1 Entwicklungsumgebung

Dieses Kapitel dient als Einstieg in die Entwicklungsumgebung. Es werden die grundlegenden Funktionen erklärt, der Umgang mit den Tools von LabVIEW erläutert und erste Projekte erstellt. Mit den Übungsaufgaben am Ende des Kapitels kann das LabVIEW-Handling geübt und das angeeignete Wissen vertieft werden.

1.1 Was ist LabVIEW?

LabVIEW ist eine grafische Programmierumgebung zum Entwickeln von anspruchsvollen Mess-, Prüf-, Steuer- und Regelaufgaben. Dabei wird auf die sogenannte Virtual Instrumentation (virtuelle Instrumentierung) von Messgeräten gesetzt, also reale Messgeräte durch Softwaretools abgebildet und miteinander kombiniert. So können nicht wie bei einem konventionellen Multimeter nur Strom und Spannung gemessen, sondern auch gleich noch ein Oszilloskop, ein Generator oder andere beliebige Messgeräte in einer einzigen Applikation verwendet werden (Bild 1.1).

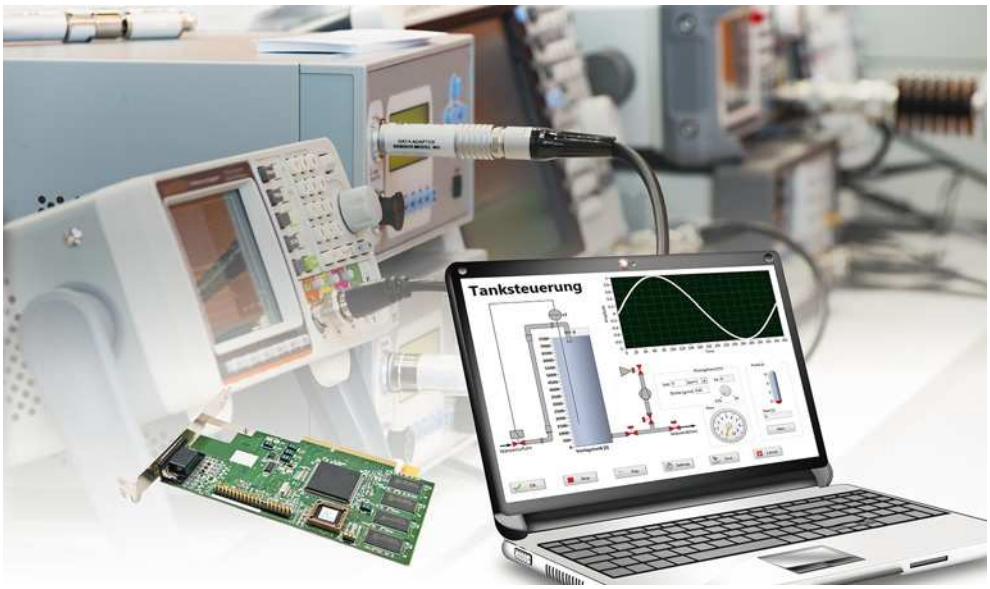


Bild 1.1 Eine einzelne Applikation zur Erfassung, Auswertung und Darstellung

Ein LabVIEW-Programm oder eine einzelne Datei oder Funktion wird deshalb auch als VI (Virtual Instrument) bezeichnet. Die Dateiendung solcher Dateien lautet ebenfalls .vi.

Es findet also mit LabVIEW eine Vernetzung von klassischen Messgeräten mit der IT-Welt statt. Dabei können, sofern es die entsprechende Hardware zulässt, eine sehr große Anzahl von Datenkanälen parallel erfasst und verarbeitet werden. LabVIEW arbeitet mit einer Vielzahl an Hardwareprodukten und beinhaltet eine große Anzahl von Analysefunktionen.

1.2 Grundeinstellungen

Wer die Entwicklungsumgebung nach eigenen Vorlieben konfigurieren möchte, hat unter Tools → Options die Möglichkeit, verschiedenste Grundeinstellungen vorzunehmen. Auf der linken Seite wählt man die Kategorie, während rechts die Optionen zur entsprechenden Kategorie zu sehen sind (Bild 1.2).

