

LABVIEW

Graphen

	Waveform Chart (Fortlaufende Kurve)	Waveform Graph (Kurvenanzeige)	XY-Graph																		
Datenformat	Einzelne Y-Werte	Y-Array	X-und Y-Array																		
Input	<table border="1"> <thead> <tr> <th>Data</th> <th>Resulting Chart</th> </tr> </thead> <tbody> <tr> <td>Scalar</td> <td>Single plot - 1pt</td> </tr> <tr> <td>1D</td> <td>Single Plot - 1 or more pts</td> </tr> <tr> <td>WDT</td> <td>Single Plot - 1 or more pts</td> </tr> <tr> <td>2D</td> <td>Multiplot - 1 or more pts</td> </tr> </tbody> </table>	Data	Resulting Chart	Scalar	Single plot - 1pt	1D	Single Plot - 1 or more pts	WDT	Single Plot - 1 or more pts	2D	Multiplot - 1 or more pts	<table border="1"> <thead> <tr> <th>Y Array</th> <th>Resulting Graph</th> </tr> </thead> <tbody> <tr> <td>1D</td> <td>Single Plot</td> </tr> <tr> <td>WDT</td> <td>Single Plot</td> </tr> <tr> <td>2D</td> <td>Multiplot</td> </tr> </tbody> </table>	Y Array	Resulting Graph	1D	Single Plot	WDT	Single Plot	2D	Multiplot	<p>single</p> <p>multiple</p>
Data	Resulting Chart																				
Scalar	Single plot - 1pt																				
1D	Single Plot - 1 or more pts																				
WDT	Single Plot - 1 or more pts																				
2D	Multiplot - 1 or more pts																				
Y Array	Resulting Graph																				
1D	Single Plot																				
WDT	Single Plot																				
2D	Multiplot																				
Als Bündel																					
Eine Kurve																					

Bestandteile eines virtuellen Instrumentes (VI) (=Programme)

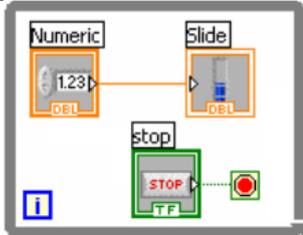
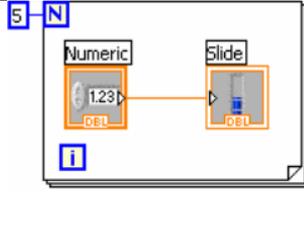
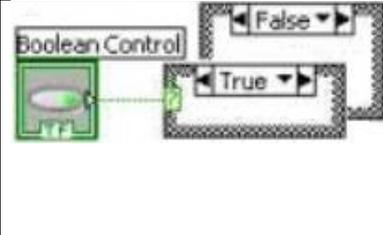
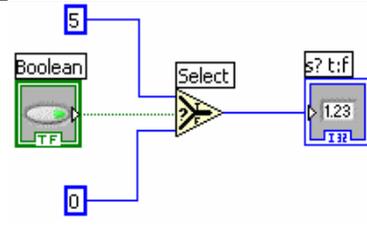
Frontpanel (Benutzeroberfläche)	Blockdiagramm (Steuerung des Frontpanels)	Symbol und Anschlussfeld (Verwendung von VI in einem Anderen)
<p>Bedienelemente: Eingaben Anzeigelemente: Ausgaben</p>	<p>Bedienelemente, Anzeigelemente, Konstanten, Freie Beschriftungen</p>	<p>(Elementtypen) Anzeigelemente Bedienelemente</p> <p>Unterschiede: Beschriftung =Label</p>

Knoten

- Ein Knoten wird dann ausgeführt, wenn an allen Eingangsanschlüsse Daten anliegen.
- Die Daten werden erst an den Ausgangsanschluss weitergegeben, wenn die Ausführung des Knoten beendet ist.
- Liegen die Eingangsgrößen gleichzeitig an, ist nicht eindeutig, was zuerst ausgeführt wird.

