

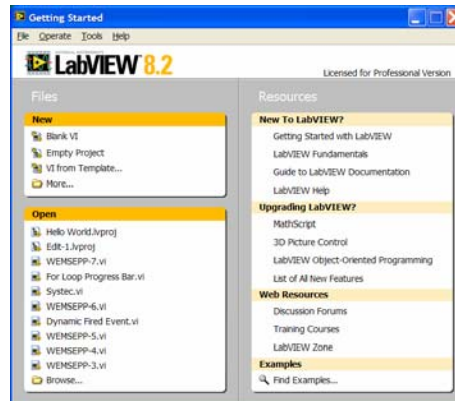
LabVIEW 8.21 Vorlesung 2007/2008

Herbert Pichlik

herbert@pichlik.de

0175-2421820 0911-998955631

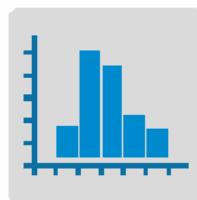
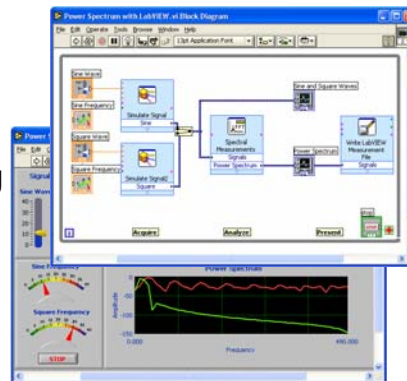
September 2007



1

Grafische Programmierplattform

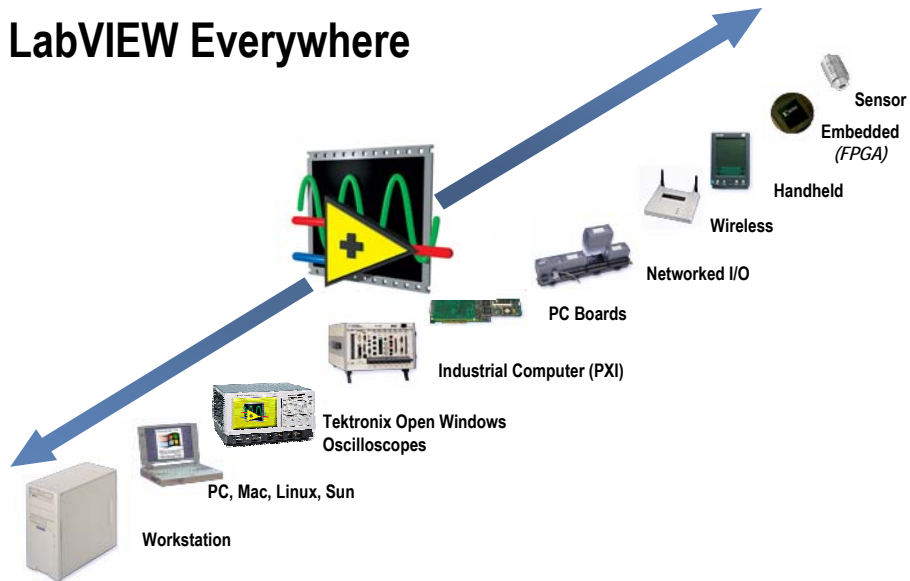
- Rapid Application Development und Rapid Prototyping
- Ereignissteuerung, Objektorientierung und Verteilte Anwendungen
- Targets: Real-Time, FPGA, PDA, Embedded, DSP, etc.
- Versionen in engl., dt. fr., jap., chin.



Erfassen, Analysieren und Präsentieren

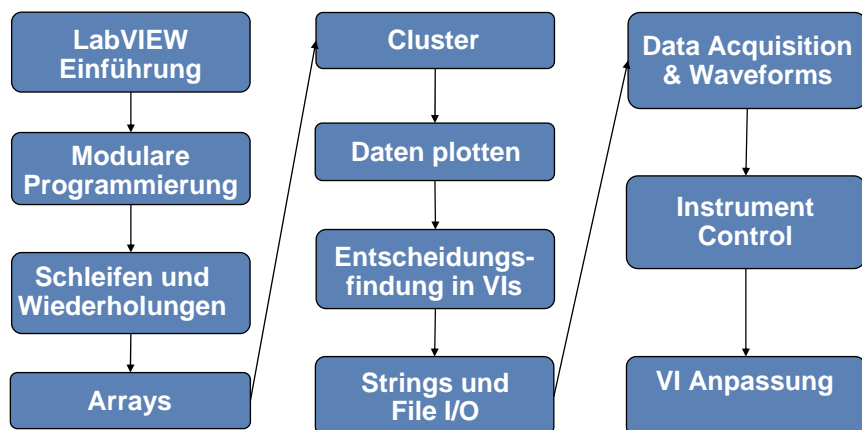
2

LabVIEW Everywhere



3

Kursübersicht



4

Lektion 1

Einführung in LabVIEW



Themen

LabVIEW Entwicklungsumgebung
 Front Panel
 Block Diagram
 Datenflussprogrammierung
 LabVIEW Hilfe und Dokumentation
 Fehlersuche und Debugging