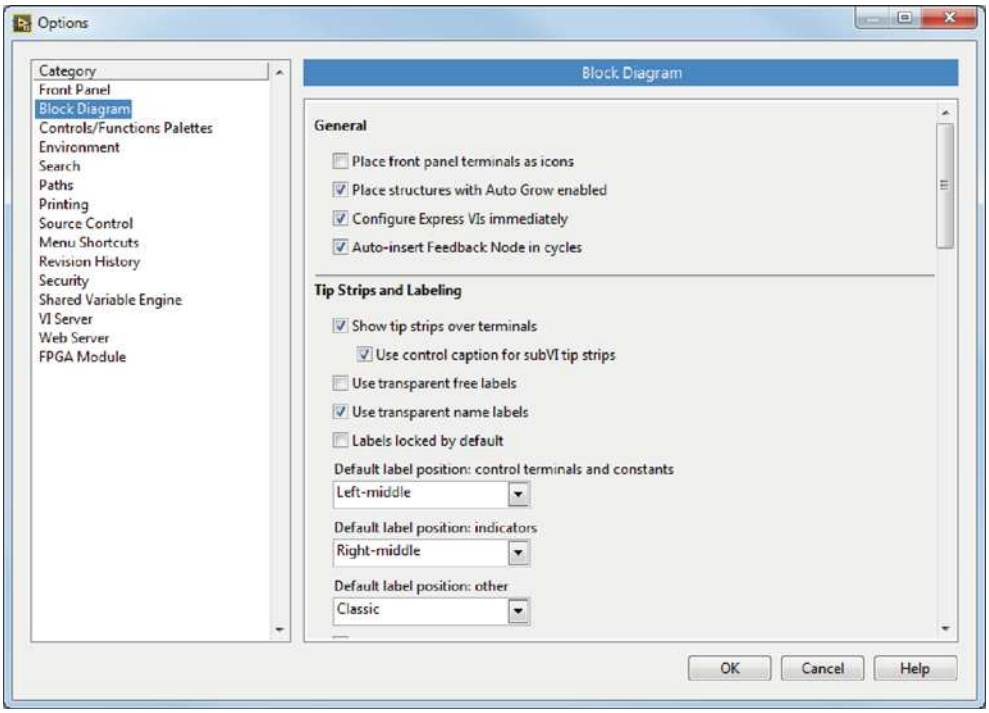


Bild 1.2 Einstellungen des Blockdiagramms

Die meisten Optionen können auf dem Standard belassen werden. Lediglich für das Blockdiagramm bietet es sich an, darauf zu achten, dass die Checkbox Place front panel terminals as icons nicht selektiert ist. Dies spart Platz und erhöht die Übersicht im Blockdiagramm.

1.3 Bestandteile eines VIs

Ein LabVIEW-Programm kann unter Umständen aus nur einem einzigen VI, also einer einzelnen Datei bestehen. Das VI unterscheidet drei wesentliche Hauptbestandteile (Bild 1.3):

- Frontpanel: ist die Bedienoberfläche, enthält Controls (Bedienelemente, zur Eingabe von Daten) und Indicators (Anzeigeelemente, zur Ausgabe von Daten)
- Blockdiagramm: enthält den grafischen Quellcode und damit die Funktion des VIs, enthält Anschlüsse für Controls und Indicators auf dem Frontpanel
- Symbol/Anschlussblock: repräsentiert das VI und ermöglicht die Verwendung als SubVI, zeigt die Belegung der Ein- und Ausgänge des VIs an

