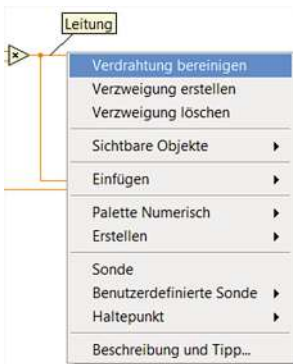
**Erstellen:**

Anschlüsse können automatisch mit Bedienelement, Anzeigeelement oder Konstante beschaltet werden. Sehr hilfreiche Funktion!

**Ersetzen:**

Bereits bestehende Objekte können durch andere ähnliche ersetzt werden. Das erspart das Erstellen von neuer Verdrahtung.

### 1.5.3 Kontextmenü einer Verbindungsleitung

**Verdrahtung bereinigen:**

Erstellt kurze, übersichtliche Verbindung.

**Verzweigung löschen:**

Löscht Verbindungen.

**Einfügen:**

Erlaubt das Einfügen von Funktionsobjekten in Leitungen.

**Sonde:**

Setzt eine Sonde auf die Leitung.

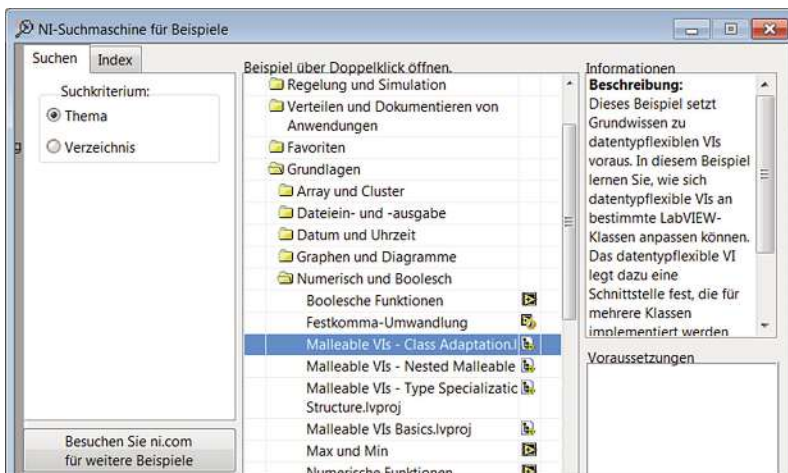
**Haltepunkt:**

Setzt einen Haltepunkt auf die Leitung.

## 1.6 Hilfsmittel

### 1.6.1 Beispiele suchen

Aufruf im Startfenster oder im Hauptmenü unter *Hilfe* → *Beispiele suchen*.

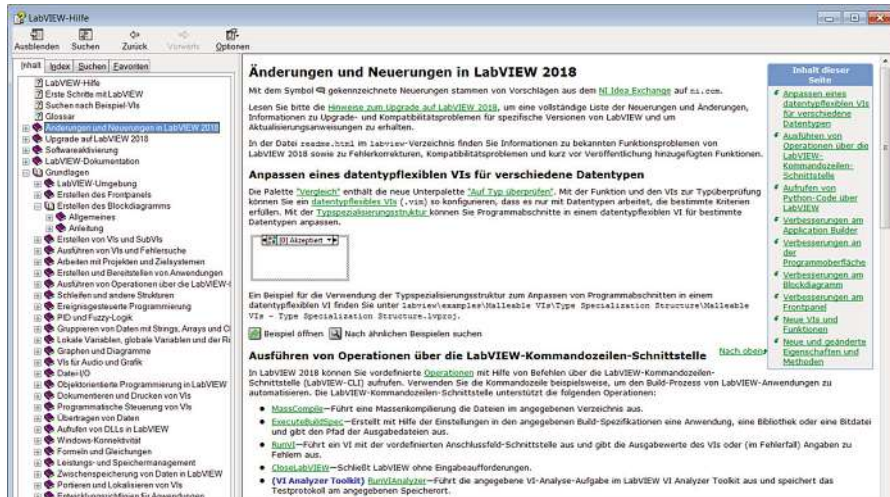




Nach Aufruf bietet eine Suchmaschine eine Vielzahl fertiger VIs, an denen die Arbeitsweise und Beschaltung von Objekten untersucht werden kann.

## 1.6.2 LabVIEW-Hilfe

Aufruf im Hauptmenü unter *Hilfe* → *LabVIEW Hilfe* oder *Strg+?* oder *F1*.




Dahinter verbirgt sich die offizielle Referenz für LabVIEW. Alles Wissenswerte über LabVIEW, alle gültigen Begriffe (Glossar!), Regeln und Festlegungen findet man hier.

