

IN DIESEM KAPITEL

Allgemeiner Aufbau und Prinzip von LabVIEW

Kennenlernen von Werkzeugen und Paletten, die zur Programmierung zur Verfügung stehen

Anwendung des Kontextmenüs als wichtiges Hilfsmittel in LabVIEW

Kapitel 1

LabVIEW – die etwas andere Art der Programmierung

In diesem Kapitel erklären wir Ihnen, was LabVIEW ist, wofür es eingesetzt wird und wie man es benutzt. Sie werden die Bedienoberfläche und deren einzelne Elemente kennenlernen (Frontpanel, Blockdiagramm, Connector Pane, Icon). Um die Bedienoberfläche von LabVIEW mit Leben zu füllen, stehen verschiedene Werkzeuge und Paletten zur Verfügung, welche wir Ihnen in diesem Kapitel ebenfalls vorstellen werden. Ein wichtiges Hilfsmittel in LabVIEW ist das Kontextmenü, denn man kann mit dessen Hilfe wichtige Zeit beim Programmieren sparen. Deshalb wollen wir es Ihnen in diesem Kapitel zeigen und kurz auf die wichtigsten Punkte eingehen.

LabVIEW ist eine graphische Programmiersprache, deren Quellcode bzw. Programmlogik aus graphischen Elementen besteht. Die Abkürzung steht für **L**aboratory **V**irtual **I**nstrument **E**ngineering **W**orkbench. Im Gegensatz zu textbasierten Programmiersprachen (wie zum Beispiel C++) können Sie mit LabVIEW Laborprozesse anschaulich mit bewegten Bildern darstellen. Zum Beispiel können Sie einen Kessel im Labor simulieren, in dem eine bunt gefärbte Flüssigkeit auf- und absteigt oder einen Farbumschlag in Abhängigkeit vom pH-Wert darstellen. Das klingt doch verlockender als irgendwelche kryptischen Texte, die schwarz auf weiß über einen Bildschirm flackern, oder?

Deshalb wird LabVIEW gerne in Ingenieur- und Naturwissenschaften eingesetzt. Mithilfe von LabVIEW können (Mess-)Geräte über sogenannte virtuelle Instrumente (VI) angesteuert werden. Diese virtuellen Instrumente werden von Ihnen in LabVIEW programmiert, das heißt sie sind das vollständige Programm. Jedes VI besteht aus dem Frontpanel, dem Blockdiagramm, dem Connector Pane und dem Icon. Das Frontpanel enthält Anzeige- und Bedienelemente, das Blockdiagramm enthält die eigentliche Programmlogik und das Connector Pane wird zur Erstellung von Unterprogrammen (SubVIs) genutzt. Das Icon wird zur Identifizierung des SubVIs angelegt, damit man es im Hauptprogramm schneller findet.

In den nächsten Unterkapiteln werden wir noch ausführlicher auf die Funktionen dieser vier Elemente eingehen. Jedes VI ist eigenständig lauffähig, kann aber auch beliebig als Sub-VI mit weiteren virtuellen Instrumenten verschachtelt werden, so dass auch hochkomplexe Programme erstellt werden können.

So wird die Programmiersprache sogar von der NASA eingesetzt. Wundern Sie sich also bitte nicht, wenn im Verlauf des Buches immer wieder Beispiele mit Bezug zur NASA auftauchen. Diese dienen zum besseren Verständnis, weil die meisten Menschen sich unter dem Begriff NASA etwas vorstellen können.

Datenflussprinzip

Die Programmierung in LabVIEW basiert auf dem Datenflussprinzip: graphische Symbole werden über Drähte (Wires) miteinander verbunden, so dass ein Flussdiagramm entsteht, welches von links nach rechts gelesen bzw. abgearbeitet wird. Die Farbe und Dicke der Drähte ist abhängig von dem verwendeten Datentyp. Im Verlauf des Datenflusses werden die Daten von einem Knoten (Node) zum nächsten übergeben. Nur wenn an allen Eingängen eines Knotens Daten verfügbar sind, wird dieser abgearbeitet und die Daten am Ausgang des Knotens weitergegeben. In Abbildung 1.1 ist ein einfacher Multiplikationsvorgang dargestellt. Das Multiplikationszeichen ist in diesem VI der Knoten. Erst wenn an beiden Eingängen des Knotens die Daten von Faktor 1 und Faktor 2 verfügbar sind, wird der Knoten abgearbeitet und das Ergebnis ausgegeben. In Kapitel 8 zeigen wir Ihnen, wie Sie den Datenfluss im Rahmen der Fehlersuche (Debugging) sichtbar machen können.

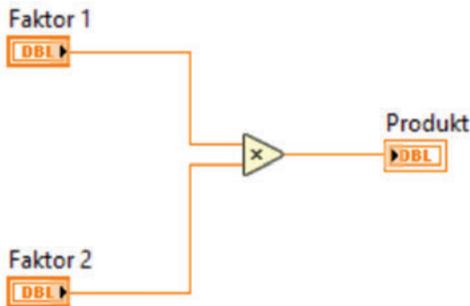


Abbildung 1.1: Knoten im Datenflussdiagramm

Der Datenfluss in LabVIEW wird über die Drähte gesteuert. Erst wenn alle Daten (a,b,c) an dem Block d anliegen – wie in Abbildung 1.2 dargestellt – wird der Block d ausgeführt. Bei der Programmierung sollte darauf geachtet werden, dass der Datenfluss immer von links nach rechts verläuft.

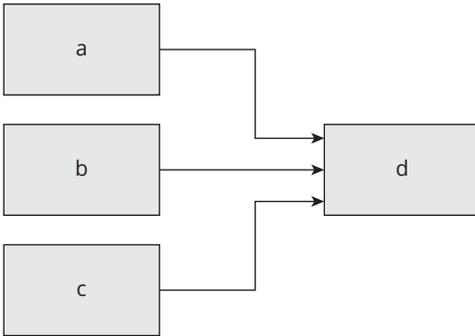


Abbildung 1.2: Erklärung zum Datenfluss in LabVIEW

Denn in LabVIEW ist es prinzipiell auch möglich Daten von rechts nach links laufen zu lassen (Abbildung 1.3b) – dies entspricht jedoch keiner guten Programmierweise und sollte nicht vorkommen!



(a) Datenfluss von links nach rechts

(b) Datenfluss von rechts nach links

Abbildung 1.3: Anordnung der Elemente in LabVIEW

Man sollte dem Programmfluss immer folgen können, indem man dem Verlauf der Drähte von links nach rechts folgt. Das wäre in Abbildung 1.3b nicht der Fall, weil der Draht vom Element y von rechts nach links verläuft.

Allerdings kann es auch mal vorkommen, dass bestimmte Programmteile voneinander unabhängig sind und deshalb nicht über Drähte verbunden sind. Diese voneinander unabhängigen Bereiche werden in der Regel parallel abgearbeitet. Jedoch ist dafür auch die Anzahl der verfügbaren Prozessoren ausschlaggebend, wie viel tatsächlich parallel abgearbeitet werden kann. Deshalb kann man in einem solchen Fall nicht genau vorhersagen, welcher Programmteil als erstes ausgeführt wird. In Abbildung 1.1 haben Sie einen einfachen Multiplikationsvorgang gesehen. Nehmen wir einmal an, dass in diesem Programm eine weitere Multiplikation von einem Faktor 3 und einem Faktor 4 durchgeführt werden müsste (Abbildung 1.4); dann würde sich der dazugehörige Code wahrscheinlich unter dem ersten Multiplikationsvorgang im Blockdiagramm befinden. Beide Programmteile wären voneinander unabhängig und würden als Ergebnis Produkt und Produkt 2 ausgeben. Man weiß in diesem Fall nie, welche der beiden Rechnungen als erstes abgearbeitet wird:

Im Normalfall werden beide parallel ausgeführt; es kann aber auch passieren, dass zuerst der obere oder zuerst der untere ausgeführt wird.

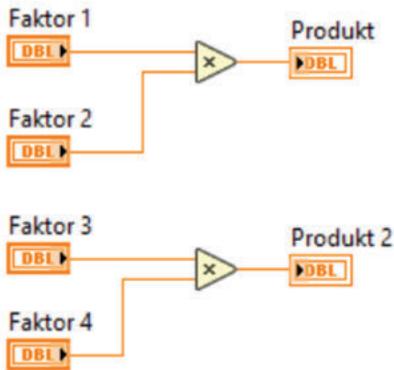


Abbildung 1.4: Zwei voneinander unabhängige Knoten im Datenflussdiagramm



Man weiß nie, in welcher Reihenfolge voneinander unabhängige Programmteile abgearbeitet werden!

Sollte es in einem Ihrer Programme wichtig sein, dass bestimmte Schritte nacheinander abgearbeitet werden, sollten diese in irgendeiner Form über Drähte (von links nach rechts) miteinander verbunden sein, um dem Datenflussprinzip Folge zu leisten.

Benutzeroberfläche

Wenn LabVIEW gestartet wird, erscheint die Oberfläche wie in Abbildung 1.5 dargestellt. In neueren LabVIEW-Versionen kann die Oberfläche auch schon wieder anders aussehen (z. B. in der Farbgestaltung). Die im Rahmen dieses Buches beschriebenen Elemente sollten Sie jedoch immer vorfinden.

Über die Schaltfläche FILE gibt es die Möglichkeit, ein neues VI (NEW VI) zu erstellen. Unter OPEN EXISTING (oder über FILE unter OPEN) können Sie auch bereits vorhandene VIs öffnen. Es öffnet sich nun ein VI, welches aus zwei Fenstern besteht:

- ✓ Frontpanel
- ✓ Blockdiagramm

Wenn Sie unter dem Reiter VIEW die Option TILE LEFT AND RIGHT anwählen, teilt sich der Bildschirm. Optional können Sie auch den Shortcut `Strg` + `T` verwenden. Auf der linken Seite sehen Sie dann das Frontpanel und auf der rechten Seite das Blockdiagramm (Abbildung 1.6). Das Frontpanel enthält zusätzlich oben rechts den Connector Pane.

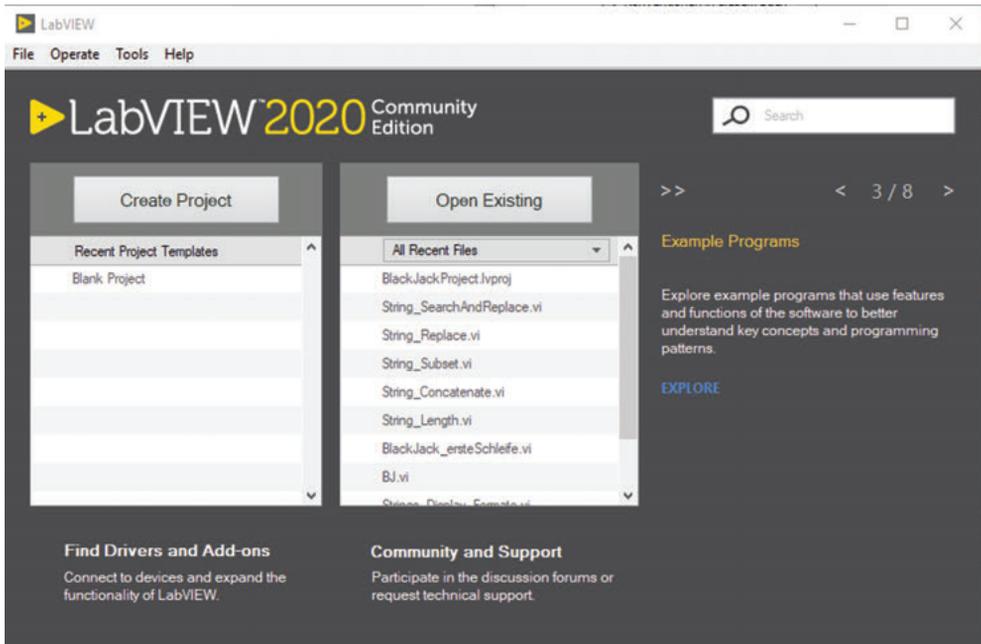


Abbildung 1.5: Startoberfläche in LabVIEW zum Navigieren

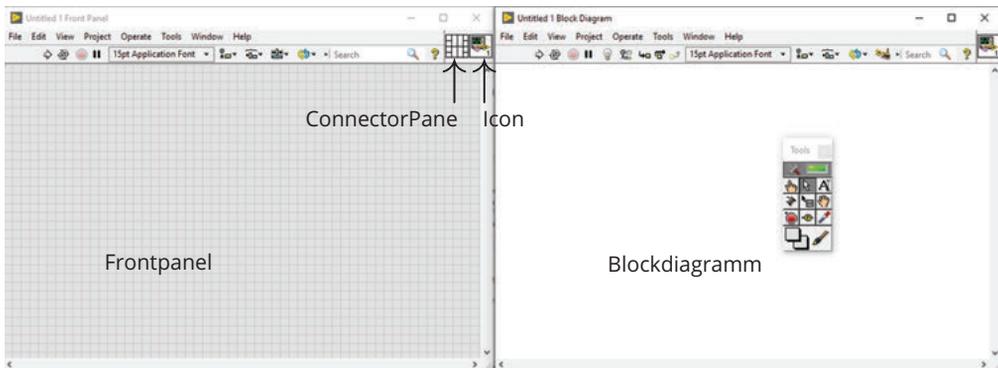


Abbildung 1.6: Startbildschirm nach dem Öffnen eines neuen VIs mit den drei Komponenten: Frontpanel, Blockdiagramm und Connector Pane

Haben Sie ihr Programm fertig erstellt bzw. möchten Sie einen Zwischenstand festhalten, können Sie es unter dem Reiter FILE → SAVE AS... unter einem beliebigen Namen mit der Endung .vi (wird automatisch hinzugefügt) abspeichern.