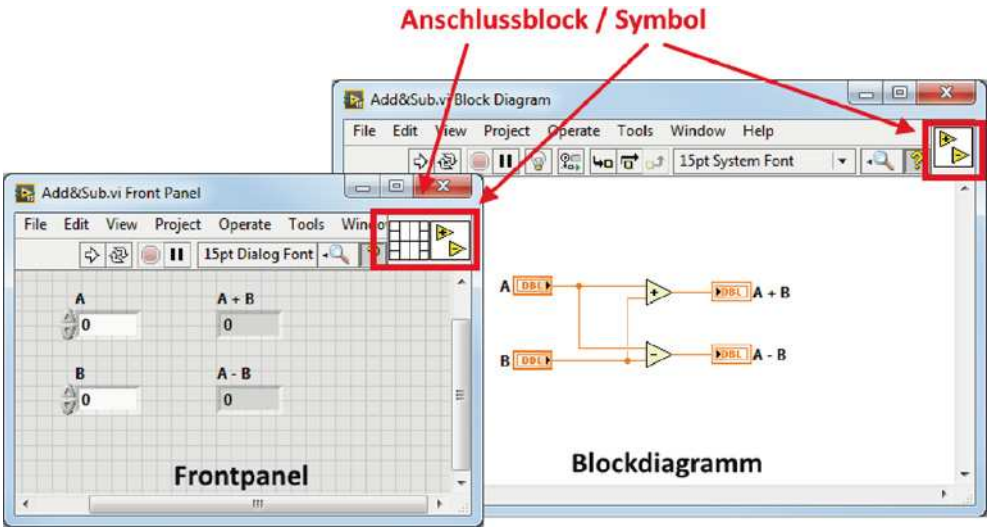


Bild 1.3 Hauptbestandteile eines VIs

1.3.1 Frontpanel

Das Frontpanel stellt in LabVIEW die Schnittstelle zum Benutzer dar oder zu anderen VIs. Es beinhaltet sämtliche Eingänge und Ausgänge, die für eine Interaktion mit der Welt außerhalb des VIs benötigt werden (Bild 1.4).

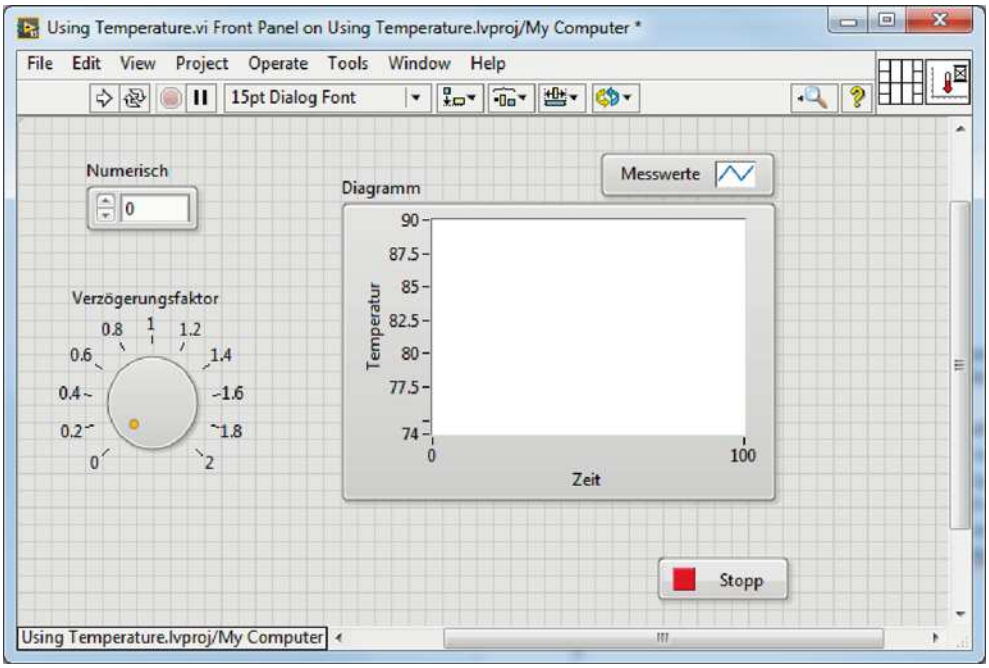


Bild 1.4 Frontpanel eines VIs

Die Elemente werden in zwei Gruppen unterteilt: in Bedien- und Anzeigeelemente (Tabelle 1.1).

Tabelle 1.1 Controls (Bedienelemente) und Indicators (Anzeigeelemente)

Controls	Indicators
Buttons, Drehknöpfe, Schieberegler, Texteingabefelder etc.	LEDs, Graphen, Diagramme, Textboxen etc.

Die Symbolleiste des Frontpanels enthält verschiedene Funktionen zur Ausführung des VIs sowie zur Gestaltung und Anordnung der vorhandenen Frontpanel-Elemente (Bild 1.5).

Bild 1.5 Symbolleiste des Frontpanels

Die in Tabelle 1.2 beschriebenen Buttons dienen der Kontrolle der Ausführung des entsprechenden VIs.

Tabelle 1.2 Funktionelle Buttons der Symbolleiste

	Ausführen des VIs
	Wiederholtes Ausführen des VIs
	Sofortiger Abbruch der Ausführung des VIs
	Pausieren der Ausführung des VIs

Mit dem Dropdown-Menü, gleich daneben, lassen sich Schriftarten der Frontpanel-Elemente und deren Aussehen verändern (Bild 1.6).