Es gibt vier Wege (Registerkarten) zum gesuchten Thema:

1. eine Explorer-Darstellung *Inhalt*, kapitelweise sortiert ähnlich einem Fachbuch,
2. eine nach Stichwörtern sortierte alphabetische Liste *Index*,
3. eine Suchfunktion *Suchen*,
4. eine Registerkarte *Favoriten* zur Speicherung von oft benutzten Hilfe-Seiten. Sehr hilfreich!

## 1.6.3 Kontexthilfe

Aufruf im Hauptmenü unter *Hilfe* → *Kontexthilfe* anzeigen oder *Strg+H* oder ganz

rechts in der Symbolleiste mit 

Die Kontexthilfe ist die kleine Schwester der LabVIEW-Hilfe. Wird sie aufgerufen, erscheint ein kleines Fenster auf der Arbeitsfläche, das eine Minihilfe zu dem vom Mauszeiger ausgewählten Objekt bietet.

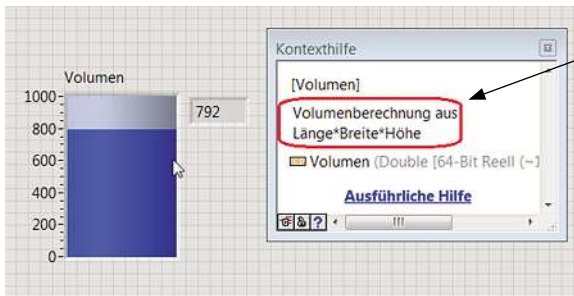


Die Kontexthilfe ist aktiv:

- im Blockdiagramm,
- im Frontpanel,
- im Bearbeitungsmodus,
- im Ausführungsmodus.

[Ausführliche Hilfe](#) führt zur großen Schwester LabVIEW Hilfe.

Im Frontpanel kann die Kontexthilfe einen benutzerspezifischen Text anzeigen:



Erstellt wird der Text im Kontextmenü des Objekts unter:  
[Eigenschaften](#) → Reiter [Dokumentation](#) → [Beschreibung](#).

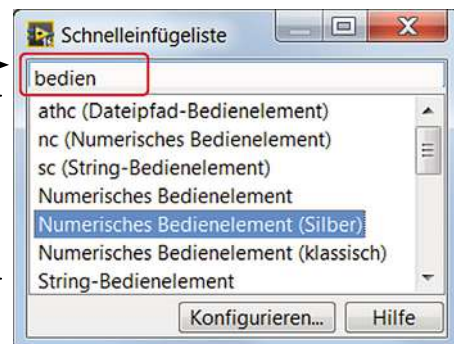
## 1.6.4 Hilfsmittel im Bearbeitungsmodus

### Schnelleinfügeliste

Aufruf im Hauptmenü unter [Ansicht](#) → [Schnelleinfügeliste](#) oder [Strg+Leertaste](#).

Die Schnelleinfügeliste erlaubt das Auffinden der LabVIEW-Objekte über die Namensangabe. Beispiel: Gesucht wird im Frontpanel ein numerisches Bedienelement.

Nach Aufruf öffnet sich ein Fenster mit einem Eingabefeld für das gesuchte Objekt. Darunter erscheinen die zur Eingabe passenden Indexeinträge. Mit einem Doppelklick wird das gesuchte Objekt ausgewählt und kann platziert werden.






## 1.6.5 Hilfsmittel im Ausführungsmodus

Sind im Programm logische Fehler enthalten, d.h. das Programm läuft, macht aber nicht, was es soll, so gibt es Hilfsmittel, diese aufzuspüren.

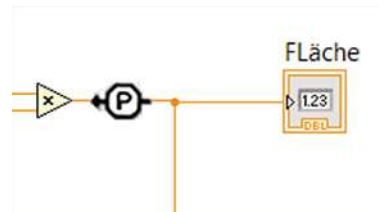
### Highlight-Funktion

Damit ist der Datenfluss zu verfolgen. Eingeschaltet wird die Glühlampe in der Symbolleiste des Blockdiagramms. Nach dem Start des Programms werden die aktuellen Daten, die Orte ihrer Änderung und ihre Wege dargestellt.

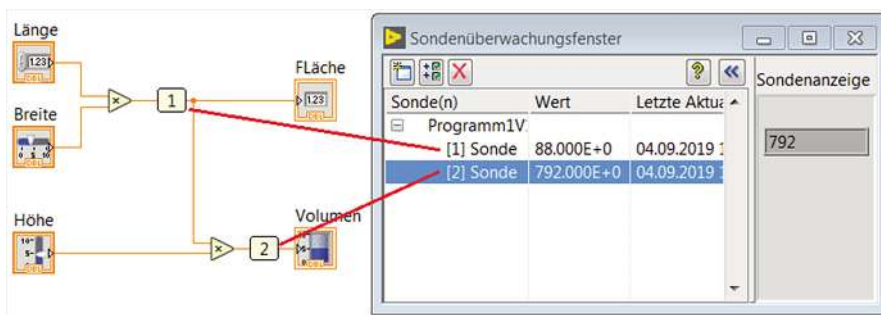


Sonden  P steht für *engl.: probe*: Tastkopf, Sonde.