5

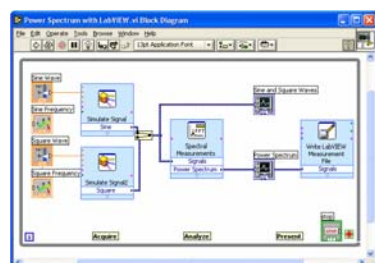
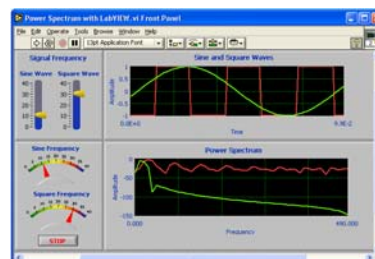
Virtuelle Instrumente (VIs)

Front Panel

- Controls = Eingabeelemente
- Indicators = Anzeigeelemente

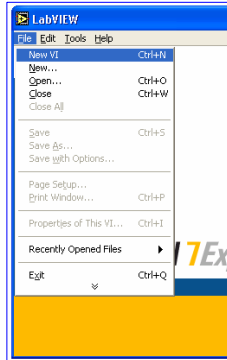
Block Diagram

- Sourcecode
- Komponenten verdrahtbar

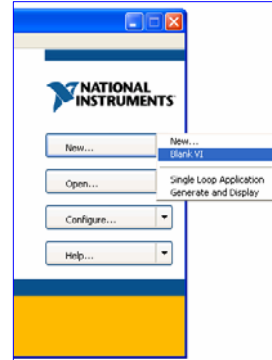


6

Neue Elemente erzeugen



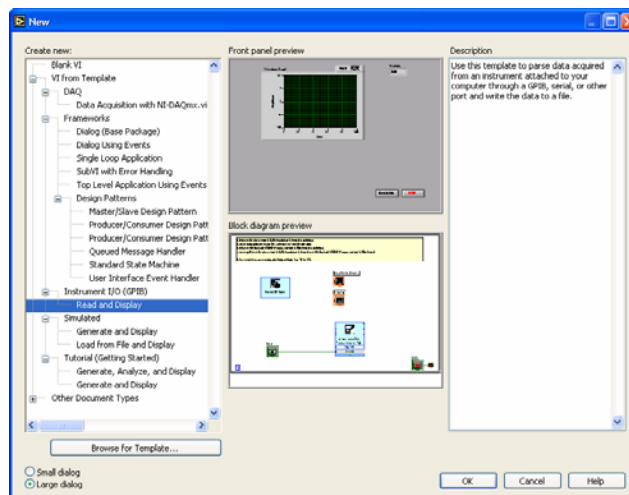
- File»New VI to open a blank VI



- File»New... ermöglicht die Erzeugung neuer Elemente wie Templates (Vorlagen), Controls, glob. Variablen, VIs, etc.