Bild 1.6 Schrifteditor

Die weiteren Funktionen dienen der optimierten Darstellung und der Ausrichtung der Elemente auf dem Frontpanel und haben keinen Einfluss auf das Programmverhalten des VIs. Trotzdem sollte ein Frontpanel ansprechend sein und alle Controls und Indicators nach einem übersichtlichen Muster angeordnet sein. Die folgenden Tools in Tabelle 1.3 helfen dem Programmierer bei der Gestaltung.

Die verschiedenen Datentypen werden auf dem Frontpanel als Controls und Indicators dargestellt (Tabelle 1.4).

Tabelle 1.3 Funktionen zur Darstellung und Ausrichtung von Frontpanel-Elementen

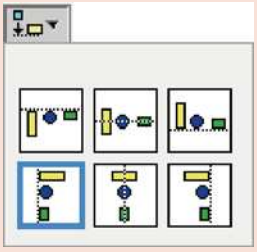
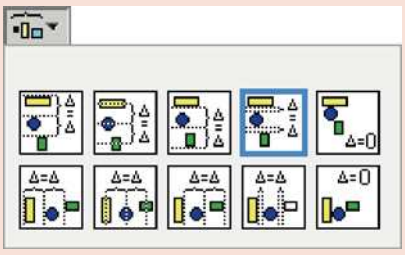
	Align – Ausrichtung der Elemente: Mit diesem Tool können selektierte Elemente sowohl horizontal (linker Rand, Mitte, rechter Rand) als auch vertikal (oberer Rand, Mitte, unterer Rand) zueinander ausgerichtet werden.
	Distribute – Verteilen der Elemente: Mit diesem Tool können selektierte Elemente gleichmäßig zueinander verteilt werden, z. B. in gleiche Abstände zwischen mehreren Elementen (Selektion).

Tabelle 1.3 Funktionen zur Darstellung und Ausrichtung von Frontpanel-Elementen (Fortsetzung)

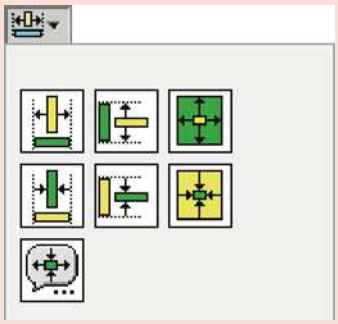
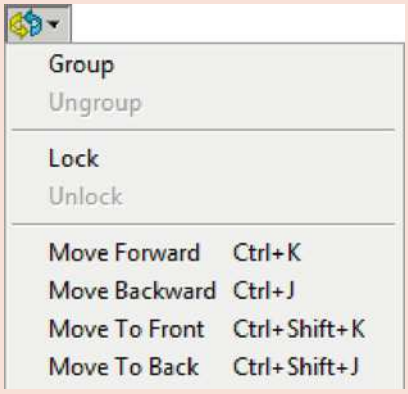
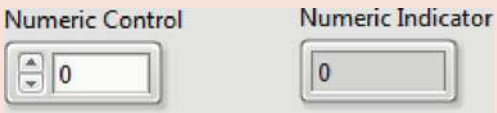
	Resize – Größenanpassung von Elementen: Mit diesem Tool können selektierte Elemente auf die gleichen Dimensionen getrimmt werden.
	Reorder – Gruppieren und Ebenen festlegen: Mit diesem Tool können selektierte Elemente gruppiert und gelockt werden. Ebenfalls kann bei übereinanderliegenden Elementen die Reihenfolge bestimmt und Elemente in den Vorder- bzw. Hintergrund verschoben werden.

Tabelle 1.4 Controls und Indicators der gebräuchlichsten Datentypen in LabVIEW

	Mit einem numerischen Datentyp können Zahlen verschiedenen Typs dargestellt werden, z. B. Gleitkommazahlen, ganzzahlige Typen etc. Numerische Controls sowie Indicators gehören zu den am häufigsten eingesetzten Elementen.
	Mit dem booleschen Datentyp lassen sich die Zustände TRUE und FALSE darstellen. Andere Elemente außer Buttons sind z. B. Slide Switches oder Radio Buttons.
	Strings sind zusammengesetzte Ketten aus ASCII-Zeichen. Häufig werden Controls als Eingabeelemente für Benutzertext verwendet, Indicators z. B. zur Anzeige einer Statusmeldung oder anderer Messages.