Frontpanel und dessen Elemente

Auf dem Frontpanel können Sie die gesamte interaktive Bedien- und Anzeigeoberfläche des Programms gestalten. Zur Simulation von Bedienelementen (Controls) steht Ihnen eine große Auswahl an Knöpfen und Schaltern zur Verfügung. Ganz platt gesagt könnten Sie hier den Startknopf für eine Rakete mit Hilfe eines LED-Schalters darstellen. Anzeigeelemente (Indicators) gibt es zum Beispiel in Form von Graphen und Diagrammen. Dies könnte ein Tachometer sein, welches die Geschwindigkeit der Rakete anzeigt. Die jeweiligen Elemente finden Sie auf der Controls Palette (ausführlichere Informationen dazu in Kapitel 4), die Sie mit einem Rechtsklick im Frontpanel aufrufen können. Das gewünschte Element kann dann per Drag-and-Drop auf dem Frontpanel platziert werden.

Falls Sie in der Fülle der Möglichkeiten kein passendes Element finden, steht es Ihnen frei, eigene Kreationen zu gestalten. In der Controls Palette stehen unter DECORATIONS diverse Dekorationselemente zur Verfügung. Auf der Tools Palette (diese wird im Verlauf des Kapitels noch näher erläutert) finden Sie zudem auch noch ein Farbwerkzeug zum Verändern der Farbe. Allerdings birgt diese schnelle und einfache Gestaltungsmöglichkeit die Gefahr, dass man es beim Design ein wenig übertreibt. Gerade das Farbwerkzeug lädt gerne dazu ein, zu tief in den Farbtopf zu greifen.



Deshalb möchten wir an dieser Stelle ein Gebot unseres Professors auch Ihnen mit auf den Weg geben: Ein guter Programmierer kennt neben schwarz und weiß nur drei Farben – grau, hellgrau und dunkelgrau. Natürlich können und dürfen Sie auch andere Farben verwenden, zum Beispiel die Signalfarben rot und grün, welche die Aufmerksamkeit des Anwenders auf eine bestimmte Region lenken. Dennoch sollte man sich immer fragen, ob eine farbliche Gestaltung zwingend notwendig ist, und ob sie den Anwender nicht möglicherweise nur verwirren würde.

Außerdem sollten Sie bei komplexen Programmen überlegen, einen Graphikdesigner zu Rate zu ziehen. Diese haben einen anderen Blickwinkel auf das Frontpanel und können Ihnen zu einem übersichtlicheren Programm verhelfen.

Sollten Ihnen auch diese Gestaltungsmöglichkeiten immer noch nicht ausreichen, können Sie in einem Graphikprogramm ihrer Wahl eigene Kreationen erstellen und diese dann in LabVIEW über sogenannte Custom Controls einbinden. Wie das funktioniert, ist in Kapitel 4 beschrieben, da wir für das Frontpanel des Black Jack-Programms ein Bild von einem Croupier eingefügt haben. Ebenso haben wir uns ein eigenes Kartendeck als Custom Control erstellt und in LabVIEW eingebunden.

Menüleiste vom Frontpanel

Die Menüleiste des Frontpanels finden Sie in 1.7 abgebildet, deren einzelne Schaltflächen nun in den folgenden Unterkapiteln erläutert werden.

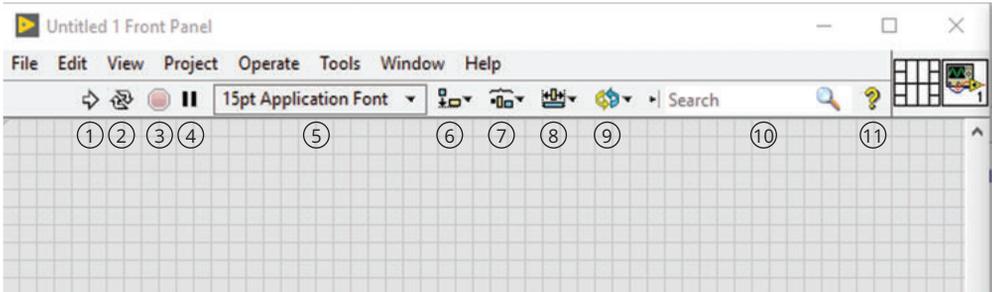


Abbildung 1.7: Symbolleiste des Frontpanels,

- ① RUN; ② RUN CONTINUOUSLY;
- ③ ABORT EXECUTION; ④ PAUSE;
- ⑤ TEXT SETTINGS; ⑥ ALIGN OBJECTS;
- ⑦ DISTRIBUTE OBJECTS; ⑧ RESIZE OBJECTS;
- ⑨ REORDER; ⑩ SEARCH;
- ⑪ CONTEXT HELP WINDOW

Starten, Beenden, Pausieren

Mit dem RUN Button ① können Sie das Programm starten. Während der Programmierung können Sie außerdem anhand der Darstellung des Pfeils erkennen, ob das Programm lauffähig ist oder Fehler behoben werden müssen. Das Programm ist lauffähig, wenn der Pfeil im ganzen dargestellt ist (Abbildung 1.8a). Gibt es noch Fehler im Programm, wird der Pfeil zerbrochen dargestellt (Abbildung 1.8b) und das Programm ist somit auch nicht ausführbar. Wird das Programm schlussendlich gestartet, erscheint der Pfeil während der Laufzeit wie in Abbildung 1.8c dargestellt.



Abbildung 1.8: Verschiedene Zustände des RUN Buttons

Möchten Sie das Programm kontinuierlich laufen lassen, können Sie den ② Button RUN CONTINUOUSLY verwenden. Das Programm wird dann nach Beendigung direkt wieder gestartet. Diesen Vorgang müssen Sie unter erneuter Verwendung des ② RUN CONTINUOUSLY wieder stoppen.

Die Verwendung des ABORT EXECUTION BUTTON, um ein Programm zu beenden, bedeutet ein echtes Abbrechen des Programms. Dies kann unter Umständen zu Schwierigkeiten führen und sollte nur verwendet werden, wenn es unbedingt notwendig ist. Im Normalfall sollte das Programm aus sich selbst heraus beendet werden.

Möchten Sie eine Pause im Programm einlegen, wird ④ PAUSE verwendet. Ein erneuter Klick auf den PAUSE-Button startet das Programm dann wieder.

Beschriftungen

Möchten Sie die Beschriftungen manipulieren, geht dies über ⑤ TEXT SETTINGS. Dort haben Sie folgende Möglichkeiten zur Änderung:

- ✓ die Schriftart
- ✓ die Schriftgröße
- ✓ die Schriftfarbe
- ✓ und vieles mehr.

Ausrichten der Elemente

Das Frontpanel ist die virtuelle Bedienoberfläche des späteren Anwenders und ist somit ein Aushängeschild des Programmierers. Deshalb sollten die Elemente auf dem Frontpanel gut zueinander ausgerichtet und nicht wie Kraut und Rüben verteilt sein. Eine ordentliche Ausrichtung der Elemente gelingt Ihnen mit:

- ⑥ ALIGN OBJECTS (Objekte ausrichten)
- ⑦ DISTRIBUTE OBJECTS (Objekte verteilen)
- ⑧ RESIZE OBJECTS (Größe ändern)
- ⑨ REORDER (Umsortieren)

Die dazugehörigen Untermenüs, welche jeweils rechts über den kleinen schwarzen Pfeil aufgerufen werden, sind in Abbildung 1.9 zu sehen. Bei ALIGN OBJECTS (1.9a) gibt es unter anderem die Möglichkeit, die Objekte mittig, links oder rechts zueinander zu positionieren. Mit DISTRIBUTE OBJECTS (1.9b) werden die Abstände der Objekte zueinander bestimmt: ob der gleiche Abstand zwischen allen Objekten herrschen soll oder diese möglichst nah beieinander liegen sollen. Außerdem gibt es verschiedene Möglichkeiten zur horizontalen oder vertikalen Positionierung.

Die maximale/minimale Länge oder Breite der Elemente kann über RESIZE OBJECTS festgelegt werden. Ganz unten auf der entsprechenden Palette gibt es noch die Option, die Länge und Breite in einem separaten Dialogfenster einzustellen.

Wichtig ist auch die Funktion REORDER: Hier können Sie Ebenen festlegen, in denen Ihre Elemente angeordnet sind. Beispielsweise wäre das zuletzt gesetzte Element in der höchsten Ebene und die darunterliegenden Elemente nicht mehr anklickbar. Dies ist gerade bei der Verwendung von DECORATIONS sehr wichtig, da diese meist erst am Ende ins Programm eingefügt werden und die vorher eingefügten Elemente dann nicht mehr anklickbar sind, obwohl sie sichtbar sind. Ebenso haben Sie bei REORDER die Möglichkeit, mehrere Objekte miteinander zu gruppieren (GROUP/UNGROUP) und zu fixieren (LOCK/UNLOCK).

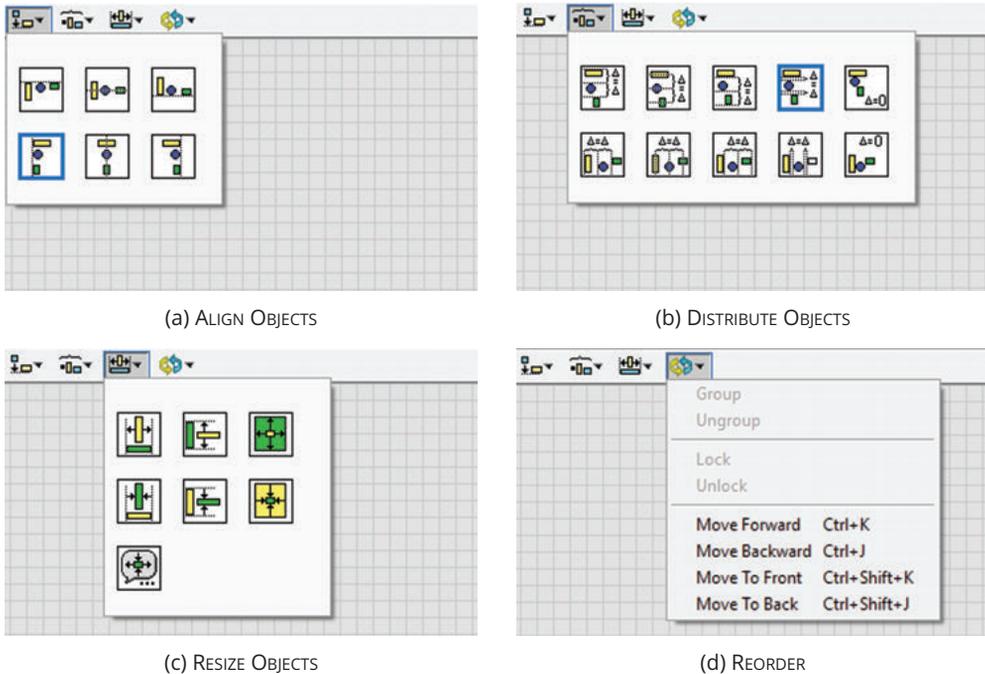


Abbildung 1.9: Unterpalletten zur Positionierung auf dem Frontpanel

Hilfen

In der ⑩ Suchmaske SEARCH, die es übrigens auch im Blockdiagramm gibt, können Sie über den Namen die verschiedenen Anzeige- und Bedienelemente, Strukturen und Funktionen suchen, die Sie zur Erstellung Ihres Programms benötigen. Dazu müssen Sie natürlich schon eine gewisse Ahnung haben, wonach Sie genau suchen, sonst kommen Sie an dieser Stelle nicht weiter.