Bei Fehlersuche in großen Programmen ist die Highlight-Funktion etwas unübersichtlich und langsam. Zur Verfolgung der Programmfunktion können auf Leitungen Sonden angebracht werden. Bringt man im Ausführungsmodus den Mauszeiger über eine Leitung, erscheint ein Sondenzeichen. Ein Linksklick befestigt die Sonde.

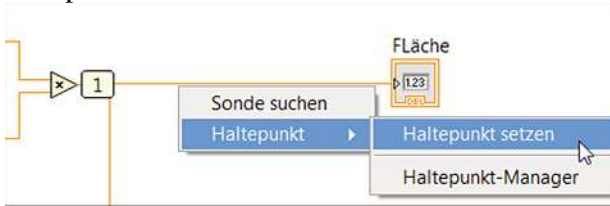


In einem Sondenüberwachungsfenster werden die zu der Sonde gehörigen Daten dargestellt.



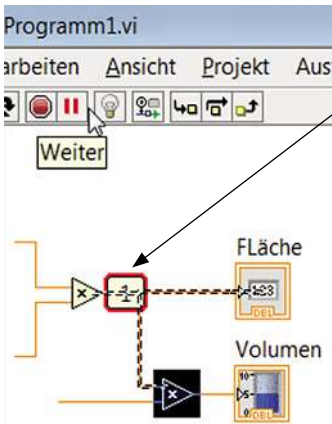
Die Befestigung von Sonden ist auch im Bearbeitungsmodus (Kontextmenü) möglich.

## Haltepunkt



Zur genauen Untersuchung von Sondenwerten kann ein laufendes Programm an gewünschten Stellen unterbrochen werden.

Ein Haltepunkt wird im Kontextmenü eines Funktionsobjektes oder einer Leitung gesetzt.



Hält die Ausführung des VI an, wird das mit einer Markierung gekennzeichnet.

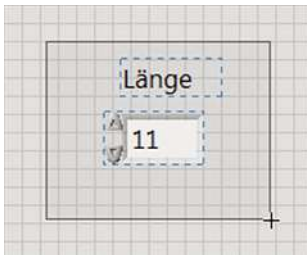
Ein Klick auf die Pause-Taste startet das Programm wieder.

Haltepunkte können mit dem Haltepunkt-Manager bearbeitet werden.

## 1.7 Umgang mit Objekten

### 1.7.1 Markieren von Objekten

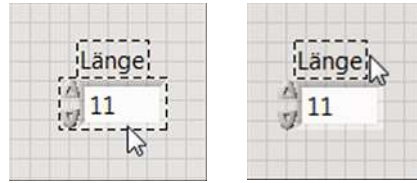
Eine Markierung erkennt man an einer sich bewegend, blauen Umrandung (Ameisen). Die Markierung von Objekten, insbesondere von Bedien- und Anzeigeelementen, kann in LabVIEW etwas schwierig werden, da sich der Mauszeiger bei Bewegung über dem Objekt oft ändert.



Die einfachste Form der Markierung ist das Aufziehen eines Rechtecks über ein oder mehrere Objekte. Dies ist auf den Arbeitsflächen Frontpanel und Blockdiagramm möglich, wenn der Mauszeiger ein Pluszeichen + bildet.

Hat das Maussymbol die normale Pfeilform, markiert ein Linksklick das gewählte Objekt einschließlich Beschriftung.

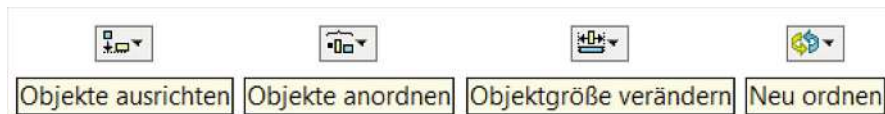
Zeigt der Mauszeiger auf die Beschriftung, wird nur diese markiert und kann getrennt vom Eingabefeld bewegt, aber nicht gelöscht werden.



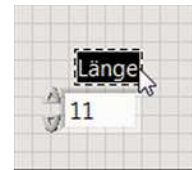
Markierte Objekte können

- ❑ wie bei einem Textverarbeitungsprogramm mit **Strg+X** ausgeschnitten, mit **Strg+C** in die Zwischenablage kopiert und mit **Strg+V** wieder auf der Arbeitsfläche eingefügt werden,
- ❑ mit einem Mauszeiger in Pfeilform oder mit den **Shift+←↑↓→** Pfeiltasten der Tastatur verschoben werden (in beiden Arbeitsflächen ist eine Einrastfunktion am Gitter mit **Strg+#** ein- und ausschaltbar),
- ❑ auf beiden Arbeitsflächen automatisch ausgerichtet und angeordnet werden (siehe Icons in den Symbolleisten **Objekte ausrichten/anordnen**),
- ❑ in Vorder- oder Hintergrund gebracht werden (unter Icon **Neu ordnen**).