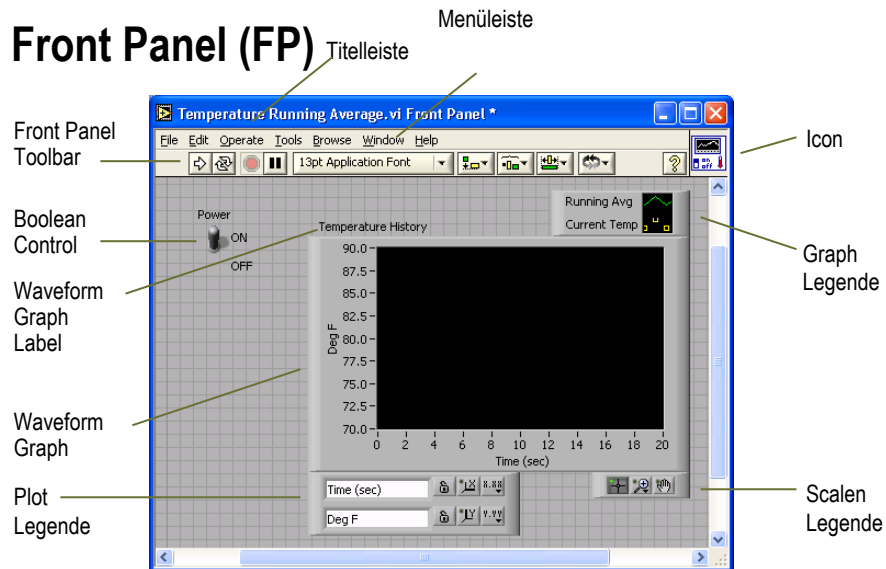
7

Template Browser

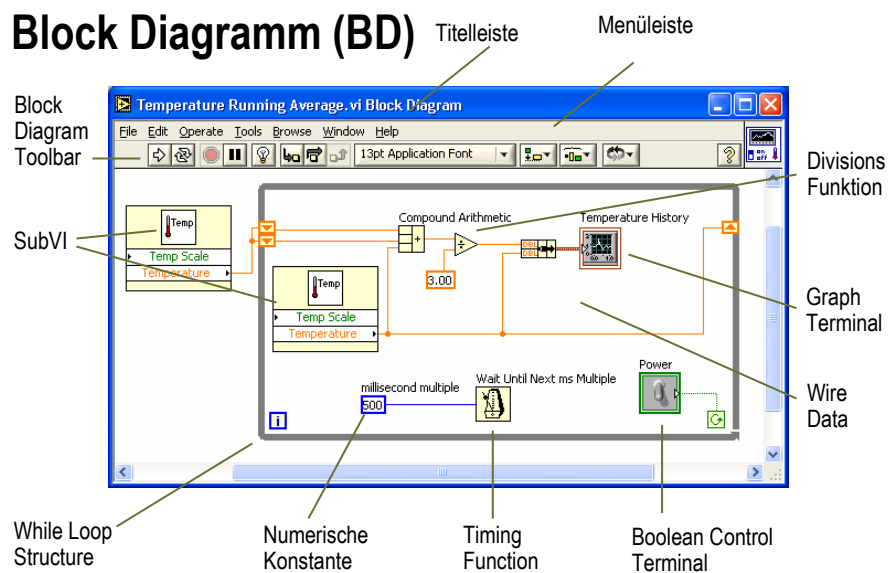


8

Front Panel (FP) Titelleiste



Block Diagramm (BD) Titelleiste



Front Panel und Block Diagramm Werkzeugleisten



Run button

Continuous Run button

Abort button

Pause/Continue button



Warning indicator



Enter button



Broken Run button



Additional Buttons on the
Block Diagram Toolbar

- Execution Highlighting button
- Step Into button
- Step Over button
- Step Out button

Font ring

Alignment ring

Distribution ring

Resize ring

Reorder ring

Context Help Button

11

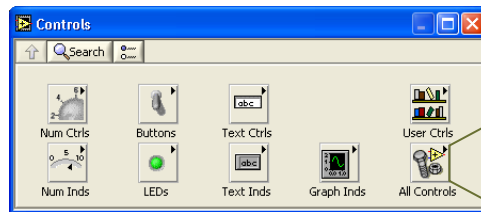
Tools Palette



- LabVIEW kann automatisch Werkzeuge selektieren
- Manuelle Selektion mit Leertaste (2 wichtigste Elemente) oder Tab-Taste (4 wichtigste Elemente)
- Im FP und BD verfügbar
- Werkzeuge sind Instanzen von Mauszeigeroptionen
- Werkzeuge ermöglichen die Veränderung von Elementen in FP und BD
- Anzeige mit **Window»Show Tools Palette** oder Shift+PopUp

12

Front Panel – Controls Palette



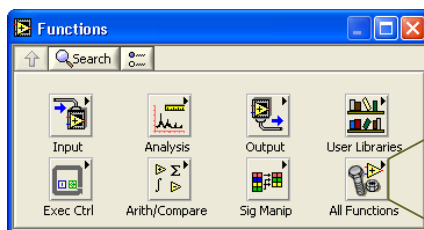
Controls Palette



All Controls Palette

13

Block Diagram – Functions Palette



Functions Palette
Express VIs

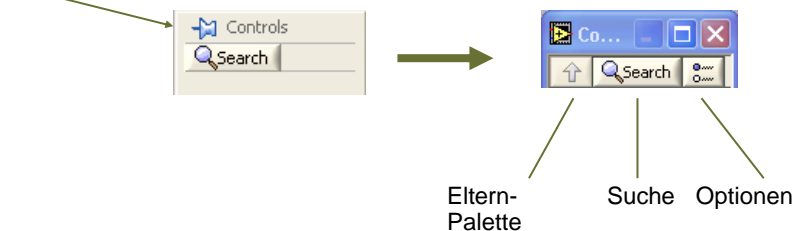


All Functions Palette
Alle Funktionen

14

Palette Tools

Pin, um Palette zu selektieren

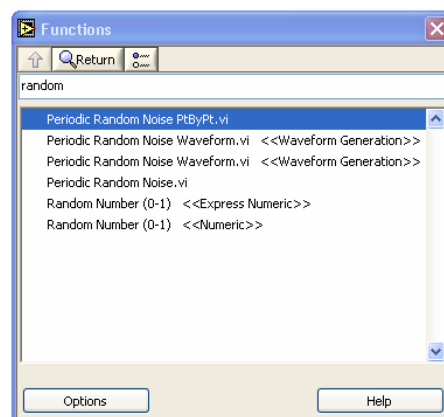


- Grafische "fließende" (auf Desktop frei bewegbare) Paletten
- Subpaletten können in "floating palettes" konvertiert werden
- Mit den Optionen kann zwischen Express und Advanced Palettensubset umgeschaltet werden

15

Suche nach Controls, VIs und Funktionen

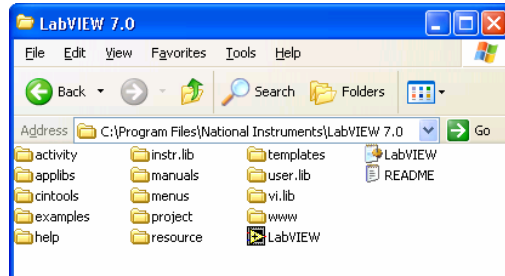
- Search-Knopf für Textsuche
- "Click and drag"-Funktionalität um Elemente in BD zu bringen. Mit Doppelklick wird Elternpalette geöffnet.