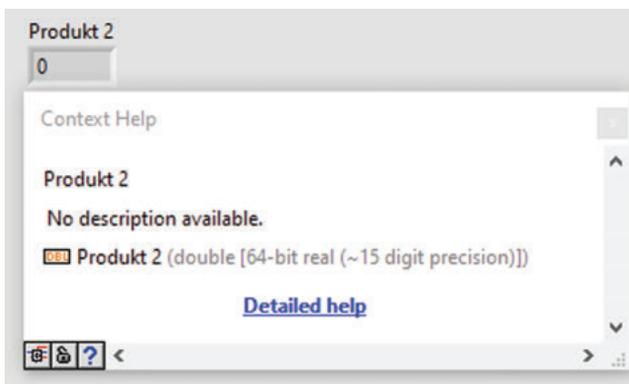


Abbildung 1.10: Kontexthilfe von »Produkt 2«

Das gelbe Fragezeichen  ist die sogenannte Kontexthilfe. Wenn Sie die Kontexthilfe aufrufen, indem Sie das gelbe Fragezeichen anklicken, öffnet sich ein kleines Fenster. Wenn Sie nun mit dem Mauszeiger über einem Element im Frontpanel verharren, werden in dem kleinen Fenster (Abbildung 1.10) weitere Informationen zu dem entsprechenden Element angezeigt (zum Beispiel, um welchen Datentyp es sich handelt – in diesem Fall `DOUBLE` – mehr dazu aber in Kapitel 3).

Blockdiagramm

Wenn Sie nun im Frontpanel ein paar schöne Schalter oder Drehknöpfe platziert haben, finden Sie dazu korrespondierend im Blockdiagramm die passenden Symbole. Die Symbole können auch als Terminals dargestellt werden. So benötigen sie weniger Platz im Blockdiagramm (Abbildung 1.11). Die Umstellung auf die Terminals erfolgt über `VIEW AS ICON` im Kontextmenü des Symbols und ist in diesem Buch Standard. Das Kontextmenü können Sie über einen Rechtsklick auf das Symbol aufrufen. Im weiteren Verlaufe des Buches wird das Kontextmenü noch häufig auftauchen, da es eine wichtige Rolle bei der Programmierung in LabVIEW innehat.



Auch wenn sich Kontexthilfe und Kontextmenü sehr ähnlich anhören, sind dies grundverschiedene Dinge:

- ✓ Die Kontexthilfe zeigt Ihnen in einem extra Fenster Hilfen zu Funktionen usw. an.
- ✓ Über das Kontextmenü können Sie Funktionen, VIs und Elemente des Frontpanels beeinflussen.

Diese Symbole oder Terminals fungieren dann als Datenquelle (Controls) oder Datensenke (Indicators). Ein Bedienelement wie der LED-Schalter der Rakete ist zum Beispiel eine Datenquelle und besitzt rechts einen Ausgang. Das Tachometer, welches die Geschwindigkeit der Rakete anzeigt, ist eine Datensenke und besitzt links einen Eingang. Die Ein- und Ausgänge werden jeweils durch kleine Pfeile symbolisiert.



(a) Symbol



(b) Terminal

Abbildung 1.11: Verschiedene Darstellungsweisen im Blockdiagramm

Die Anzeige- und Bedienelemente können nun durch verschiedenste Funktionen und Strukturen zu einem logischen Programm verknüpft werden. Das Blockdiagramm dient nicht nur der Illustration der Abläufe, sondern ist auch das auszuführende Programm selbst. Dies funktioniert allerdings nur, wenn die Terminals über sogenannte Wires (Drähte) mit Funktionen und Strukturen verbunden werden.

Denn erst die Wires sorgen dafür, dass die Daten von einem Knoten zum nächsten fließen können. Im wahren Leben funktioniert beispielsweise ein Drucker ja auch erst dann, wenn Sie ihn mit einem Kabel an Ihren PC anschließen (es sei denn, es handelt sich um ein WLAN-Drucker).

Sämtliche Funktionen und Strukturen für das Blockdiagramm finden Sie in der Functions Palette, die Sie – analog zur Controls Palette des Frontpanels – per Rechtsklick im Blockdiagramm aufrufen können. Auch hier werden die einzelnen Elemente per Drag-and-Drop auf dem Blockdiagramm platziert. Wenn Sie nun mit der Maus über die einzelnen Elemente fahren, werden die Anschlusspunkte sichtbar und der Mauszeiger verwandelt sich in eine Drahtrolle, so dass Sie die einzelnen Verdrahtungen nun mit der Maus vornehmen können.

Menüleiste im Blockdiagramm

Die meisten Symbole der Menüleiste im Blockdiagramm in Abbildung 1.12 kennen Sie schon vom Frontpanel. Deshalb werden wir hier nur die »neuen« Funktionen erläutern. Die Funktion RESIZE OBJECTS gibt es übrigens NICHT für das Blockdiagramm, weil die Elemente dort eine fest definierte Größe haben und dementsprechend nicht verändert werden können.

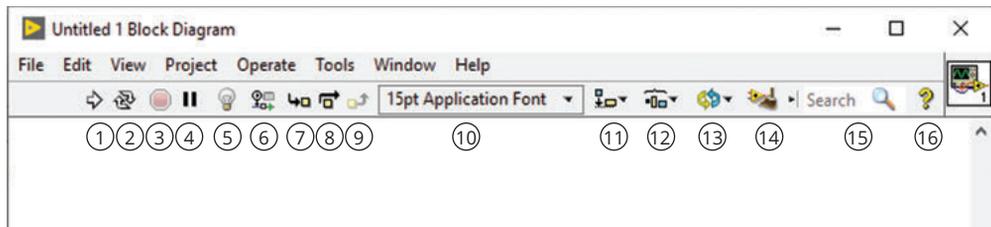


Abbildung 1.12: Symbolleiste des Blockdiagramms;

- ① RUN; ② RUN CONTINUOUSLY;
- ③ ABORT EXECUTION; ④ PAUSE;
- ⑤ HIGHLIGHT EXECUTION; ⑥ RETAIN WIRE VALUES;
- ⑦ STEP IN; ⑧ STEP OVER;
- ⑨ STEP OUT; ⑩ TEXT SETTINGS;
- ⑪ ALIGN OBJECTS; ⑫ DISTRIBUTE OBJECTS;
- ⑬ REORDER; ⑭ CLEAN_UP DIAGRAM;
- ⑮ SEARCH; ⑯ CONTEXT HELP WINDOW

Debugging

Die Symbole ⑤ bis ⑨ (HIGHLIGHT EXECUTION, RETAIN WIRE VALUES, STEP IN, STEP OVER, STEP OUT) verwendet man im Rahmen der Fehlersuche (Debugging). Sie werden in Kapitel 8 eingehend erläutert.

Clean_Up Diagram

Mit  CLEAN_UP DIAGRAM können Sie das gesamte Blockdiagramm automatisch aufräumen lassen. In der Regel werden die vorhandenen Elemente und Wires durch die Anwendung von CLEAN_UP DIAGRAM auf kleinerem Raum komprimiert und gleichmäßig zueinander angeordnet. Am Quellcode wird dabei nichts geändert, lediglich die Anordnung der Symbole auf dem Blockdiagramm. Möchten Sie nur einen Teilbereich des Blockdiagramms aufräumen lassen, müssen Sie diesen zunächst markieren und dann erst den Besen anklicken.



Die Funktion CLEAN_UP DIAGRAM ist häufig sehr sinnvoll. Sie kann unter Umständen jedoch auch mehr Schaden als Nutzen bringen, da durch die Funktion teilweise auch totaler Murks produziert wird, weshalb das Programm noch unübersichtlicher werden kann. Deshalb sollten Sie sich nach deren Anwendung vergewissern, dass das Blockdiagramm tatsächlich aufgeräumt und somit übersichtlicher ist als vorher!

Falls das leider nicht der Fall sein sollte, können Sie das Aufräumen über EDIT → UNDO oder  +  rückgängig machen.

Connector Pane

Das Connector Pane (Anschlussfeld) ist der dritte Bestandteil der Entwicklungsumgebung eines VIs. Sie finden es im Frontpanel oben rechts neben der Menüleiste. Das Symbol ist ein Quadrat, welches in mehrere Kästchen unterteilt ist (zu sehen in Abbildung 1.6). Über das Connector Pane werden Ein- und Ausgänge definiert, wenn das Programm als SubVI (Unterprogramm) in einem Hauptprogramm verwendet werden soll. Klickt man mit der rechten Maustaste auf die Schaltfläche neben dem Connector Pane, kann man unter EDIT ICON ein eigenes Symbolbild für das neu erstellte SubVI kreieren, damit es im Blockdiagramm schneller identifiziert werden kann. Mehr zum Thema SubVIs und die Verwendung des Connector Panes erfahren Sie in Kapitel 6.

Werkzeuge und Paletten

Wenn Sie mit LabVIEW programmieren, stehen Ihnen drei verschiedene Paletten dafür zur Verfügung:

- ✓ die Tools Palette (Werkzeugpalette)
- ✓ die Controls Palette (Palette für Anzeige- und Bedienelemente)
- ✓ die Functions Palette (Funktionspalette)

Die Controls Palette dient zur Gestaltung des Frontpanels und die Functions Palette dient zur Gestaltung des Blockdiagramms. Die Tools Palette kann sowohl im Frontpanel als auch im Blockdiagramm verwendet werden. Sie enthält mehrere Werkzeuge, um verschiedene Ausführungsmodi einzustellen.

Tools Palette

Die Tools Palette ist in Abbildung 1.13 dargestellt und Sie können diese vom Frontpanel und Blockdiagramm aus unter dem Reiter VIEW → TOOLS PALETTE dauerhaft aufrufen. Ruft man die Tools Palette über die Taste Shift und einen Rechtsklick auf, wird sie nur kurzzeitig eingeblendet. Bei der Erstellung und Bearbeitung eines Programms gibt es verschiedene Ausführungsmodi, welche aus der Werkzeugpalette ausgewählt werden können. Die einzelnen Werkzeuge sind in Tabelle 1.1 aufgelistet und kurz erläutert.



Abbildung 1.13: Tools Palette

Der aktuelle Modus wird immer durch eine Änderung des Aussehens des Mauszeigers dargestellt, wenn Sie mit der Maus über die jeweiligen Elemente fahren. Dafür muss die LED der AUTOMATIC TOOL SELECTION jedoch grün leuchten . Im Blockdiagramm ist dies zum Beispiel die Drahtrolle (Modus: CONNECT WIRE ) , wenn Sie die einzelnen Funktionen miteinander verdrahten wollen oder wenn Sie über bereits vorhandenen Wires verharren. Im Frontpanel kann der Mauszeiger die Form einer Hand annehmen (Modus: OPERATE VALUE ) , wenn Sie an einem Thermometer die Temperatur einstellen wollen. LabVIEW wählt die entsprechenden Tools dann situationsbedingt automatisch aus.

Mit dem Modus EDIT TEXT  können Sie freie Textfelder im Frontpanel und Blockdiagramm an beliebiger Stelle erstellen. Das kann notwendig sein, wenn Sie Teile Ihres Programms an Ort und Stelle genauer beschreiben wollen. Außerdem wird es verwendet, wenn Texte und Beschriftungen geändert werden sollen.

Bild	Name	Beschreibung
	AUTOMATIC TOOL SELECTION	Abhängig von der Position des Cursors wird eines der ersten vier Werkzeuge automatisch ausgewählt, wenn AUTOMATIC TOOL SELECTION aktiviert (grün) ist.
	OPERATE VALUE	Zum Ändern von Werten, z. B. bei einem Drehschalter oder zum Ein- und Ausschalten einer LED-Anzeige
	POSITION/SIZE/SELECT	Die Elemente können hiermit an eine andere Stelle gezogen, in der Größe geändert oder ausgewählt werden.
	EDIT TEXT	Zur Bearbeitung von Beschriftungen oder zum Erstellen von Textfeldern
	CONNECT WIRE	Zum Erstellen von Drähten im Blockdiagramm
	OBJECT SHORTCUT MENU	Zum Öffnen des Objektkontextmenüs
	SCROLL WINDOW	Zum Scrollen ohne Bildlaufleiste
	SET/CLEAR BREAKPOINT	Zum Setzen oder Löschen von Breakpoints
	PROBE DATA	Zur Überprüfung von Zwischenwerten zur Laufzeit des Programms
	GET COLOR	Zur Ermittlung von Farbwerten
	SET COLOR	Zur Anwendung von Farben auf ein Objekt

Tabelle 1.1: Werkzeuge der Tools Palette



Das Erstellen freier Textfelder geht noch einfacher, wenn Sie einen Doppelklick an beliebiger Stelle im Frontpanel/Blockdiagramm ausführen. Dann erscheint dort ein blinkender Cursor und Sie können direkt den gewünschten Text schreiben. Das funktioniert allerdings nur, wenn AUTOMATIC TOOL SELECTION auf der Tools Palette grün leuchtet.

Der Ausführungsmodus OBJECT SHORTCUT MENU  öffnet das Kontextmenü des jeweiligen Elementes. Allerdings gibt es auch hier einen schnelleren Weg, um das Kontextmenü zu öffnen: einfach mit einem Rechtsklick das gewünschte Element anklicken!

Mit SCROLL WINDOW  können Sie im Frontpanel/Blockdiagramm durch das Programm scrollen, ohne dafür die Bildlaufleiste benutzen zu müssen. Dies kann bei größeren Programmen hilfreich sein.