### Icons Symbolleisten



Die **Markierung von Text** erkennt man an dem schwarzen Hintergrund. Zahlen in einem Eingabefeld oder Text in einem Beschriftungsfeld werden durch einen Doppellincksklick markiert. Ein Dreifachklick markiert alle Wörter in einem Textfeld.

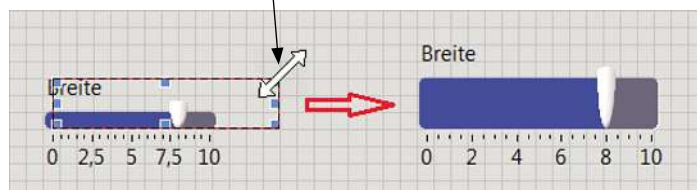
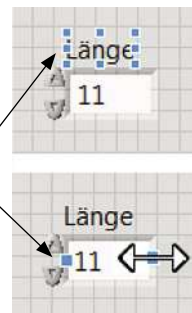


### 1.7.2 Objektgröße ändern

Bewegt man im Bearbeitungsmodus die Maus über Bedien- oder Anzeigeelemente, so tauchen an den Eingabefeldern, den Beschriftungen, an Schiebereglern oder anderen Flächen blaue Punkte auf. Diese so genannten

«Anfasspunkte» erlauben die Veränderung der Abmessungen eines Elements. Nähert man sich einem Anfasspunkt, wird der Mauszeiger zu einem waagrechten, senkrechten oder schrägen Pfeil, der die mögliche Aufziehrichtung angibt.

«Anfasspunkte»



## 1.8 Express-VI

Ein aufmerksamer Benutzer von LabVIEW bemerkt bald, dass es im Blockdiagramm auf der Funktionenpalette (unten mit  $\vee$  aufklappen) unter *Express* so genannte Express-VIs gibt, die eine umfangreiche Funktionalität aufweisen. Es sind vorgefertigte VIs, die bei Platzierung auf der Arbeitsfläche ein Konfigurationsfenster öffnen, in dem der Benutzer Einstellungen vornehmen kann.



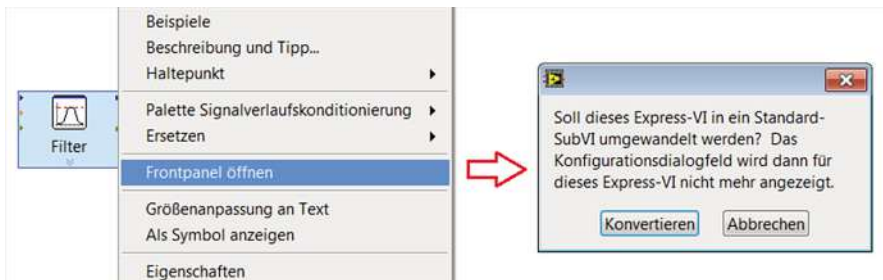
### Beispiel:

Unter *Funktionen* → *Express* → *Arithmetik & Vergleich* → *Formel* findet man ein Express-VI zur arithmetischen Berechnung.



Wird in das Konfigurationsdialogfeld eine Formel eingegeben und mit OK bestätigt, passt das Express-VI sein Erscheinungsbild an und bietet Anschlussmöglichkeiten für die eingegebenen Variablen. Diese können dann entsprechend beschaltet werden. Viele Express-VIs werden durch Doppelklick geöffnet und können mit dem erscheinenden Dialogfeld an ihre Aufgabe angepasst werden. Möchte man diese Einstellmöglichkeiten im Ausführungsmodus ändern, findet man die Anschlüsse dafür durch Aufziehen an den Anfasspunkten.

Nach dem Motto «Konfigurieren statt Programmieren» stehen für Ein-/Ausgabe-Operationen und Signalbearbeitung zahlreiche Express-VIs zur Verfügung. Viele Aufgabenstellungen sind damit relativ einfach zu lösen, das Verständnis von LabVIEW nimmt aber leider nicht zu, da die internen Funktionen verborgen bleiben. Nicht verborgen bleibt die innere Funktionalität, wenn man im Kontextmenü die Möglichkeit der Umwandlung eines Express-VIs in ein SubVI (Unterprogramm) nutzt.



Im Kontextmenü *Frontpanel öffnen* oder *In SubVI Umwandeln* anklicken, und es öffnet sich ein VI mit Frontpanel und Blockdiagramm, an dem sich der Aufbau des VI studieren lässt.