16

Anpassungen von Control & Funktions Paletten

Programme» National Instruments»LabVIEW 8.x

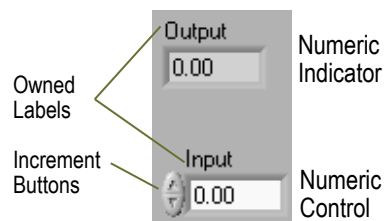
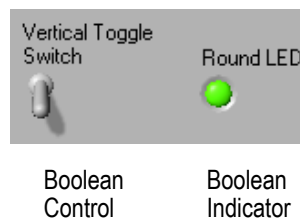


- `vi.lib` muss im LabVIEW Verzeichnis bleiben
- I.d.R. nur `user.lib` oder `instr.lib` erweitern um neue Elemente hinzuzufügen

17

VI Front Panel erzeugen

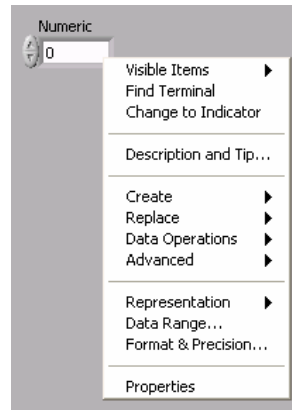
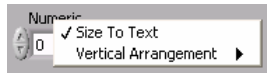
Erzeugen des Front Panels durch Platzieren von Eingabeelementen (Controls) sowie Anzeigeelementen (Indicators) und dekorativen Elementen (Decorations)



18

Shortcut Menüs für Front Panel Objekte

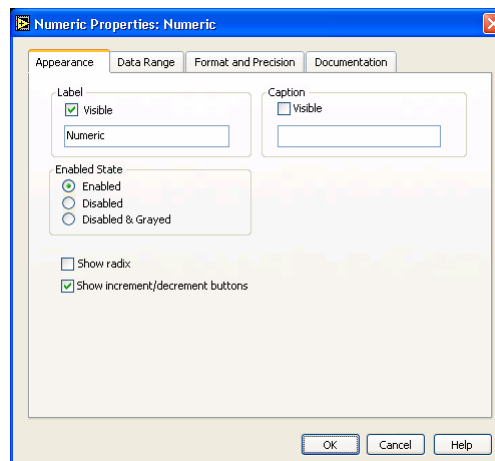
Rechtsklick auf Elemente ermöglicht die Einstellung von Optionen



19

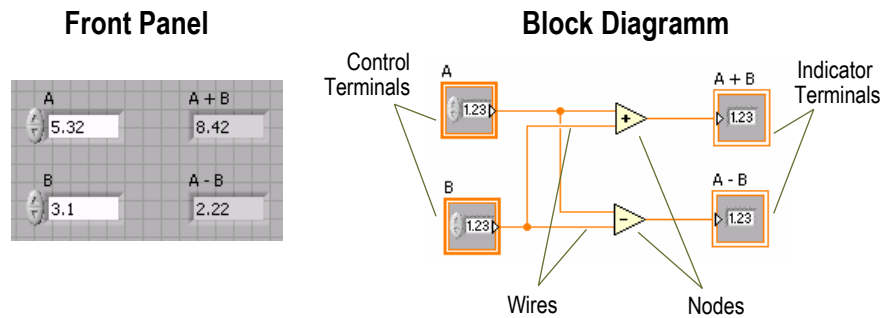
Property Page (Eigenschaftsseiten)

Rechtsklick auf Elemente und Auswahl "Properties" ermöglicht die Einstellung von Eigenschaften



20

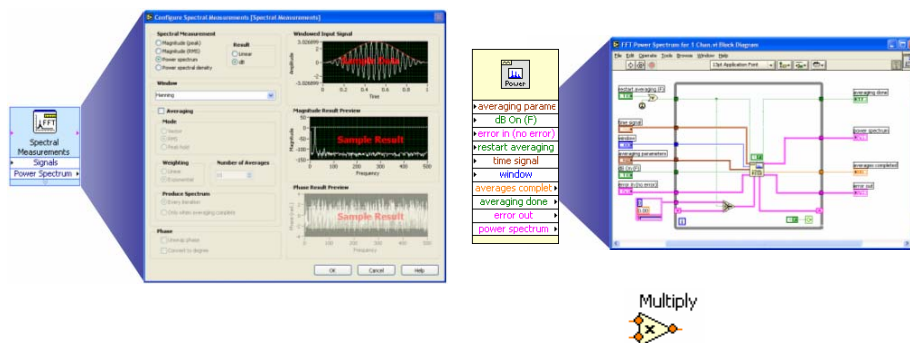
VI Block Diagramme erzeugen



21

Express VIs, VIs und Funktionen

- **Express VIs:** Interaktive VIs mit konfigurierbaren Dialogen
- **Standard VIs:** modularisierte VIs, Anpassung über Terminals
- **Functions:** LabVIEWs fundamentale Elemente; FP und BD nicht vorhanden