Das Frontpanel dient als Benutzeroberfläche eines VIs.

1.3.2 Blockdiagramm

Hinter dem Blockdiagramm versteckt sich die eigentliche Funktionalität des VIs (Bild 1.7). Im Gegensatz zum Frontpanel ist das Blockdiagramm für den Benutzer zur Laufzeit nicht sichtbar. Zu typischen Objekten des Blockdiagramms zählen Anschlüsse, SubVIs, Funktionen, Konstanten, Strukturen und Wires (Verbindungen), über welche die Daten zwischen den Objekten des Blockdiagramms übertragen werden.

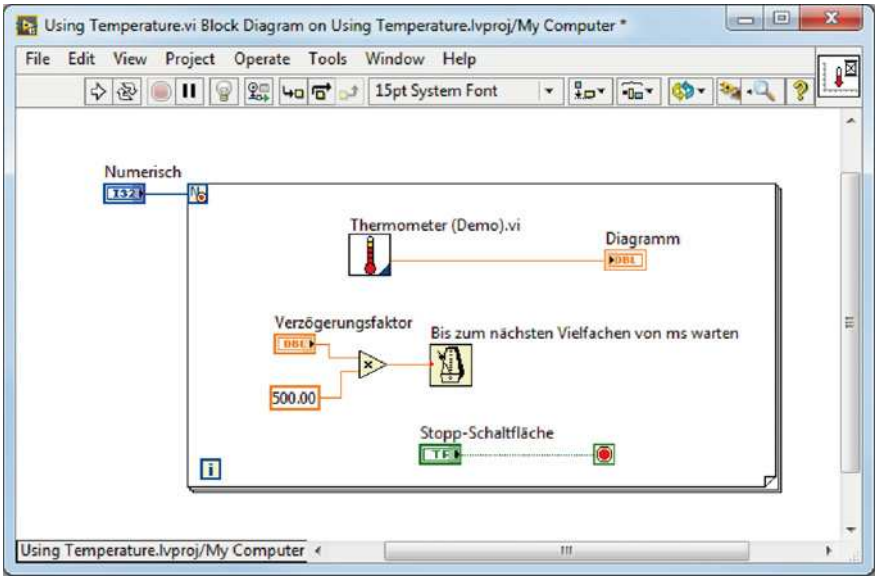


Bild 1.7 Blockdiagramm eines VIs

Die Symbolleiste des Blockdiagramms (Bild 1.8) enthält zum Teil dieselben Funktionen wie diejenige des Frontpanels, ist aber zusätzlich mit Debug- und Analysetools ausgerüstet (Tabelle 1.5).

Bild 1.8 Symbolleiste des Blockdiagramms

Tabelle 1.5 Debugging-Funktionen des Blockdiagramms

	Highlight-Modus: Ausführung des Programms wird extrem verlangsamt, sodass alle Werte auf den Wires sichtbar werden
	Verbindungswerte speichern: Wenn eingeschaltet, behalten die Wires ihre Werte, auch wenn das Signal das Wire schon passiert hat
	Step into, step over, step out: Programmausführung im Einzelschritt-Modus zur Kontrolle aller Signale

Das Blockdiagramm enthält den Programmcode und die Funktionalität eines VIs.

Die Controls (Bedienelemente) und Indicators (Anzeigeelemente), welche bereits im Frontpanel eingesetzt wurden, müssen im Blockdiagramm natürlich funktional eingebunden werden.

In Bild 1.9 sind die gleichen Controls und Indicators abgebildet, welche im Frontpanel schon in Tabelle 1.4 aufgeführt wurden. Grundsätzlich unterscheiden sich Controls und Indicators im Blockdiagramm aufgrund der Position des schwarzen Anschlusspfeils (wo eine Verbindung erfolgen muss) sowie der Breite des Rahmens. Controls sind immer Datenquellen, weshalb der Ausgangsanschluss rechts liegt, Indicators sind die Verbraucher der Daten, weshalb der Eingang stets links liegt. Die Farbe des Elements, egal ob Control oder Indicator, entspricht der Farbe des enthaltenen Datentyps.