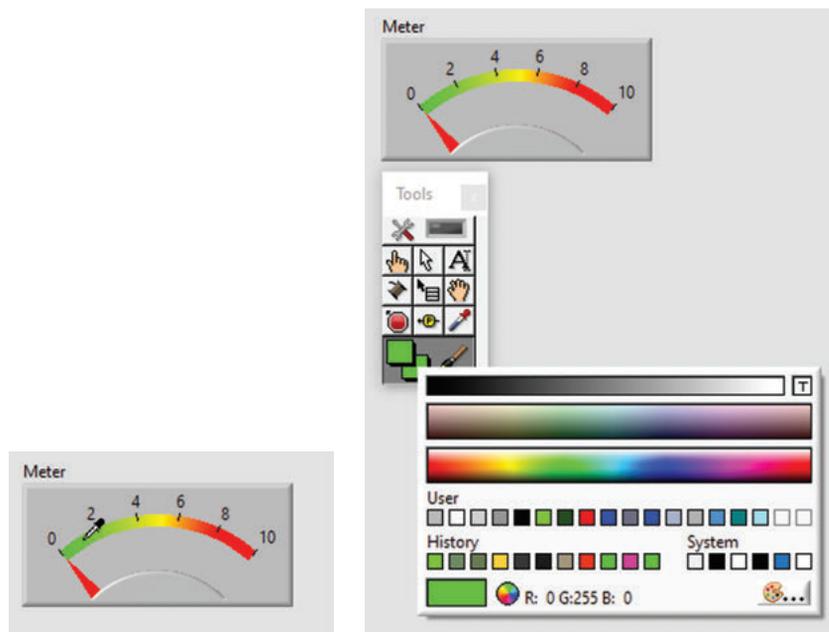
Die Werkzeuge SET/CLEAR BREAKPOINT  und PROBE DATA  werden zur Fehlersuche (Debugging) verwendet und im entsprechenden Kapitel 9 genauer erläutert.

Zu guter Letzt wollen wir Ihnen die beiden Farbwerkzeuge GET COLOR  und SET COLOR  erläutern. Wenn Ihnen im Frontpanel (im Blockdiagramm geht es nicht) eine Farbe unterkommt, die Ihnen so gut gefällt, dass Sie wissen möchten aus welchen Rot-, Grün- und Blauanteilen sie sich zusammensetzt, können Sie das Tool GET COLOR verwenden (Abbildung 1.14a). Zunächst müssen Sie die Farbpipette in der Tools Palette anklicken, anschließend auf die gewünschte Farbe im Frontpanel. Die beiden Vierecke ganz unten auf der Tools Palette (neben dem Pinsel) färben sich dann in der gewünschten Farbe. Wenn Sie nun die gefärbten Vierecke anklicken, öffnet sich ein Fenster, in welchem Ihnen die Farbzusammensetzung angezeigt wird (Abbildung 1.14b). Möchten Sie nun ein anderes Objekt mit dieser Farbe einfärben, klicken Sie auf SET COLOR und anschließend auf das gewünschte Objekt. SET COLOR ist wiederum auch im Blockdiagramm beispielsweise bei Schleifen anwendbar. Mehr zur Verwendung von Farben in LabVIEW und deren Zusammensetzung erfahren Sie in Kapitel 4.

Controls Palette

Aus der Controls Palette können Sie Anzeige- und Bedienelemente entnehmen, um das Frontpanel zu gestalten. Aufgerufen wird die Palette über einen Rechtsklick auf das Frontpanel. Sie enthält unter anderem:

- ✓ Numerische Anzeige- und Bedienelemente (z. B. Schalter, Drehknöpfe, Thermometer, Tacho)
- ✓ boolesche Anzeige- und Bedienelemente (z. B. Schalter, Knöpfe, LED-Anzeige)
- ✓ Strings zur Textverarbeitung im Programm
- ✓ Datencontainer (Arrays und Cluster)
- ✓ Tabellen, Diagramme und Graphen
- ✓ Ring & Enum (Aufzählungselemente)
- ✓ Decorations



(a) Verwendung von GET COLOR

(b) Nach Anwendung von GET COLOR

Abbildung 1.14: Anwendung von GET COLOR

In Abbildung 1.15 sehen Sie die Controls Palette des Frontpanels. Würden Sie beispielsweise mit dem Mauszeiger auf die Kachel `BOOLEAN` fahren, öffnet sich eine Unterpalette, von der Sie verschiedene boolesche Elemente entnehmen und im Frontpanel platzieren können. Das ist auch bei den anderen Kacheln der Fall. Die Unterpaletten und ihre Elemente werden in den jeweiligen Kapiteln eingehend erläutert. Auch hier steht eine Suchmaske zur Verfügung, wenn Sie ein bestimmtes Element nicht finden können, aber dessen Namen kennen. Manchmal ist es hilfreich, die Palette über die Pinnadel oben links zu fixieren, denn dann verschwindet die Palette nicht immer wieder, wenn man mit dem Mauszeiger woanders hinfährt.

Decorations

Eine spezielle Unterpalette der Controls Palette ist die `DECORATIONS` Palette in Abbildung 1.16. Diese kann verwendet werden, um Elemente auf dem Frontpanel sinnvoll zu strukturieren und so die Aufmerksamkeit des Anwenders gezielt zu lenken. Programmtechnisch haben Sie keinen Einfluss.

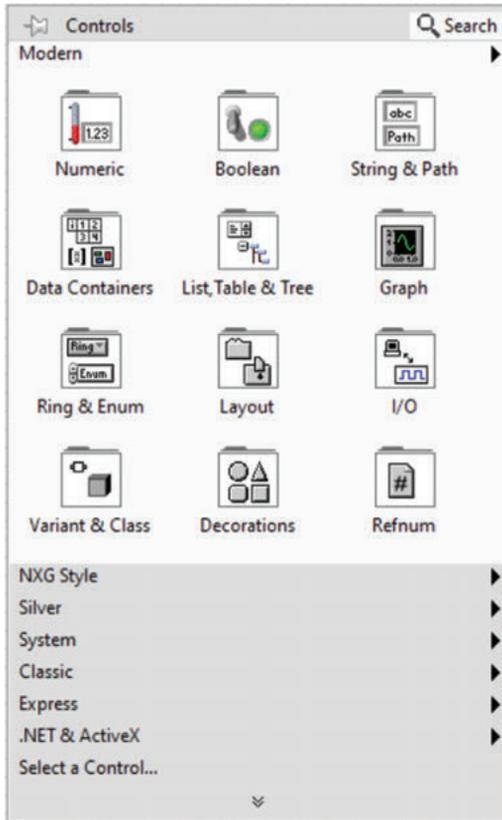


Abbildung 1.15: Controls Palette auf dem Frontpanel



Die Reihenfolge der Elemente ist gerade bei Decorations sehr wichtig, weil es sein kann, dass, wenn Sie eine THICK LOWERED BOX über bestehende Elemente setzen, die darunter liegenden Elemente zwar sichtbar, aber nicht anwählbar sind.

In so einem Fall müssen Sie die Box markieren und MOVE TO BACK auswählen.

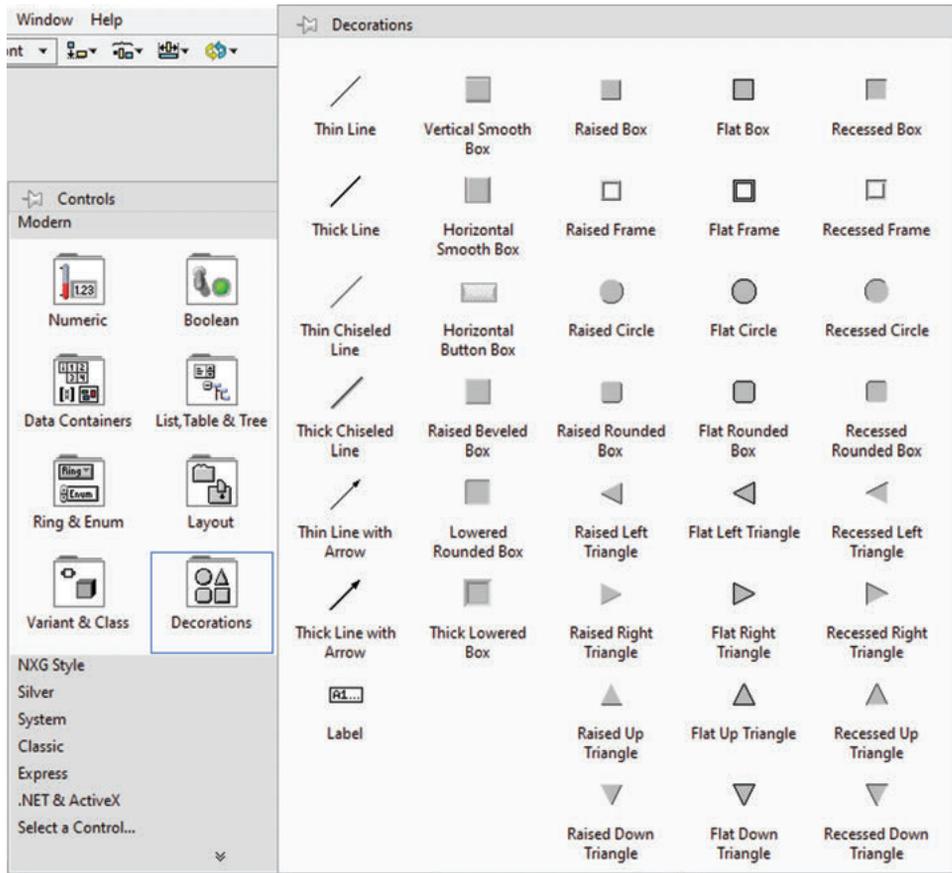


Abbildung 1.16: DECORATIONS Unterpalette auf dem Frontpanel

Functions Palette

Der Functions Palette des Blockdiagramms können Sie Funktionen und Strukturen entnehmen, die Sie über Wires mit den Anzeige- und Bedienelementen aus dem Frontpanel verbinden können. Aufzurufen ist die Palette mit einem Rechtsklick in das Blockdiagramm. Durch die Verwendung von Funktionen und Strukturen entsteht die eigentliche Programmlogik – der Quellcode. Sie enthält unter anderem:

- ✓ Strukturen (Schleifen, Fallunterscheidung, Sequenz, Ereignis, Formula Node)
- ✓ Array und Cluster Funktionen
- ✓ numerische Funktionen (z. B. Additions- und Subtraktionszeichen)
- ✓ boolesche Funktionen (z. B. AND und OR)
- ✓ String Funktionen

Die Ansicht der Functions Palette sehen Sie in Abbildung 1.17. Auch hier gilt das gleiche wie für die Controls Palette: Es gibt Unterpalletten, die in den nachfolgenden Kapiteln vorgestellt werden. Ebenso steht auch hier eine Suchmaske und eine Pinnadel zur Fixierung zur Verfügung.

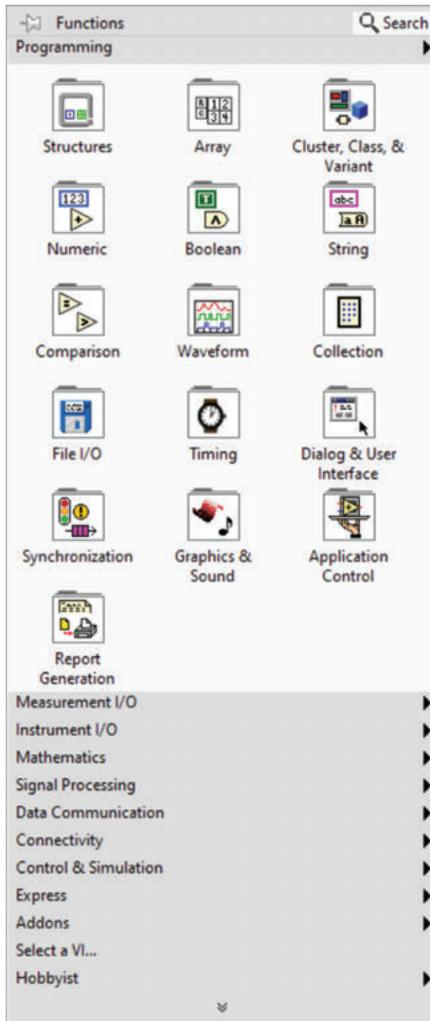


Abbildung 1.17: Functions Palette auf dem Blockdiagramm

Kontextmenü

Das Kontextmenü erreichen Sie, wie bereits erwähnt, über einen Rechtsklick auf ein beliebiges Objekt im Frontpanel oder Blockdiagramm. Jedes Anzeige- und Bedienelement im Frontpanel und jede Funktion im Blockdiagramm besitzt ein eigenes Kontextmenü.