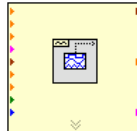


22

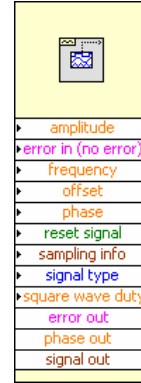
Block Diagramm – Darstellung von VIs



Expandable Node



Expanded Node

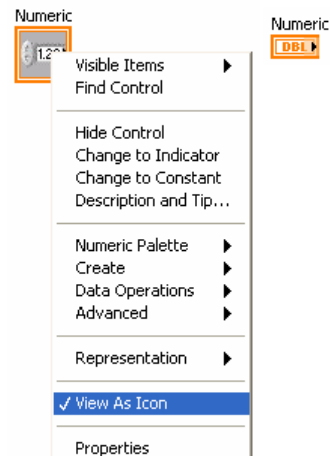


- Function Generator VI
- Drei Darstellungen
- Links: Standard VI
- Mitte und rechts: Express VI

23













Block Diagramm Terminals

- Terminals sind Ein- und Ausgangsstellen um Informationen zwischen VIs sowie FP und BD auszutauschen
- Terminals sind vergleichbar mit Parametern in textbasierten Programmiersprachen
- “View As Icon” über rechte Maustaste ein- und ausschaltbar



24

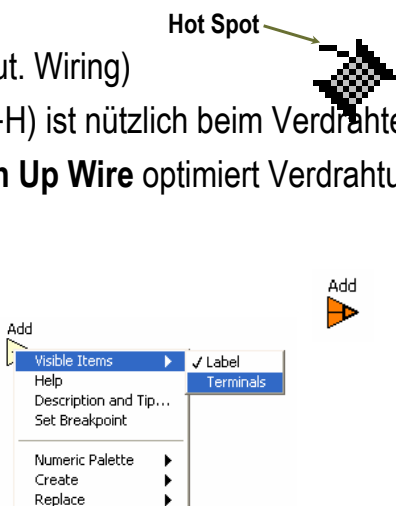
Block Diagramme verdrahten Darstellung

	Scalar	1D Array	2D Array
Numeric			
Boolean			
String			
Dynamic			

25

Wiring Techniques

- Automatische Verdrahtung (aut. Wiring)
- Context Help Fenster (STRG+H) ist nützlich beim Verdrahten
- Rechtsklick auf Draht + **Clean Up Wire** optimiert Verdrahtung
- Tip Strips verwenden
- Automatic wire routing
- Rechtsklick auf Terminals und selektieren **Visible Items»Terminals**

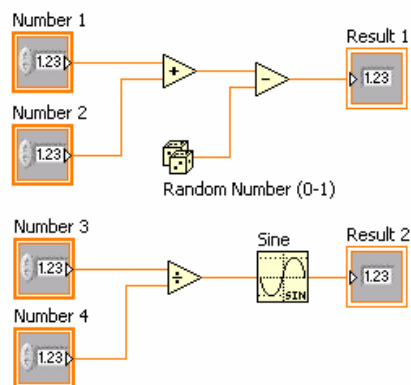


Anzeige von Terminalverbindungen in einer Funktion

26

Dataflow Programming

- Datenfluss analog zu Stromfluss (Datenquelle zu Datensenke)
- Knoten werden abgearbeitet, wenn alle Eingangsterminals gültige Daten haben
- Nach der Abarbeitung stehen Informationen an den Ausgangsterminals zur Verfügung



27

Kontexthilfe

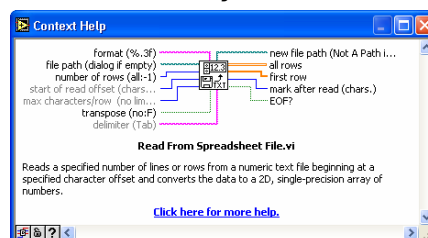
- **Help»Show Context Help**, <STRG-H> oder **Show Context Help Window** Knopf in Werkzeugpal.
- Hilfe Anzeigen durch Mauscursor über Objekt

- Verbindungen:

Required – fett

Recommended – normal

Optional - angegraut



Simple/Detailed Context Help

Lock Help

More Help

28

LabVIEW Hilfe

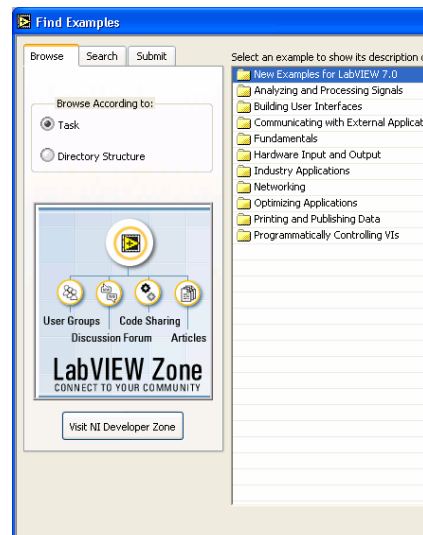
- Klicke auf More Help Knopf in Kontext Hilfe
- Wähle **Help»VI, Function, & How-To Help**
- Klicke auf **Click here for more help** in Kontext Hilfe
- Detaillierte Hilfe rund um LabVIEW
- Wichtig: Verwaltung von Favoriten