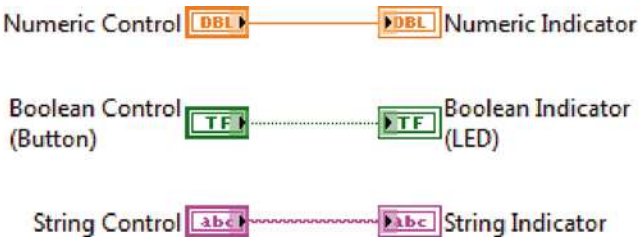


Bild 1.9 Controls und Indicators im Blockdiagramm

Die Elemente können im Blockdiagramm auch als Icons dargestellt werden (Bild 1.10). Eventuell ist dies bei der Installation auch so voreingestellt. Da diese Ansicht jedoch keinen Vorteil bringt und erst noch mehr Platz benötigt als die normalen Symbole, sollten die Anschlüsse nie als Icons angezeigt werden.

Bild 1.10 Darstellung der Elemente als Icons

Um einzelne Controls oder Indicators zu ändern, muss mit Rechtsklick auf das Element und unter View As Icon der Haken entfernt werden. Um die Einstellung für alle zukünftig platzierten Elemente zu setzen, muss unter Tools → Options in der Kategorie Block Diagram der entsprechende Haken entfernt werden (Bild 1.11).

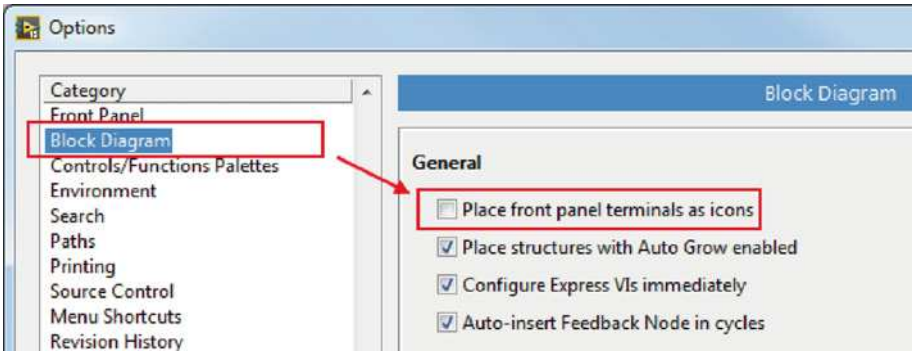


Bild 1.11 Options Dialog

1.3.3 Symbol und Anschlussblock

Jedes VI verfügt über ein Symbol und einen Anschlussblock. Wird ein VI im Programmcode (Blockdiagramm) eines anderen VIs verwendet, ist es wichtig, dass es auf den ersten Blick identifizierbar ist und der Programmierer anhand des Symbols erkennt, welche Funktion das VI übernimmt. In Bild 1.7 befindet sich ein VI mit dem Namen Thermometer (Demo).vi im Programmcode des Haupt-VIs. Anhand des Symbols ist klar, dass sich hinter dem VI eine Temperaturapplikation verbergen muss.

Das Symbol kann einfach über einen Doppelklick auf das VI-Symbol in der rechten oberen Ecke des Frontpanels oder des Blockdiagramms editiert werden. Darauf öffnet sich der Icon-Editor (Bild 1.12), wo Text und vordefinierte Icons im 32 × 32 Pixel großen Symbolfeld platziert werden können.