## 1.9 Diverses

### 1.9.1 Passwortschutz

Ein VI kann mit einem Passwort geschützt werden. Das Programm kann gestartet und bedient werden, es ist jedoch kein Zugriff auf das Blockdiagramm mehr möglich. Die Passwortvergabe findet man im Hauptmenü unter *Datei* → *VI-Einstellungen* → *Kategorie* → *Schutz*. Der Schutz wird erst wirksam, wenn entweder

1. das Programm gespeichert, LabVIEW geschlossen und neu gestartet wird oder
2. im Hauptmenü unter *Werkzeuge* → *Optionen* → *Umgebung* → *Passwort Cache löschen* gewählt wird.

Wird einem gesperrten Programm das Passwort eingegeben, wird dieses im Passwort Cache gespeichert und bleibt so lange bearbeitbar, bis dieser wieder gelöscht wird.

### 1.9.2 EXE-Programm

Von einem LabVIEW-Programm *progrname.vi* kann ein direkt vom Betriebssystem aufrufbares Programm *progrname.exe* hergestellt werden. Dies ist jedoch nur mit der Vollversion von LabVIEW und nicht mit der Studentenversion möglich.

### 1.9.3 Blockdiagramm neu zeichnen

Klickt man in der Symbolleiste des Blockdiagramms des Icons «Aufräumen»



an, so wird es neu gezeichnet.

Die Parameter dafür sind einstellbar im Hauptmenü unter

*Werkzeuge* → *Optionen* → *Blockdiagramm* → *Blockdiagramm aufräumen*.

Falls das Ergebnis nicht gefällt, stellt **Strg+Z** den alten Zustand wieder her.

### 1.9.4 VI speichern

Hauptmenü *Datei* → *Speichern* oder *Strg+S*

oder mit neuem Namen speichern:

Hauptmenü *Datei* → *Speichern unter*. Die Voreinstellung schließt das aufgerufene Programm unverändert und speichert das aktuelle unter dem neuen Namen.

### 1.9.5 VI ausdrucken

Hauptmenü *Datei* → *Drucken*.

### 1.9.6 Tastenkombinationen (Shortcuts)

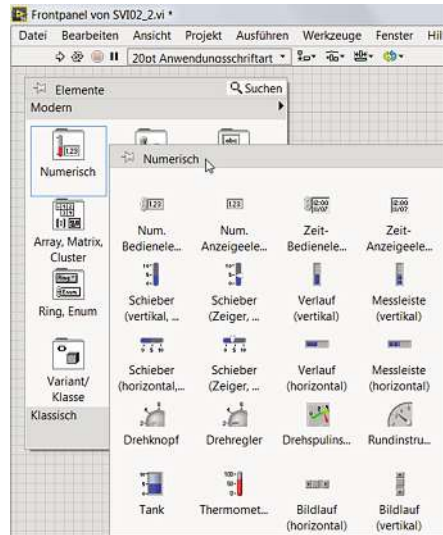
Strg+Z	letzten Befehl rückgängig machen
Strg+T	beide Fenster (Frontpanel und Blockdiagramm) öffnen
Strg+/	Fenster auf Bildschirmgröße erweitern
Strg+E	Zwischen beiden Fenstern umschalten
Strg+B	alle gebrochenen Leitungen löschen
Strg+R	VI starten
Strg+.	Ausführung des laufenden VI stoppen
Strg+?	LabVIEW-Hilfe
Strg+H	Kontexthilfe einschalten / ausschalten
Strg+N	Neues VI
Strg+S	VI speichern
Strg+C	markiertes Objekt in Zwischenablage kopieren
Strg+V	Inhalt der Zwischenablage einfügen
Strg+X	markiertes Objekt löschen
Strg+A	alle Objekte markieren
Strg+#	Einrasten am Gitter einschalten / ausschalten
Strg+Shift	Verschiebeband; zum Verschieben der Arbeitsfläche
Shift+←↑↓→	Schnelles Verschieben markierter Objekte

#### Nicht vergessen!

- Mit Strg+Z kann man gemachte Fehler ungeschehen machen.
- Kopieren von Programmteilen immer nur im Blockdiagramm.
- Anfasspunkte (kleine blaue Vierecke) eines Objekts, weisen auf Aufziehmöglichkeit hin.
- STRG+B löscht alle gebrochenen Leitungen.
- Der Stoppknopf der Symbolleiste verhindert die Ausgabe von Schleifeninhalten.
- Das Kontextmenü *Erstellen* → ... liefert immer den passenden Datentyp zum Anschluss.
- Das Kontextmenü von Objekten ist an verschiedenen Stellen unterschiedlich.
- Zusammengesetzte Objekte markiert man mit einem aufgezogenen Rahmen.
- Ein Klick auf den gebrochenen Ausführungsknopf zeigt alle vorhandenen Fehler.
- Die Kontexthilfe zeigt die Datentypen von Leitungen und Anschlüssen.
- Die Kontextmenüpunkte *Ersetzen* und *Einfügen* sparen viel Verdrahtungsarbeit.
- Auch Leitungen haben ein Kontextmenü.