Die Kontextmenüs enthalten jeweils zahlreiche Möglichkeiten, mit denen Einstellungen an dem zugehörigen Element vorgenommen werden können. Manche Elemente besitzen sogar mehrere Kontextmenüs, je nachdem, auf welchen Teil des Elements geklickt wird. Ein Beispiel dafür sind Diagramme und Graphen, mit denen Signalverläufe dargestellt werden (mehr dazu erfahren Sie in Kapitel 9).

Am sinnvollsten ist es, wenn Sie sich im Rahmen der Programmentwicklung jedes Kontextmenü bei Bedarf anschauen und ausprobieren, welche Möglichkeiten es für jedes einzelne Objekt bietet. Dennoch möchten wir im Folgenden die hilfreichsten Punkte der Kontextmenüs aufzeigen und kurz erläutern. Weitere Punkte der Kontextmenüs werden im Verlauf des Buches im Zusammenhang mit ihren jeweiligen Elementen erläutert.

Kontextmenü im Frontpanel

Im Kontextmenü von Bedien- und Anzeigeelementen im Frontpanel gibt es unter anderem die folgenden Möglichkeiten anzuwählen:

- ✓ VISIBLE ITEMS
- ✓ FIND TERMINAL
- ✓ CREATE
- ✓ PROPERTIES

In Abbildung 1.18 ist exemplarisch das Kontextmenü des numerischen Tank-Elementes zu sehen, in dem Sie beispielsweise den Füllstand simulieren können.

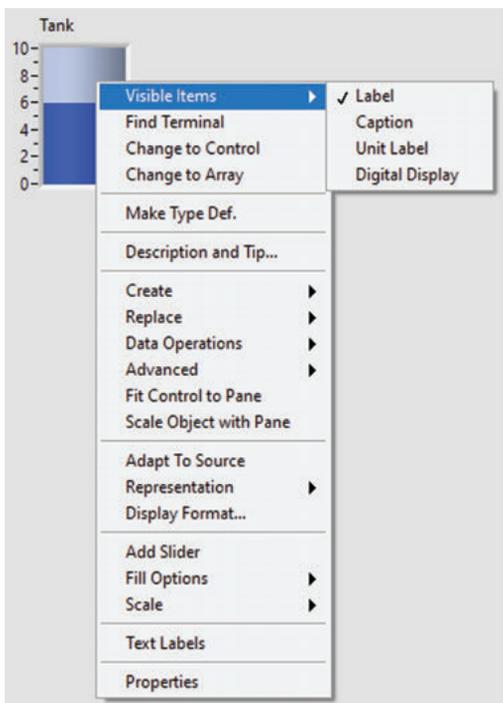


Abbildung 1.18: Kontextmenü des Frontpanels

Visible Items

Bei VISIBLE ITEMS öffnet sich nochmals ein Untermenü, in dem LABEL und CAPTION angewählt werden können. Man sollte dazu wissen, dass CAPTIONS die Beschriftungen im Frontpanel und LABELS die Beschriftungen im Blockdiagramm sind. Dementsprechend könnte das Tank-Element auf dem Frontpanel beispielsweise »Kesselfüllhöhe« heißen, während es auf dem Blockdiagramm »Control1« heißt. Das kann während der Programmierung oder Fehlersuche unter Umständen zu Verwirrung führen. Die unterschiedlichen Beschriftungsmöglichkeiten können jedoch auch hilfreich sein, wenn im Frontpanel ein längerer Name notwendig wird und dieser aber im Blockdiagramm zu sperrig wäre.

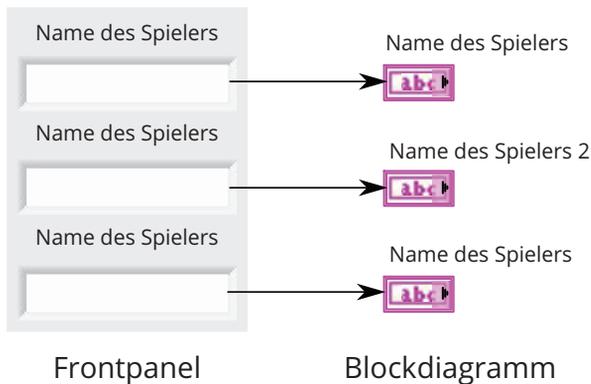


Abbildung 1.19: Beispiel für den Unterschied zwischen CAPTION und LABEL



Die Beschriftung im LABEL sollte immer eindeutig sein und nicht doppelt vorkommen, damit die Elemente im Blockdiagramm nicht verwechselt werden. LabVIEW bietet leider die Möglichkeit, dass zwei Elemente anhand des Labels gleich benannt werden können, obwohl diese nichts miteinander zu tun haben! Zwar ist es so, dass LabVIEW automatisch mehrere gleiche Elemente nummeriert, aber diese Ziffern können vom Programmierer wieder ganz leicht entfernt werden. In Abbildung 1.19 ist dies exemplarisch dargestellt. Im Frontpanel sind drei Eingabefelder zu sehen, die alle den Namen »Name des Spielers« haben, was dort durchaus erwünscht sein kann. Spätestens im Blockdiagramm sollten die Beschriftungen jedoch eindeutig zugeordnet werden können. In diesem Fall sollte man im letzten Element wieder die 3 in der Beschriftung einfügen, damit keine Fehler bei der Programmierung entstehen.

Wenn Sie möchten, dass auf dem Frontpanel der gleiche Name für zwei Elemente vorhanden ist, dann sollten Sie sich über das Kontextmenü die CAPTION anzeigen lassen und diese gleich nennen. Das LABEL auf dem Blockdiagramm sollte jedoch eindeutig benannt sein, um spätere Verwirrung vorzubeugen!

Es kann zum Beispiel sinnig sein, dass auf dem Frontpanel »Spieler Nummer« steht, auf dem Blockdiagramm aber nur die Abkürzung »SN« verwendet wird. Ebenso ist es möglich, dass Sie auf dem Frontpanel Elemente mit dem gleichen Namen verwenden möchten (z. B. »Name«), diese jedoch im Blockdiagramm unterscheidbar sein müssen, weil es sonst zu Programmierfehlern kommen kann. Dann können Sie mit Hilfe der CAPTION die Elemente im Frontpanel gleich benennen und mit dem LABEL im Blockdiagramm unterschiedliche Namen vergeben.

Meist gibt es abhängig vom Datentyp unter VISIBLE ITEMS noch unterschiedliche Möglichkeiten, Dinge ein- oder auszublenden. Dazu gehören zum Beispiel:

- ✓ Einheiten bei einem numerischen Element
- ✓ Index Display beim Datencontainer Array
- ✓ Legende bei Diagrammen und Graphen

Ein Blick in diese Kategorie lohnt sich also, wenn Sie einen Datentyp neu verwenden.

Find Terminal

Manchmal kann es vorkommen, dass Sie Ihr Programm nicht direkt von Anfang an übersichtlich gestaltet haben, oder dass Sie bei gleichen Captions auf dem Frontpanel nicht direkt wissen, welches Terminal im Blockdiagramm dazugehört. Dann können Sie über den Befehl FIND TERMINAL direkt zu dem entsprechenden Element auf dem Blockdiagramm gelangen, ohne lange suchen zu müssen. Ein Doppelklick auf das Element hat übrigens denselben Effekt.

Create

Der Reiter CREATE bietet die Möglichkeit lokale Variablen, Property Nodes oder Invoke Nodes zu erstellen. Mit Property Nodes kann man zur Laufzeit des Programms Elemente im Frontpanel verändern oder auf deren Werte zurückgreifen. Im Black Jack-Programm steht dem Spieler zum Beispiel kurzzeitig das Bedienelement des Kartendecks nicht mehr zur Verfügung, weil der Croupier als nächster am Zug ist. Über den Property Node kann eingestellt werden, dass das Kartendeck für den Spieler in diesem Moment ausgegraut erscheint. Ausführlichere Informationen zu Property/Invoke Nodes und Lokalen Variablen finden Sie in Kapitel 4.

Properties

Unter den PROPERTIES können für alle Anzeige- und Bedienelemente die folgenden Eigenschaften eingestellt werden:

- ✓ APPEARANCE
- ✓ DOCUMENTATION
- ✓ DATABINDING
- ✓ KEY NAVIGATION

Unter APPEARANCE kann das Erscheinungsbild des jeweiligen Elementes verändert werden. Sie können beispielsweise dessen Größe, Position und Farbe entsprechend Ihren eigenen Wünschen anpassen. Die das Erscheinungsbild betreffenden Eigenschaften eines Elementes können unterschiedlich sein – je nach Art und Funktion des jeweiligen Elementes. Zum Beispiel könnten Sie an einem Thermometer die Farbe ändern, was bei einem einfachen Feld zur Zahleneingabe jedoch nicht von Bedeutung und dementsprechend nicht auswählbar wäre. Am besten schaut man sich bei jedem Element an, welche Möglichkeiten für dessen optische Veränderung zur Verfügung stehen.

Allen Elementen gemein ist jedoch, dass Sie unter APPEARANCE auch Einstellungen zu LABEL und CAPTION vornehmen können. So ist es möglich, diese dort sichtbar/unsichtbar zu machen und deren Beschriftung einzugeben bzw. zu verändern.

Ebenso gehört zum Erscheinungsbild der sogenannte ENABLED STATE (Bedienbarkeit) eines Elements. Sie haben dabei die Möglichkeit zwischen den Einstellungen

- ✓ ENABLED (aktiviert) – bedienbar
- ✓ DISABLED (deaktiviert) – nicht bedienbar
- ✓ DISABLED AND GRAYED (deaktiviert und ausgegraut) – nicht bedienbar und ausgegraut

zu wählen.



Abbildung 1.20: Unterschiede zwischen den ENABLED STATES: ENABLED, DISABLED und DISABLED & GRAYED

In Abbildung 1.20 ist augenscheinlich kein Unterschied zwischen ENABLED und DISABLED auszumachen. Das DISABLED-Element ist jedoch für den Anwender weder bedien- noch veränderbar. Optisch ist dies allerdings bei DISABLED nicht zu erkennen, so dass es sinnvoller ist, DISABLED & GRAYED zu verwenden. Das Element wird hier für den Anwender ausgegraut, so dass er erkennen kann, dass es nicht mehr zur Verfügung steht. Im BLACKJACK.VI soll zum Beispiel der Button zum erneuten Ziehen einer Karte für den Spieler ausgegraut werden, wenn der Croupier am Zug ist.



Leider ist uns bisher noch keine Anwendung bekannt, in der die Verwendung von DISABLED allein sinnvoller wäre; es sei denn, der Anwender des Programms soll geärgert werden!

Unter dem Reiter DOCUMENTATION kann man eine kurze Beschreibung (DESCRIPTION) und einen sogenannten TIP STRIP zu dem jeweiligen Element eingeben. Die DESCRIPTION kann über die Kontexthilfe (gelbes Fragezeichen in der Menüleiste) und einen Klick auf das entsprechende Element abgerufen werden (siehe Abbildung 1.21).

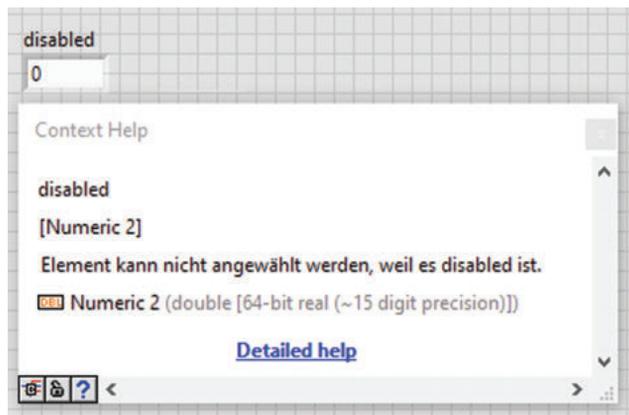


Abbildung 1.21: Kontexthilfe zu einem Frontpanel-Element mit Eintragung unter DESCRIPTION

Der TIP STRIP wird zur Laufzeit des Programms angezeigt, wenn der Mauszeiger einen kurzen Moment über dem Element (egal an welcher Stelle) verharrt (siehe Abbildung 1.22). Zur Erstellungszeit wird der TIP STRIP zwar auch angezeigt, allerdings muss der Mauszeiger dafür vorher die Form einer Hand angenommen haben. Das erreicht man, wenn man das Element bedienen möchte, z. B. die Veränderung der Temperatur beim Thermometer oder beim Ein- und Ausschalten einer LED-Anzeige. In diesem Fall ist es also nicht egal, an welcher Stelle der Mauszeiger über dem Element verharrt!

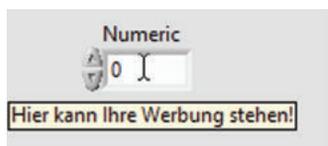


Abbildung 1.22: Beispiel für einen TIP STRIP ZUR Laufzeit

Die Tabs DATABINDING und KEY NAVIGATION sind sehr speziell und müssen nur selten verwendet werden. Deshalb möchten wir sie hier der Vollständigkeit halber aufführen, jedoch nicht weiter beschreiben und auf die eigene Seite von LabVIEW <http://zone.ni.com/> verweisen.