29

NI Example Finder

- **Help»Find Examples** um Beispiele zu suchen
- Web-Integration
- Vielfältige Suchoptionen



30

Debugging Techniken

Finding Errors



Klicken auf gebrochenen Pfeil listet Fehler

Execution Highlighting



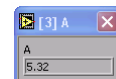
Execution Highlighting ermöglicht Visualisierung des Datenflusses (Bläschen auf Drahten, Skalare Werte und Anzahl Arrayelemente auf Terminals)

31

Debugging Techniques

Probe

Rechtsklick auf Draht und Auswahl



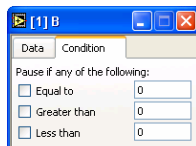
Breakpoints



Rechtsklick auf Draht und Auswahl Set Breakpoint; Hält Abarbeitung an diesem Punkt an (Pause) und ermöglicht Einzelschrittbetrieb

Conditional Probe

Kombination von Breakpoint und Probe. (mit Rechtsklick wählbar)



32

Debugging Techniken (Einzelschrittbetrieb)

Step Into, Step Over, Step Out Knöpfe



Schritt in Knoten



Schritt über Knoten



Schritt aus Knoten heraus

33

Tipps

• Tastatur-Shortcuts:

<u>Windows</u>	<u>Sun</u>	<u>Linux</u>	<u>MacOS</u>	
<Ctrl-R>	<♦-R>	<M-R>	<⌘-R>	Run a VI
<Ctrl-F>	<♦-F>	<M-F>	<⌘-F>	Find object
<Ctrl-H>	<♦-H>	<M-H>	<⌘-H>	Activate Context Help window
<Ctrl-B>	<♦-B>	<M-B>	<⌘-B>	Remove all broken wires
<Ctrl-W>	<♦-W>	<M-W>	<⌘-W>	Close the active window
<Ctrl-E>	<♦-E>	<M-E>	<⌘-E>	Toggle btwn Diagram/Panel Window

- Zugriff auf Tools Palette mit <shift>-Rechtsklick
- Increment/Decrement schneller durch Drücken der <shift> - Taste
- Tools»Options: Voreinstellungen für LabVIEW Umgebung
- VI Properties (File Menü Voreinstellungen für einzelne VIs)
- Stoppen von VIs durch Tastenkombination <STRG>+.

34

Lektion 2

Modulare Programmierung



Themen

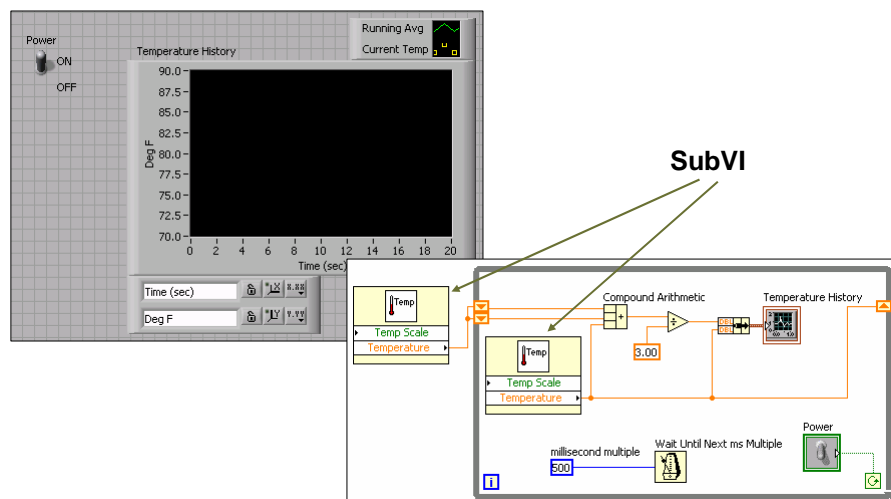
SubVIs

Icons und Connector Pane

SubVIs

35

LabVIEW Hierarchie



36

SubVIs

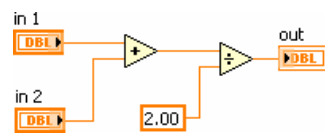
Function Pseudo Code

```
function average (in1,
    in2, out)
{
    out = (in1 + in2)/2.0;
}
```

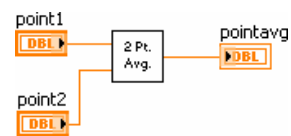
Calling Program Pseudo Code

```
main
{
    average (point1, point2,
        pointavg)
}
```

SubVI Block Diagram

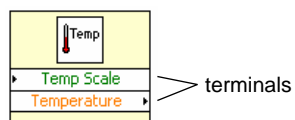


Calling VI Block Diagram



37

Icon/Connector



- Icons repräsentieren VIs in Elternblockdiagrammen
- Daten mit diesem VI werden über Terminals ausgetauscht