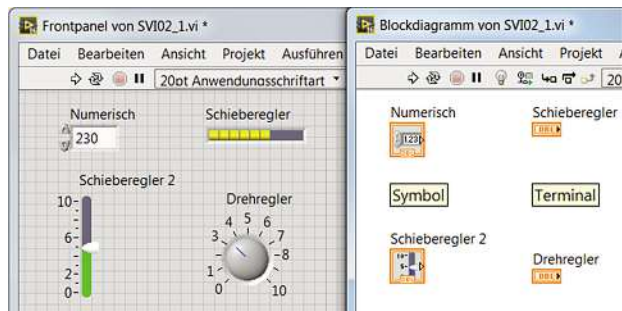
## 2 Numerische Funktionen

Bedien- und Anzeigeelemente sind zu finden im Kontextmenü des Frontpanels unter *Elemente* → *Modern* → *Numerisch*.



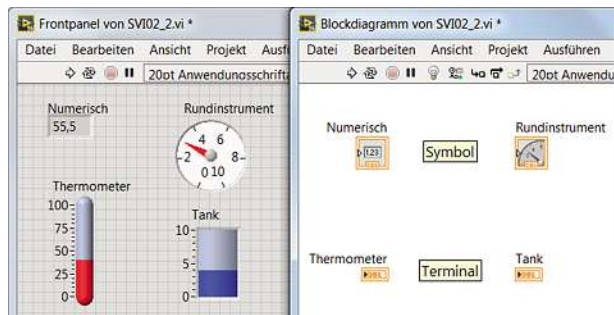
### 2.1 Bedienelemente zur Eingabe numerischer Daten (Datenquellen)

Datenquellen haben auf der rechten Seite einen «Ausgangs»-Pfeil und sind dick umrandet.



### 2.2 Anzeigeelemente zur Ausgabe numerischer Daten (Datensenken)

Datensenken haben auf der linken Seite einen «Eingangs»-Pfeil und sind dünn umrandet.





LabVIEW verwendet zur Zahlendarstellung verschiedene Datentypen:

Fließkommazahlen	
Festkommazahlen	
ganze Zahlen ohne Vorzeichen	
ganze Zahlen mit Vorzeichen	
komplexe Zahlen	

Der Unterschied zwischen den Datentypen besteht in der Anzahl der Bits, die zur Speicherung der Daten verwendet werden. Mit dem Datentyp ist der Wertebereich der Zahlen festgelegt.

Die Voreinstellung des Datentyps ist *DBL* (double), dies steht für Gleitkommazahl mit doppelter Genauigkeit (64 Bit). Der Wertebereich einer Zahl dieses Datentyps umfasst etwa den Bereich von  $\pm 2 \cdot 10^{308}$ .

Bei Verwendung anderer Datentypen wie z.B. *I8* (Integer 8 Bit) oder *U8* (Unsigned 8 Bit) ist bei arithmetischen Berechnungen der zulässige Wertebereich zu beachten.



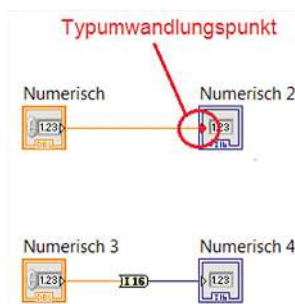
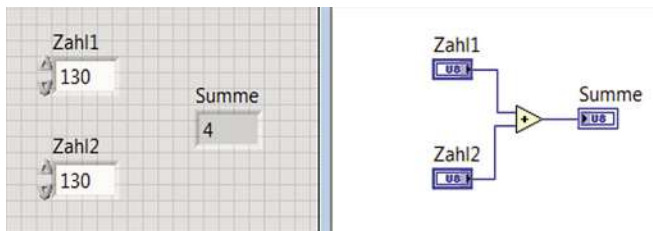
#### Merksatz:

Bei Bereichsüberschreitung wird keine Fehlermeldung ausgegeben.



#### Beispiel:

Bei einer Addition und nachfolgender Anzeige wird der zulässige Wertebereich von vorzeichenlosen Ganzzahlen mit 8 Bit (0 bis 255) überschritten!



Im Frontpanel ist der verwendete Datentyp nicht zu erkennen!