Kontextmenü im Blockdiagramm

In Abbildung 1.23 ist exemplarisch das Kontextmenü der numerischen Funktion Multiplikation dargestellt. Unter HELP können Sie sich eine genaue Beschreibung der Funktion aus der LabVIEW-Hilfe (Abbildung 1.24) anzeigen lassen.

Über CREATE erhalten Sie die Möglichkeit, eine passende Konstante (zum Beispiel die 17 und 21 beim Black Jack), ein Control (Bedienelement, z. B. Schalter) oder einen Indicator (Anzeigeelement, z. B. Tachometer) direkt zu erstellen. Dann müssen Sie diese nicht wieder extra über die Paletten aufrufen und es wird auch direkt der passende Datentyp mit vorgegeben.

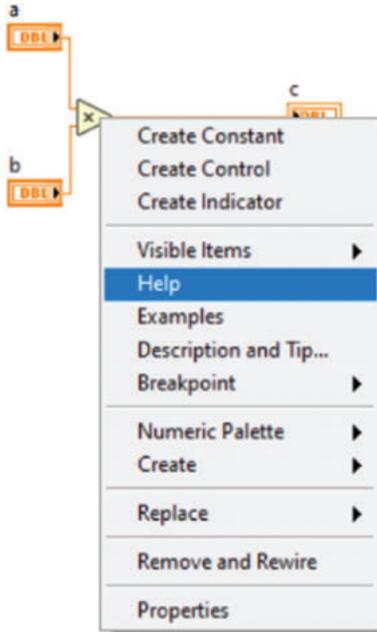


Abbildung 1.23: Kontextmenü des Blockdiagramms

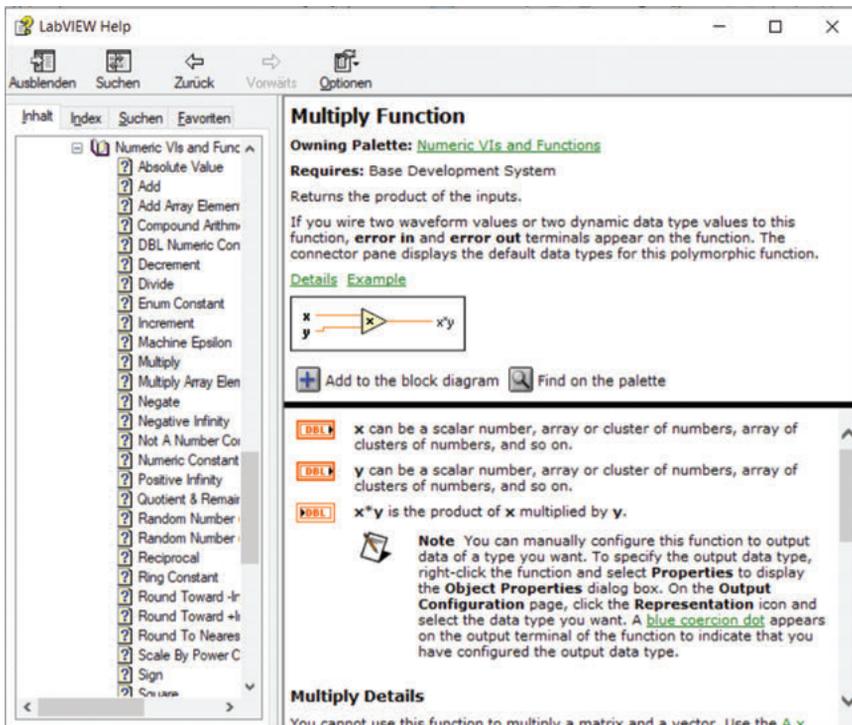


Abbildung 1.24: Hilfe zur Multiplikationsfunktion

Falls Sie eine Funktion ersetzen müssen, können Sie dafür REPLACE verwenden. Hiermit kann eine Funktion einfach durch eine andere ähnliche ersetzt werden. Auf diese Weise können Sie Zeit sparen, weil Sie nicht neu verdrahten müssen. Wenn Sie beispielsweise in der Booleschen Algebra unterwegs sind und zwei boolesche Elemente mit einer UND-Funktion verknüpft haben und merken, dass dies doch nicht die richtige Funktion ist, dann bietet REPLACE die Möglichkeit, sie etwa durch eine ODER-Funktion zu ersetzen.



Mit der ebenso nützlichen Funktion REMOVE AND REWIRE können Sie die angeklickte Funktion löschen und LabVIEW versucht möglichst sinnig die Drähte neu zu verbinden. Dies kann jedoch zu Fehlern führen und sollte von Ihnen immer kontrolliert werden!

Drähte im Blockdiagramm

In Abbildung 1.25 ist das Kontextmenü eines Wires zu sehen.

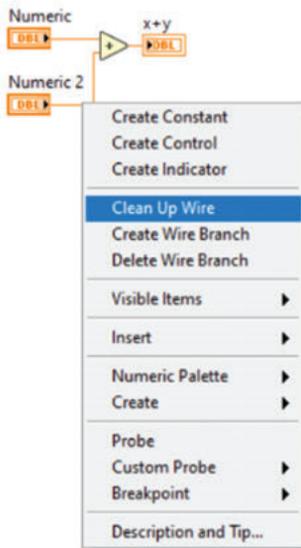


Abbildung 1.25: Kontextmenü der Drähte

Während der Verdrahtung kann es dazu kommen, dass die Wires nicht auf Anhieb gerade gezogen wurden und das Programm dadurch unübersichtlich wird. Mit CLEAN UP WIRE im Kontextmenü der Drähte können diese bereinigt werden, so dass sie hinterher ansehnlicher und übersichtlicher sind. Aber auch hier sollten Sie beachten: LabVIEW ist ein Programm und es kann möglicherweise nach dem Clean Up unübersichtlicher sein als vorher. Deshalb bitte immer überprüfen, ob die Anwendung dieser Funktion ihren Zweck erfüllt hat.



Falls LabVIEW mit CLEAN UP WIRE mehr Schaden als Nutzen angerichtet hat, können Sie dies mit EDIT → UNDO oder **Strg** + **Z** wieder rückgängig machen.

Wenn Sie während der Programmierung merken, dass beispielsweise von Ihrem Wert noch eine -1 (Funktion DECREMENT) abgezogen werden muss, können Sie diese ganz bequem mit der Option INSERT des Kontextmenüs einfügen. Über INSERT hat man die Möglichkeit, schnell auf die Functions Palette des Blockdiagramms zuzugreifen und beispielsweise in diesem Fall die Funktion DECREMENT auszuwählen. Die ursprüngliche Verdrahtung wird durch INSERT beibehalten und die gewünschte -1 komfortabel integriert.

Möchten Sie ein Element mit mehreren Funktionen verdrahten, können Sie mit CREATE WIRE BRANCH Verzweigungspunkte an den Drähten erstellen, so dass Sie problemlos mehrere Funktionen an dasselbe Objekt anschließen können.

Einstellung von Programmeigenschaften

Für ein Programm (VI) können in LabVIEW verschiedene Eigenschaften (Properties) eingestellt werden. Über einen Rechtsklick auf die Schaltfläche neben dem Connector Pane können Sie dazu die VI PROPERTIES aufrufen. Alternativ verwenden Sie **Strg** + **I**. Dort finden Sie mehrere Unterpunkte, die in Abbildung 1.26 zu sehen sind.

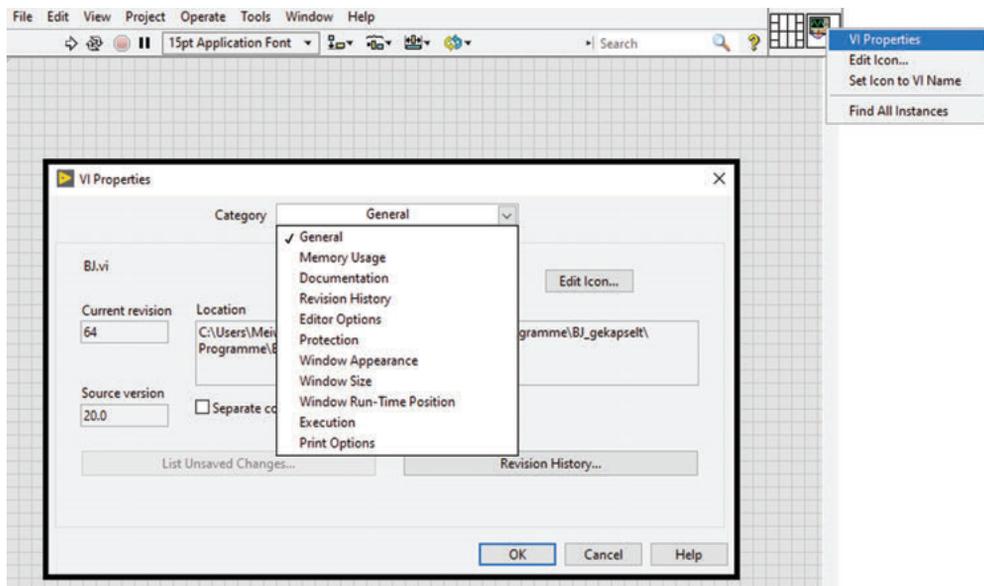


Abbildung 1.26: VI PROPERTIES in LabVIEW

General

In dem Unterpunkt GENERAL gibt es nochmals die Option EDIT ICON, um ein eigenes Symbolbild für dieses VI erstellen zu können. Wie weiter oben schon erwähnt worden ist, kann dieses VI nun als SubVI in einem Hauptprogramm verwendet werden. Das von Ihnen erstellte Icon wird dann im Blockdiagramm des Hauptprogramms angezeigt. Das SubVI kann

anhand des Icons schnell im Blockdiagramm identifiziert werden und dadurch dessen Übersichtlichkeit und Lesbarkeit deutlich verbessern. Mehr zum Thema SubVIs finden Sie in Kapitel 6.

Außerdem können Sie unter GENERAL auch den Speicherort Ihres VIs sehen und wann diese erstellt wurde, die sogenannte REVISION HISTORY (wird gleich noch erläutert).

Memory Usage

Unter MEMORY USAGE können Sie sich ansehen, wie hoch der Speicherbedarf für das Programm ist.

Dabei wird, wie Sie in Abbildung 1.27 sehen können, zwischen verschiedenen Elementen unterschieden und am Ende der Gesamtspeicherbedarf angegeben.

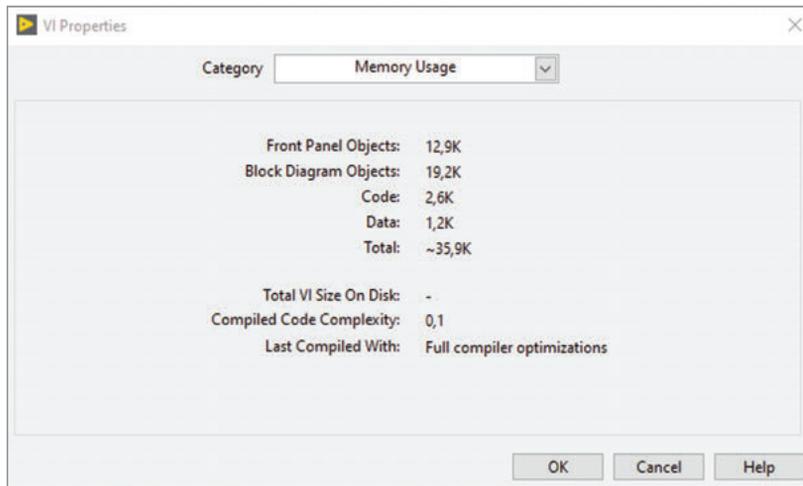


Abbildung 1.27: MEMORY USAGE in den VI PROPERTIES

Documentation

In dem Tab DOCUMENTATION (Abbildung 1.28) sollte das Programm gut dokumentiert werden. Diese Beschreibung des Programms wird dann in der Kontexthilfe (gelbes Fragezeichen in der Menüleiste) angezeigt, wenn Sie mit dem Mauszeiger über der Schaltfläche neben dem Connector Pane oder über einem eingebundenen SubVI verharren.



So wenig verlockend es ist, eine Dokumentation zu verfassen: Bitte tun Sie sich und Anderen den Gefallen und investieren Sie die Zeit für eine gute Dokumentation des Programms. Sie (und auch Andere) werden sich dafür bedanken, wenn Sie nach mehreren Monaten versuchen nachzuvollziehen, was das Programm tut.