Icon



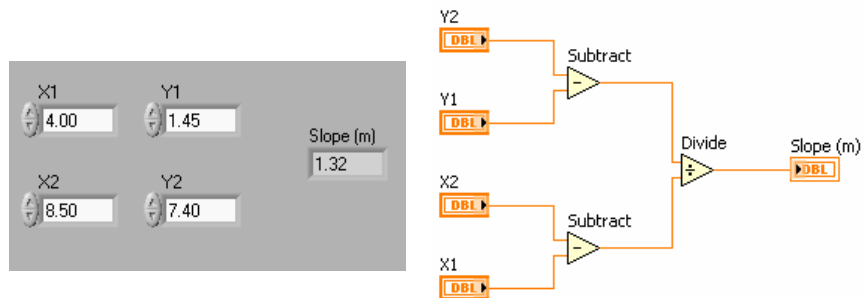
Connector



38

SubVI Beispiel – Berechnung einer Steigung

- VI innerhalb eines übergeordneten VI nennt man subVI
- Nach dem Editieren des Front Panels und des Blockdiagramms ist die ICON Pane und das Icon zu editieren. Beschreibungen nicht vergessen



39

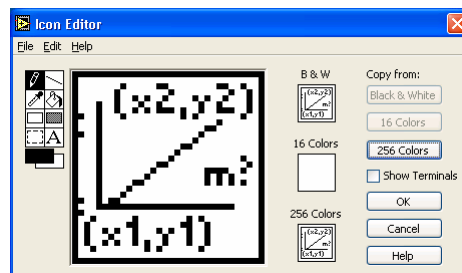
Icons erzeugen und verändern

- Icon: Grafische Repräsentation eines VIs
- Rechtsklick auf icon pane (Panel oder Diagramm)
- Immer alle Instanzen erzeugen
- Zwischenablage verwendbar über Shortcuts

Default Icon
(aut. Durch LV
durchnummeriert)

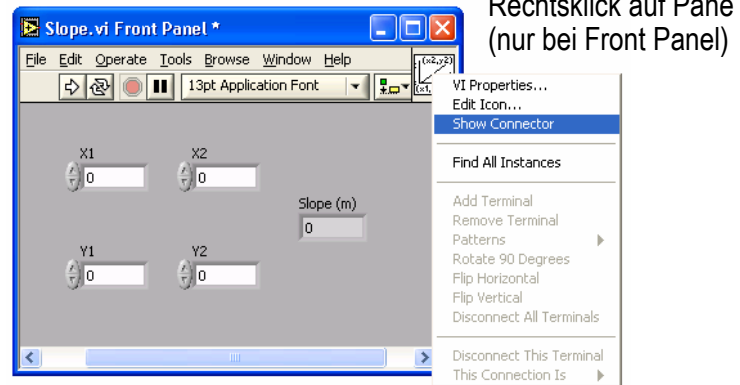


Kundenspez. Icon (besser)



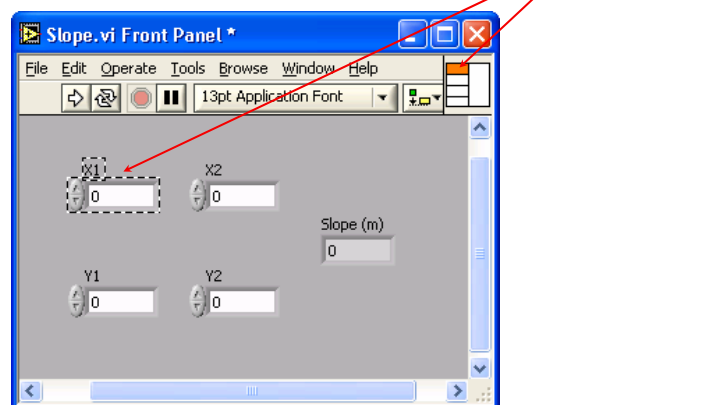
40

Konnektoren editieren



41

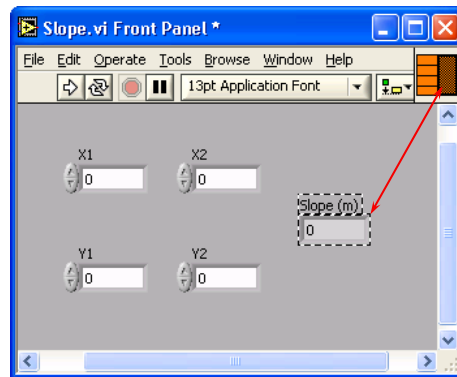
Verbindungen erzeugen



42

Die Connector Pane

Farben werden je nach Typ zugewiesen



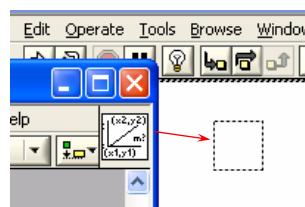
43

VIs als SubVIs benutzen

All Functions » Select a VI...