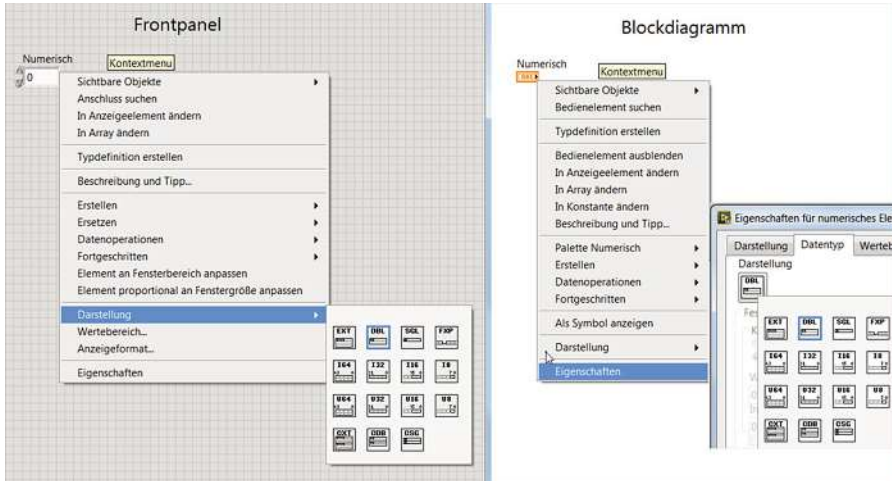
Werden Objekte unterschiedlicher Datentypen miteinander verbunden, so wird intern automatisch eine Datentypumwandlung vorgenommen und mit einem roten Typumwandlungspunkt angezeigt.

Besser ist es, eine Typumwandlung mit einer *Konvertierungsfunktion* (hier: *Nach Wort-Integer*) vorzunehmen, da eine automatische Umwandlung die Laufzeit erhöht und mehr Speicherplatz benötigt.



Weitere Informationen zu den numerischen Datentypen und ihren Wertebereichen findet man in der LabVIEW-Hilfe.

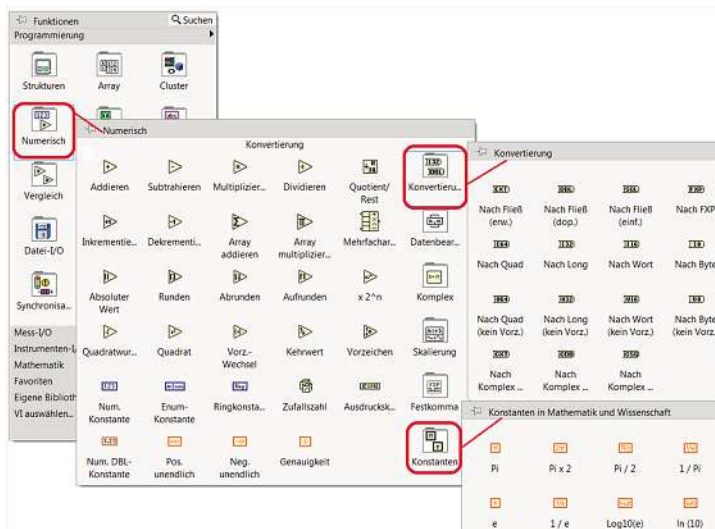
Die Auswahl des Datentyps erfolgt im Frontpanel oder im Blockdiagramm mit dem Kontextmenü, entweder über *Darstellung* oder über *Eigenschaften*.



### Wichtig:

Auch bei Konstanten ist ein Datentyp wählbar.

Sind Ein- und Ausgabeelemente auf dem Frontpanel platziert, können sie im Blockdiagramm verbunden werden. Die Objekte zur Bearbeitung numerischer Funktionen findet man auf der Palette *Funktionen* → *Programmierung* → *Numerisch*.

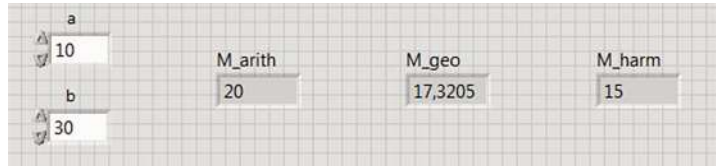


**Aufgabe:** Erstellen Sie ein VI, das den arithmetischen, geometrischen und harmonischen Mittelwert zweier positiver Ganzzahlen berechnet.

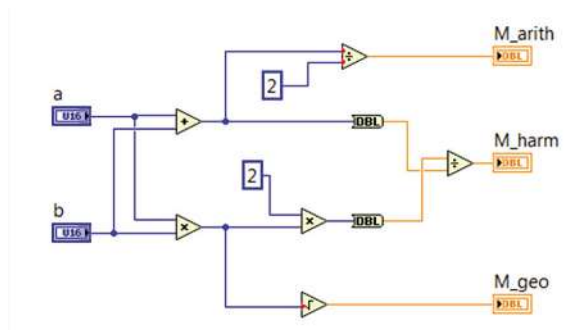
*Hinweis:* Wegen der positiven Ganzzahlen werden numerische Bedienelemente mit Datentyp *U16* gewählt. Die Anzeigeelemente müssen Datentyp *DBL* sein. Eine Konstante im Blockdiagramm wird mit Kontextmenü *Erstellen* → *Konstante* am Anschluss des Knotens erzeugt.