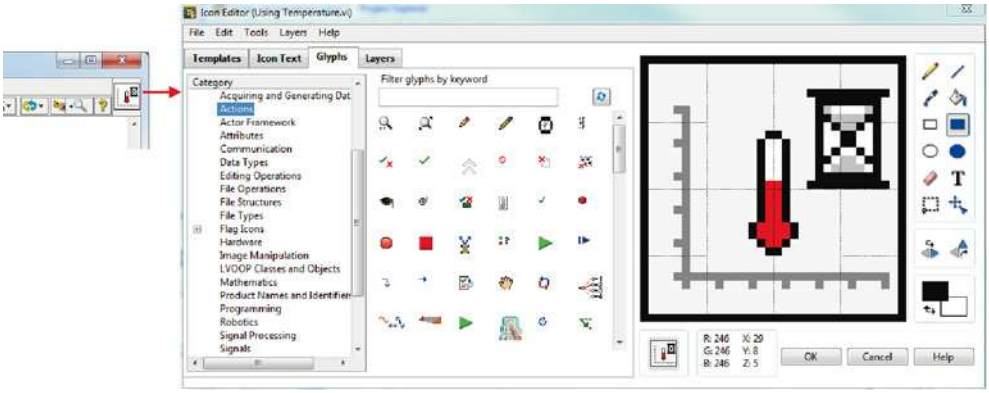


Bild 1.12 VI Icon-Editor

Das Symbol dient der einfachen Erkennung des VIs, wenn es innerhalb von anderen VIs verwendet wird. Dabei muss nicht zwingend eine Grafik als Symbol verwendet werden, es reicht auch eine kryptische Beschreibung oder eine Kombination von Symbol und Text. In Bild 1.13 sind Beispiele aussagekräftiger VI-Symbole dargestellt.

Bild 1.13 Beispiele von aussagekräftigen VI-Symbolen

Jedes VI muss mit einem individuellen Symbol ausgestattet sein.

Der Anschlussblock eines VIs ist die Schnittstelle zu einem übergeordneten VI. Er definiert, an welchen Ein-/Ausgängen welche Daten dem VI übergeben bzw. vom VI zurückgegeben werden. Werden die einzelnen Felder des Anschlussblocks mit Frontpanel-Elementen verbunden, nehmen sie automatisch die Farbe des entsprechenden Datentyps an. Hierzu kann mit der Drahtrolle (automatisches Tool) auf ein leeres Feld des Anschlussblocks geklickt werden und danach auf das Frontpanel-Element, welches mit dem Feld verbunden werden soll (Bild 1.14).

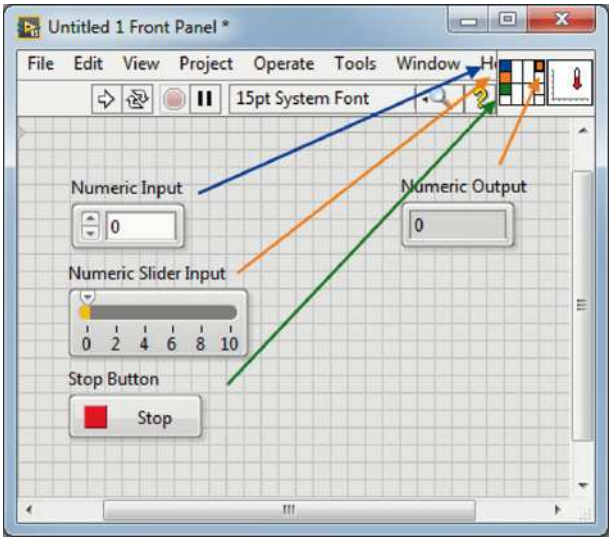


Bild 1.14 Anschlussblock eines VIs

Für die Verwendung in einem übergeordneten VI muss der Anschlussblock definiert sein, um Werte übergeben und übernehmen zu können.

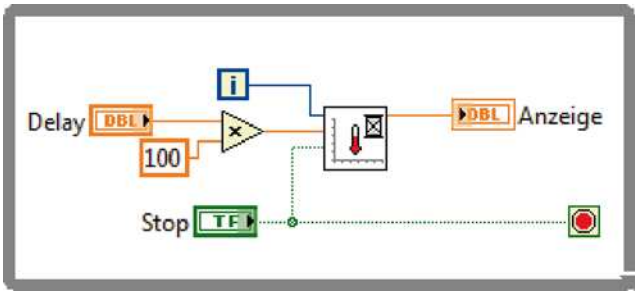


Bild 1.15 Verwendung eines VIs mit Ein- und Ausgängen

Für den Anschlussblock stehen verschiedene Muster mit einer kleineren bis größeren Zahl Anschlüsse zur Verfügung. Im Anschlussblock mit einem Rechtsklick auf Patterns lässt sich das Muster verändern (Bild 1.16).

In LabVIEW wird nach dem Datenflussprinzip programmiert (Kapitel 3). Das heißt, die Verarbeitung verläuft von links nach rechts. Deshalb ist es wichtig, diesem Prinzip auch bei der Belegung von Anschlussblöcken Rechnung zu tragen. Eingänge werden grundsätzlich auf der linken Seite verbunden, Ausgänge rechts, wie in Bild 1.17 dargestellt. Etwaige zusätzliche Anschlussfelder oben und unten werden erst verwendet, wenn die seitlichen Anschlüsse nicht mehr ausreichen.

Index

S

T

U

V

W

X

Z