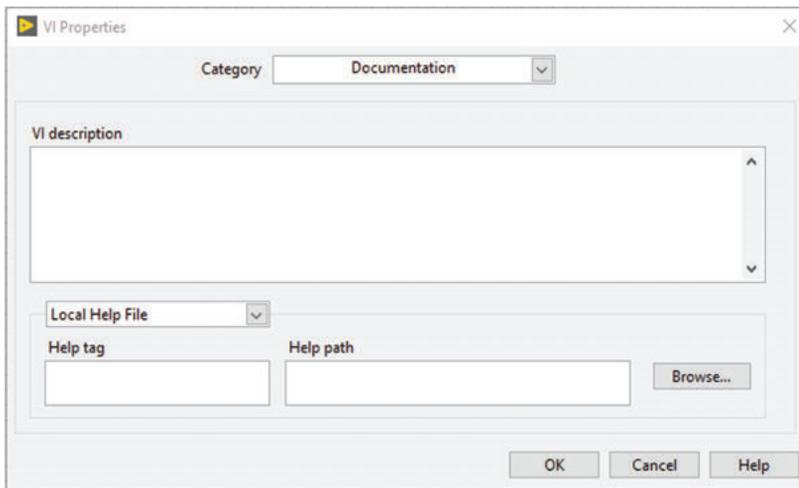


Abbildung 1.28: DOCUMENTATION in den VI PROPERTIES



Bitte die DOCUMENTATION aus den VI PROPERTIES nicht mit der DOCUMENTATION aus den PROPERTIES der Controls/Indicator verwechseln, welche über deren Kontextmenü im Frontpanel aufgerufen werden!

Revision History

In der REVISION HISTORY können Sie einstellen, ob zum Beispiel bei jeder Speicherung des VIs ein Eintrag in die History vorgenommen werden soll. Dies kann sinnvoll sein, wenn mehrere Mitarbeiter an einem Projekt arbeiten oder auch wenn Sie längere Zeit an einem Projekt arbeiten und verschiedene Stände rekonstruieren möchten.

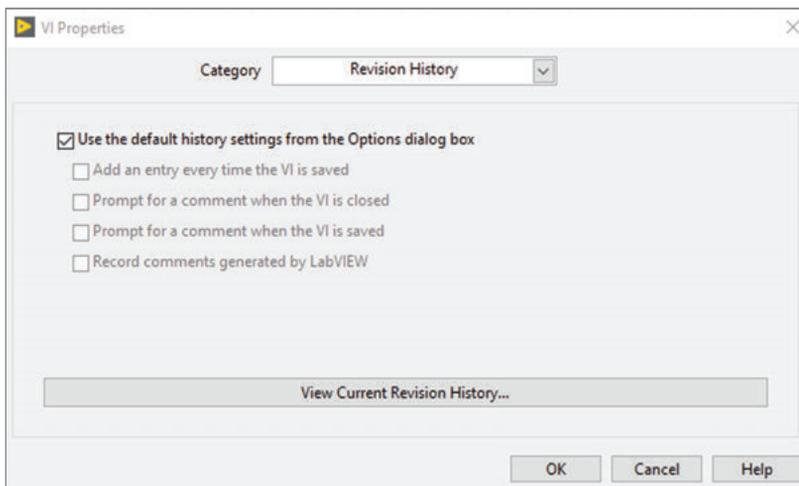


Abbildung 1.29: REVISION HISTORY in den VI PROPERTIES

In diesem Fall geben Sie zu jeder Speicherung einen kurzen Text ein, der die Veränderungen zur letzten Version beschreibt. Dies geschieht in dem Dialogfenster (Abbildung 1.30), welches sich durch einen Klick auf **VIEW CURRENT REVISION HISTORY...** oder im Programm selber über die Tastenkombination **Strg + Y** öffnet. In dem Feld **COMMENT** können Sie Ihren Kommentar eingeben, welcher anschließend unten in der **HISTORY** angezeigt wird. Außerdem sehen Sie unter **CURRENT REVISION** einen Zähler, der die vorhandenen Versionen hochzählt (in der Abbildung steht er bei 1).

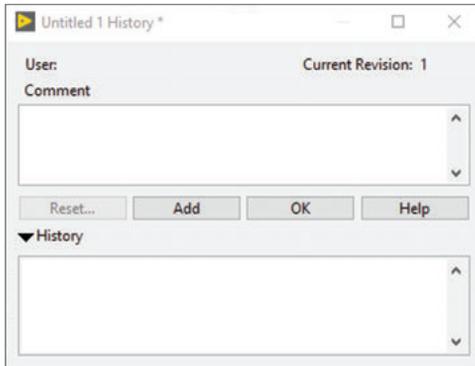


Abbildung 1.30: REVISION HISTORY-Dialogfenster



Wenn Sie mit mehreren Personen an einem Quellcode arbeiten, kann es sinnvoll sein, sich auf eine bestimmte Art der Dokumentation zu einigen. Zum Beispiel, dass immer der Name des Autors und das Datum vorne angeschrieben wird, so dass direkt ersichtlich wird, wer welche Änderungen vorgenommen hat und zu welchem Zeitpunkt.



Üblicherweise wird zur Versionskontrolle mit zusätzlichen Programmen wie SVN (Subversion) oder GitHub gearbeitet.

Editor Options

In der Unterkategorie **EDITOR OPTIONS** können Sie die Größe des Frontpanel-Rasters (**ALIGNMENT GRID SIZE**) einstellen, wenn Sie eine andere Kästchengröße als vorgegeben verwenden möchten.

Außerdem haben Sie hier die Möglichkeit, den Style für die Controls und Indicators festzulegen, wenn diese vom Blockdiagramm aus über die Option **CREATE** des Kontextmenüs erzeugt werden. Diese Einstellung bezieht sich auf die verschiedenen Designmöglichkeiten der Elemente im Frontpanel, welche ebenso über die Controls Palette ausgewählt werden können. Die verschiedenen Styles stehen unterhalb der Kachelansicht zur Auswahl.

In diesem Buch wird MODERN STYLE verwendet. Falls Ihnen die Designs der anderen Styles besser gefallen, steht es Ihnen natürlich frei, einen anderen Stil zu wählen. An der Funktion ändern die unterschiedlichen Stile nichts.

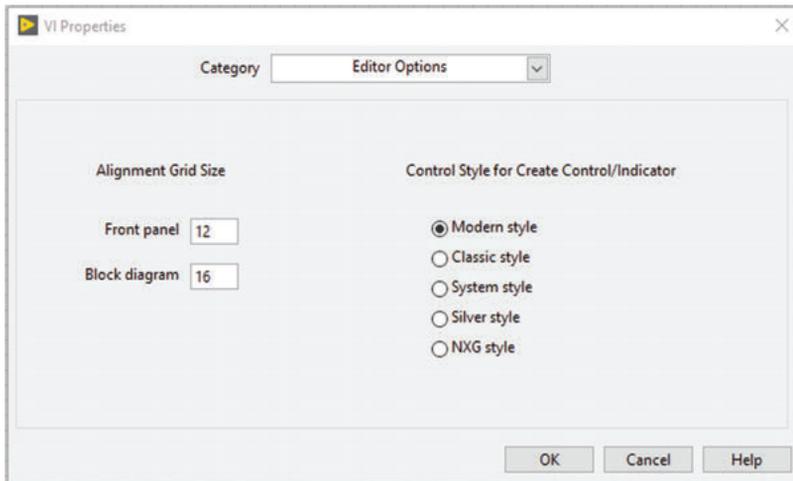


Abbildung 1.31: EDITOR OPTIONS in VI PROPERTIES

Protection

Wenn Sie das Programm an einen Kunden herausgeben, kann es sinnvoll sein, dieses durch ein Passwort zu schützen, damit Ihr Code nicht kopiert werden kann. Dafür wählen Sie die Option PASSWORD-PROTECTED aus Abbildung 1.32.

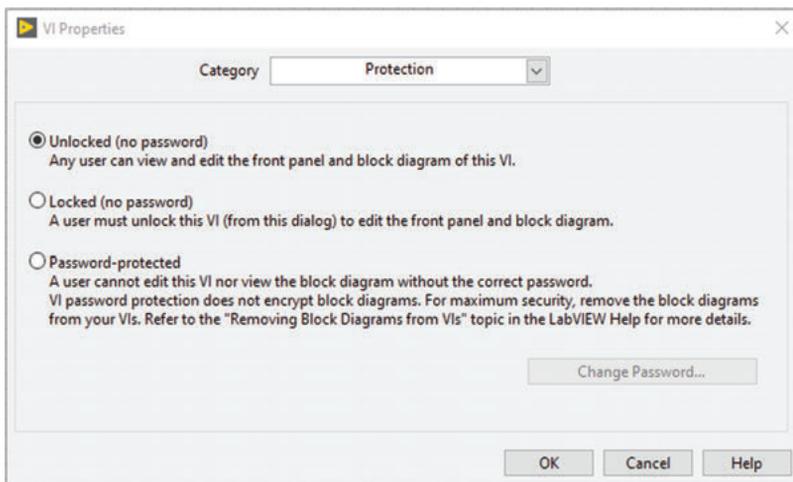


Abbildung 1.32: PROTECTION in den VI PROPERTIES

Die Eingabe des Passwortes kann übrigens nicht über die gängige EDIT → UNDO-Funktion wieder gelöscht werden. Und wie man es auch von anderen Passwort-Vergaben kennt: Zur Verifizierung muss das neue Passwort zweimal eingegeben werden.

Wählen Sie statt des Passwortschutzes die Option LOCKED (NO PASSWORD) für Ihr Programm aus, ist es für den Anwender nicht möglich, das Blockdiagramm zu editieren.

Wenn Sie den Standardmodus UNLOCKED (NO PASSWORD) wählen, kann der Anwender das komplette Programm anschauen, ausführen und bearbeiten.

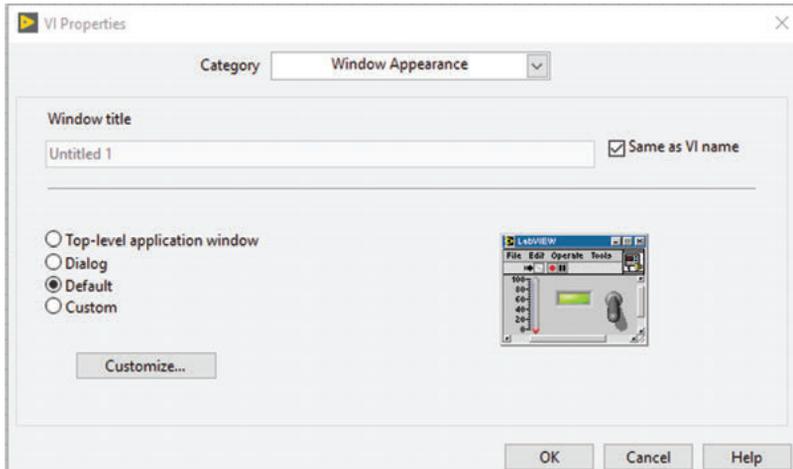


Abbildung 1.33: ENTER PASSWORD-Dialogfenster

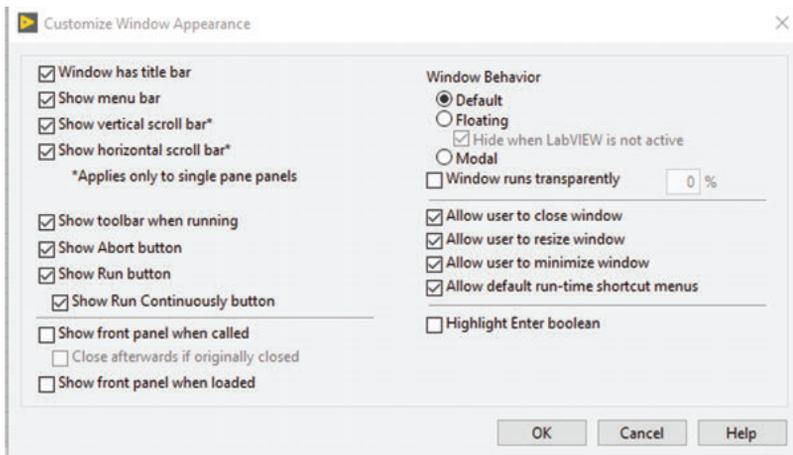
Window Appearance

Im Reiter WINDOW APPEARANCE (Abbildung 1.34a) können Sie im Startdialog auswählen, in welcher Form das Frontpanel-Fenster angezeigt werden soll (hier ist DEFAULT ausgewählt). Außerdem kann der Name, welcher zur Laufzeit des VIs angezeigt werden soll, in der Menüleiste WINDOW TITLE verändert werden, so dass dieser nicht dem eigentlichen VI-Namen entspricht. In der Regel will man zur Erstellungszeit den Überblick behalten, so dass mehrere VIs jeweils ihren eigenen eindeutigen Namen bekommen. Stellt man das Programm hinterher jedoch einem Kunden zur Verfügung, kann es sinnvoller sein, dass die VIs zur Laufzeit einen einheitlichen Namen angezeigt bekommen.