Formeln:  $M_{arith} = \frac{a + b}{2}$      $M_{geo} = \sqrt[3]{ab}$      $M_{harm} = \frac{2ab}{a + b}$

Frontpanel:



Blockdiagramm:



Wird das vorliegende VI mit den Zahlen  $a=0$  und  $b=0$  gestartet, liefert die Berechnung des harmonischen Mittelwerts den Wert NaN, was für **Not a Number** steht, da der Bruch  $\frac{0}{0}$  nicht definiert ist.



Dieser Fehler passiert häufig, wenn man ein VI neu aufruft und die Anfangswerte Null sind. Damit die Anfangswerte **nicht** Null sind, muss das VI mit Zahlen  $\neq 0$  gespeichert werden.

Zahlen in Eingabelementen werden zusammen mit dem VI gespeichert, wenn man sie vorher im Kontextmenü → *Datenoperationen* → *Aktuellen Wert als Standard* deklariert.

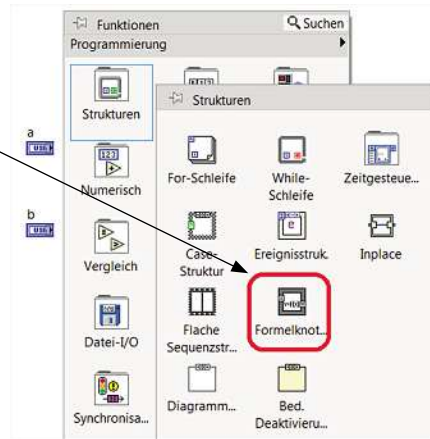


**Aufgabe:** Starten Sie das VI mit den Zahlen  $a = 127$  und  $b = 259$  und kommentieren Sie das Ergebnis von  $M_{\text{harm}}$ . Verwenden Sie die Kontexthilfe zur Anzeige des Datentyps der Leitung.

## 2.3 Formelknoten

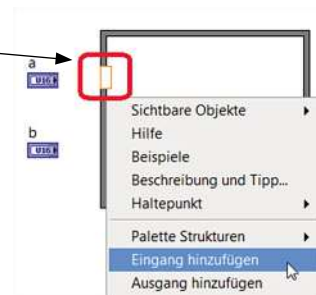
Die gleiche Aufgabe kann auch mit einem **Formelknoten** gelöst werden. Diesen findet man auf der Palette: **Funktionen** → **Programmierung** → **Strukturen** → **Formelknoten**.

Vorgehensweise:

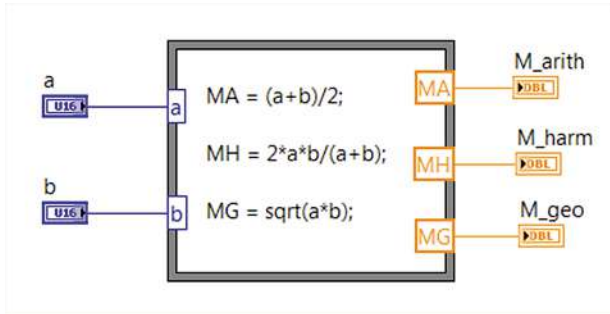


Formelknoten auswählen und aufziehen.

Mit Kontextmenü Eingänge (Eingangsvariable) und Ausgänge (Ausgangsvariable) hinzufügen und mit beliebigen Variablennamen beschriften.



In einer C-ähnlichen Syntax die gewünschten Formeln in den Formelknoten eintragen.



### Wichtig:

Jede Zuweisung muss mit einem Semikolon (;) enden.

## 2.4 Ausdrucksknoten

Ein **Ausdrucksknoten** arbeitet ähnlich, jedoch ist nur **eine** Eingangs- und Ausgangsvariable erlaubt. Damit sind einfache algebraische Ausdrücke berechenbar.



### Beispiel:

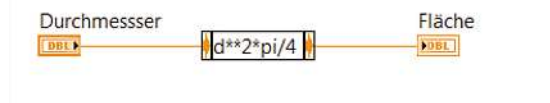
Aus dem Kreisdurchmesser soll die Kreisfläche berechnet werden.

Formel: 
$$A = \frac{d^2 \cdot \pi}{4}$$

Frontpanel:



Blockdiagramm:



Hier ist d der Eingangswert; der Ausgang liefert das Ergebnis der Berechnung.