Datentypen

numerisch	boolesch	String	Enum
ganze oder reelle Zahlen	TRUE / FALSE	ASCII-Zeichen	Kombination
Fliesskommazahl (Single/Double/Extended) Fextkommatentyp (FXP) Integer mit Vorzeichen (I ..) Integer ohne Vorzeichen (U ..) Komplexe Zahl (C..) (Single/Double/Extended)	Beim Drücken schalten Beim Loslassen schalten		String + Nummer

Instrumente			
While-Schleife	For-Schleife	Case-Strukturen	Wählen / Select
			
i = Iterationsanschluss zählt die Schleifendurchläufe Zeitverzögerung mit „Time Delay“		Bool Eingangswert	Oben = True, Unten = False
Läuft min. 1 mal oder bis Abbruchbedingung		Läuft bis N erreicht ist	

Datenzusammenfassungen		
Array	Cluster	Typdefinitionen
mehrdimensional Index beginnt mit 0 initialisieren optional Indexanzeige = erster Anzeigewert <u>Erstellen</u> Array-Container erzeugen Element in Container ziehen (evt. „Add Dimension“) <u>Verwendung</u> Elementenweise mit einer For-Schleife Ausführungen = Arraygrösse Mehrere Eingänge = kleinstes Array <u>2D-Arrays</u> zwei For-Schleifen ineinander (Innere = Spaltenelemente) Auto-Indexierung aktiv liefert Array, deaktiviert liefert es den letzten Wert	gruppieren unterschiedlichen Typen + übersichtlicher, weniger Anschlüsse max. 28 Anschlüsse, >28 verschachtelt jeder Anschluss hat eine Beschriftung nur einen Elementtyp (Bedien/Anzeigeel.) <u>Erstellen</u> Cluster-Container erzeugen Element in Container ziehen <u>Bearbeiten</u> Mit „Bündeln“ oder „Aufschlüsseln“ unabhängig durch „Nach Namen ...“ <u>Fehler-Cluster</u> enthält Status (bool), Code (32-Int) und Quelle (String)	benutzerdef. Element: lokale Änderung Type Definition: globale Änderung <u>Typdefinition</u> Datentyp ändert mit Grösse, ... ändert nicht mit <u>strikte Typdefinition</u> keine Abweichung mit Typdefinition möglich ausser Untertitel, Beschriftung, Beschreibung, Hinweisstreifen und Standardwert Datentyp, Grösse ändert mit Element Editor (RMT, Advanc., Custom.) <u>Bearbeitungsmodus</u> Einstellungen vornehmen <u>Anpassungsmodus</u> Verschieben, Typendefinition

Verwalten von Ressourcen			
Datei-I/O	High-Level-Datei-I/O	Low-Level-Datei-I/O	DAQmx-Programmierung
<u>Schritte</u> Erstellen/Öffnen, speichern, schliessen, (auf Fehler prüfen) <u>Dateiformate</u> Binär (basis) ASCII (Textdateien) LVM (LabView-Messdaten, Tabulatorgetrennt) TDMS (Binär+Indexdatei)	<u>Schritte</u> Alle <u>Typen</u> In Tabellenkalk. schreiben Aus Tabellenkalk. lesen Messwerte in Datei schreiben Messwerte aus Datei lesen <u>Verwendung</u> Alles in einem Vorgang	<u>Schritte</u> Nur einen <u>Verwendung</u> bei Schleifen Datenträger-Streaming: Dateien bleiben geöffnet <u>Ergebnis</u> schneller, da nur einmal geöffnet werden muss	Task erstellen Task konfigurieren Task starten Daten erfassen/erzeugen Task zurücksetzen VISA = High-Level-API, die Low-Level Treiber aufruft

Modularität / SubVIs	
Prinzip	Bestandteile
Untergeordnete VIs = SubVIs (Unterprogramm) Symbol wird in der rechten oberen Ecke angezeigt, Zahl bedeutet Anzahl der neuen VIs seit LabView-Start	<u>Symbol</u> : (Text oder Grafiken) <u>Anschlussfeld</u> : (Ein- und Ausgänge(max.28)) RMT auf Symbol (Show Connector)

Entwurfsmethoden und -muster		
Sequenzielle Programmierung	Zustandsprogrammierung	Zustandsautomaten
Festlegen der Reihenfolge Parallele Vorgänge sind nicht möglich können nicht abgebrochen werden =werden bis zum Schluss ausgeführt <u>Erstellen</u> Funktionen/Programming/Strukturen	komplexe Programmierungen <u>Verwendung</u> Ändern der Reihenfolge Ausführung unter Bedingungen Unterbrechen von Sequenzen	Entscheidungsalgorithmen mehrere Zustände, Übergangsfunktionen <u>Komponenten</u> Case-Struktur in While-Schleife, Schieberegister Code für Funktion des Zustandes, Code für Übergang
Parallele Ausführung bei nicht gemeinsam genutzten Ressourcen		