Unter CUSTOMIZE können Sie Parameter zur Laufzeit des Programms hinterlegen (Abbildung 1.34b). Zum Beispiel kann es sinnvoll sein, dem Benutzer zur Laufzeit nicht zu erlauben, das Programm zu beenden (ALLOW USER TO CLOSE WINDOW) oder dass die Menüleiste (SHOW MENU BAR) nicht angezeigt wird – dafür müssten die entsprechenden Häkchen in Abbildung 1.34b entfernt werden.



(a) Startdialog



(b) Customize

Abbildung 1.34: WINDOW APPEARANCE in den VI PROPERTIES

Window Size

Mit WINDOW SIZE können Sie die Größe des VI-Fensters festlegen. Wenn Sie die aktuelle Größe zur Laufzeit haben möchten, so verwenden Sie den Button SET TO CURRENT PANEL SIZE. Es kann durchaus sinnvoll sein, dass sich das Fenster an verschiedene Monitorkonfigurationen anpasst, da in der Entwicklung meistens größere Bildschirme verwendet werden als später beispielsweise in der Produktion. Dafür setzen Sie ein Häkchen bei MAINTAIN PROPORTIONS OF WINDOW FOR DIFFERENT MONITOR RESOLUTIONS.

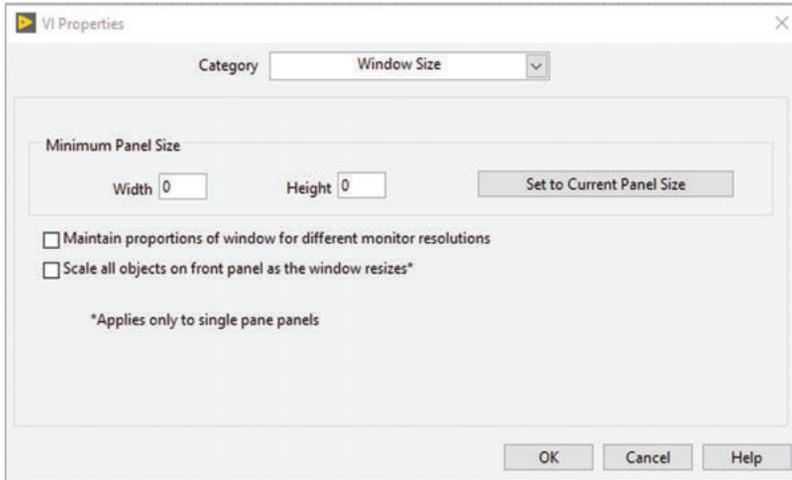


Abbildung 1.35: WINDOW SIZE in den VI PROPERTIES

Window Run-Time Position

Unter WINDOW RUN-TIME POSITION (Abbildung 1.36) können Sie die Position des Programmfensters zur Laufzeit festlegen. In der Checkbox POSITION stehen dafür folgende Optionen zur Auswahl:

- ✓ UNCHANGED
- ✓ CENTERED
- ✓ MAXIMIZED
- ✓ MINIMIZED
- ✓ CUSTOM

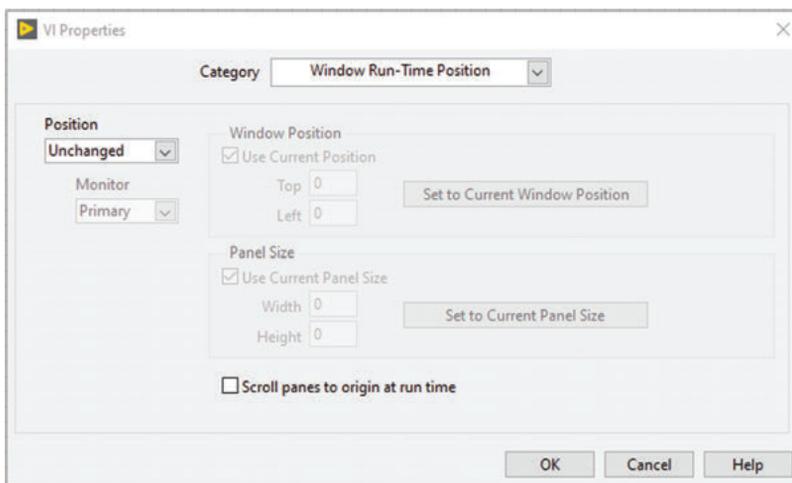


Abbildung 1.36: RUN TIME POSITION in den VI PROPERTIES

Wenn Sie Programme für Kunden schreiben, ist es meist angebracht, das Fenster in der Mitte anzuzeigen zu lassen, damit die Aufmerksamkeit des Bedieners direkt auf dem Programm liegt. Dafür können Sie die Option `CENTERED` auswählen.

Wenn mehrere Programme gleichzeitig nebeneinander liegen sollen, kann unter `CUSTOM` deren genaue Position festgelegt werden.

Execution

Über den Tab `EXECUTION` (Abbildung 1.37) können Eigenschaften, welche die Ausführung eines VIs betreffen, eingestellt werden.

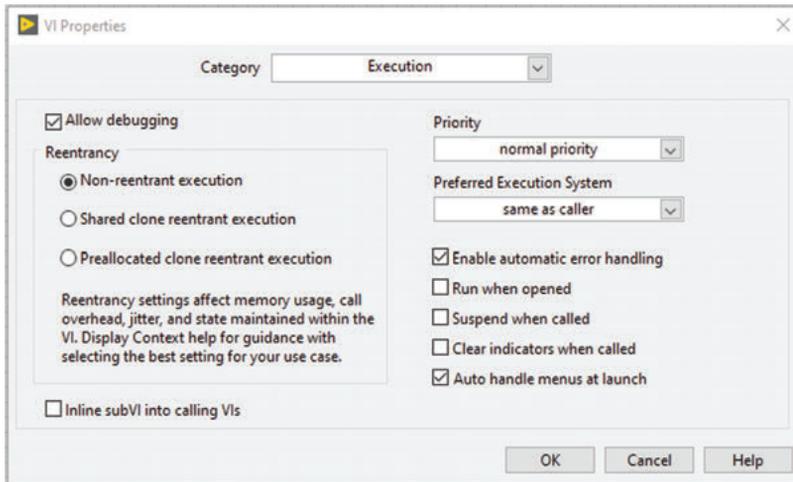


Abbildung 1.37: EXECUTION in den VI PROPERTIES

Wenn bei `ALLOW DEBUGGING` ein Häkchen gesetzt wird, stehen dem Anwender die verschiedenen Optionen zur Fehlersuche (mehr dazu in Kapitel 8) zur Verfügung. Ist `ALLOW DEBUGGING` nicht aktiviert, benötigt das Programm weniger Speicherplatz und läuft zudem auch schneller, ist dann jedoch zur Laufzeit nicht debugbar.

Print Options

Wenn Sie festlegen möchten, wie ein VI ausgedruckt werden soll, können Sie im Tab `PRINT OPTIONS` verschiedene Einstellungen dazu vornehmen (Abbildung 1.38).

Hierbei können Sie folgende Optionen wählen:

- ✓ mit Überschrift drucken (Name, Datum, Seitennummer)
- ✓ mit Umrandung des Frontpanels
- ✓ mit Seitenanpassung des Frontpanels
- ✓ mit Seitenanpassung des Blockdiagramms

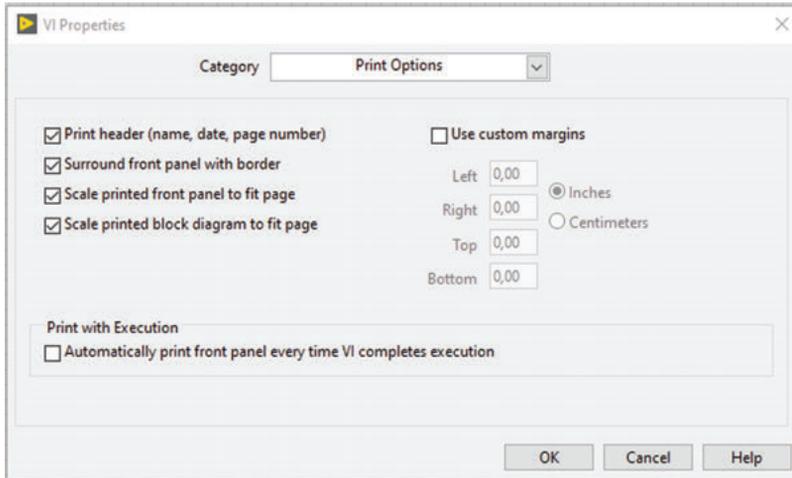


Abbildung 1.38: PRINT OPTIONS in den VI Properties

Wenn das Häkchen bei PRINT WITH EXECUTION gesetzt wird, erzeugt LabVIEW bei jedem Programmdurchlauf einen Ausdruck des Frontpanels.

Regeln für die Programmierung in LabVIEW

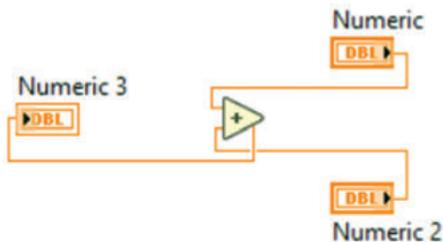
Neben ein paar offensichtlichen Regeln – zum Beispiel, dass der Code so übersichtlich wie möglich gestaltet werden sollte – noch ein paar zusätzliche Regeln, welche Sie bei der Programmierung mit LabVIEW beachten sollten:

- ✓ Das Blockdiagramm sollte aufgrund der Lesbarkeit nicht größer als eine Bildschirmseite und (wenn möglich) auf die Größe des Zielrechners abgestimmt sein, damit es zum Beispiel auch auf einem kleinen Bildschirm in der Produktion oder in einem Labor ordentlich dargestellt werden kann.
- ✓ Wires sollten nicht unter anderen Objekten entlang laufen.
- ✓ Wires sollten nicht von rechts nach links geführt werden.

Übungen zu LabVIEW

1. Was ist das Frontpanel, Blockdiagramm und Connector Pane?
2. Was sind Controls und Indicators?

3. Wo findet man
 - a. eine Drahtrolle?
 - b. ein Thermometer?
 - c. ein Diskettensymbol?
 - d. einen Besen?
 - e. einen Pinsel?
 - f. einen Graphen?
4. Was ist hier falsch?



5. Welche zwei Arten der Dokumentation gibt es?
6. Was ist der Unterschied zwischen Kontexthilfe und Kontextmenü?

Lösungen zu LabVIEW

1. Auf dem Frontpanel wird die Bedienoberfläche des virtuellen Instruments dargestellt. Das Blockdiagramm enthält den dazugehörigen Programmcode. Das Connector Pane definiert die Schnittstelle zu anderen Programmen, weil hier aus dem Programm ein SubVI erstellt werden kann.
2. Controls sind Bedienelemente (zum Beispiel ein Schalter). Sie dienen als Datenquelle und besitzen rechts einen Ausgang. Indicator sind Anzeigeelemente (zum Beispiel ein Thermometer). Sie dienen als Datensenke und haben links einen Eingang.
3. Man findet
 - a. die Drahtrolle zum Auswählen auf der Tools Palette und sie erscheint im Blockdiagramm, wenn man Objekte miteinander verdrahten möchte.
 - b. das Thermometer auf der Controls Palette im Frontpanel. Es ist auf der Kachel der Unterpalette NUMERIC zu sehen.

- c. das Diskettensymbol auf der Functions Palette im Blockdiagramm. Es ist auf der Kachel der Unterpalette FILE I/O zu sehen.
 - d. den Besen in der Menüleiste des Blockdiagramms. Die Funktion heißt CLEAN_UP DIAGRAM und sie bereinigt den Quellcode im Blockdiagramm.
 - e. den Pinsel auf der Tools Palette. Er kann zum Einfärben bestimmter Bereiche im Frontpanel genutzt werden.
 - f. den Graphen auf der Controls Palette im Frontpanel. Er ist auf der Kachel der Unterpalette GRAPH zu sehen.
4. Der Datenfluss führt hier von rechts nach links. Aufgrund der Anschlüsse der einzelnen Elemente ist es eigentlich nahezu unmöglich, so zu programmieren, aber gerade in größeren und unübersichtlichen Programmen kann es doch mal schneller passieren, dass man Wires von rechts nach links zieht.
5. Es gibt die Dokumentation in den PROPERTIES (Kontextmenü) der Controls/Indicators und es gibt die Dokumentation des ganzen Programms in den VI PROPERTIES (Rechtsklick auf die Schaltfläche neben dem Connector Pane).
6. Die Kontexthilfe ist das gelbe Fragezeichen in der Menüleiste vom Frontpanel und Blockdiagramm. Sie gibt eine kurze Auskunft über die Art des Elements, über dem der Mauszeiger gerade verharrt. Dort wird auch die Dokumentation angezeigt, wenn sie in den PROPERTIES für das entsprechende Element eingetragen wurde. Das Kontextmenü wird über einen Rechtsklick auf ein Element aufgerufen. Es bietet für jedes Element eine Vielzahl hilfreicher Optionen zu dessen Bearbeitung an.