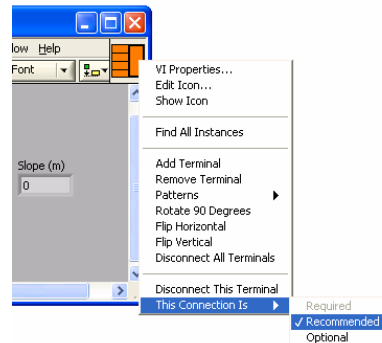
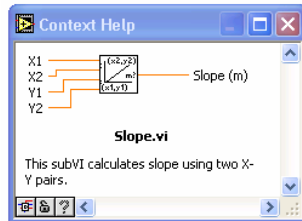
<ODER>

Ziehe Icon auf Zieldiagramm



44

Klassifizierung von Terminals und Hilfe



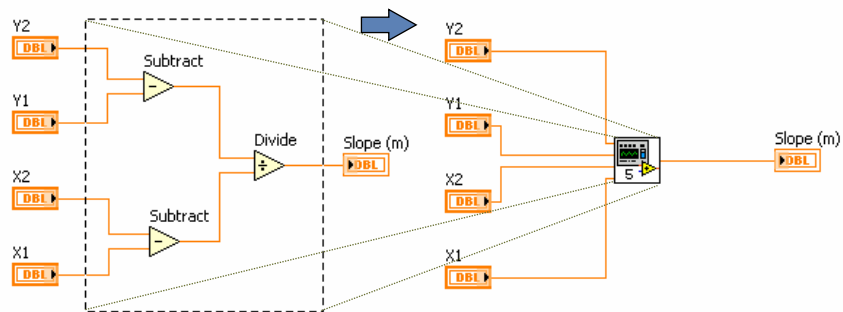
Klassifiziere Terminals

- Required — Muss verdrahtet werden sonst Fehler
- Recommended — Warnung wenn nicht verdrahtet
- Optional — kein Effekt falls nicht verdrahtet

45

Create SubVI Option

- Schließe Bereich ein
- Verwende Create SubVI aus Edit Menü



46

Lektion 3

Schleifen und Wiederholungen



Themen

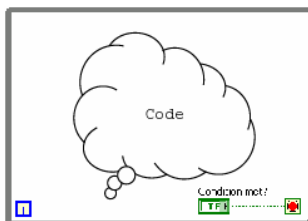
While Loops

For Loops

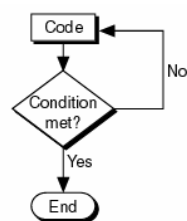
Zugriff auf Daten zurückliegender Iterationen

47

While Loops



LabVIEW While Loop



Flow Chart

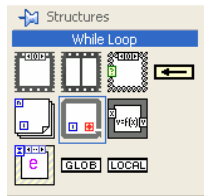
```
Repeat (code);  
Until Condition met;  
End;
```

Pseudo Code

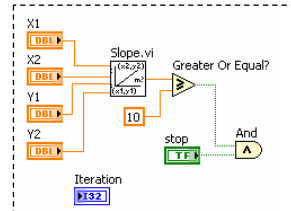
48

While Loops

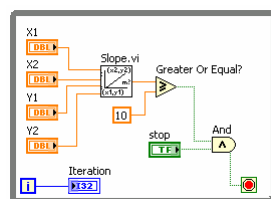
1. Wähle While Loop



2. Schließe zu wiederholenden Code ein



3. Füge zusätzliche Elemente ein und verdrahte

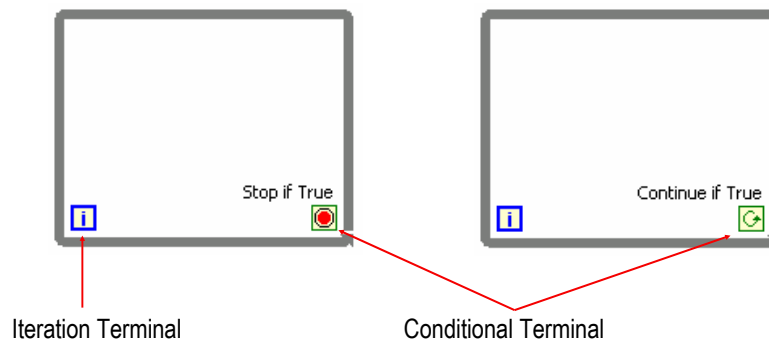


49

Loop Condition

Klicke auf Conditional Terminal mit dem Operating tool um das Abschaltverhalten zu ändern

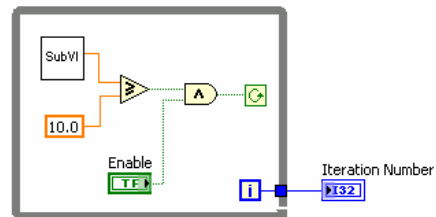
Default: Stop if True



50

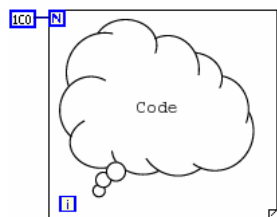
Structure Tunnels

- Tunnels ermöglichen die Datekommunikation von und zu Strukturen
- Tunnels sind Blöcke am Strukturrend die die direkte Kommunikation von und zu Strukturen ermöglichen
- Abarbeitung der Elemente erst nach Anliegen von Daten
- Daten werden erst nach Abarbeitung innerhalb der Struktur aus der Struktur ausgegeben

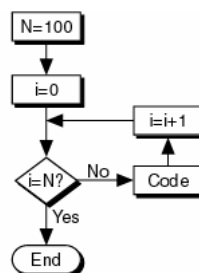


51

For Loops



LabVIEW For Loop



Flow Chart

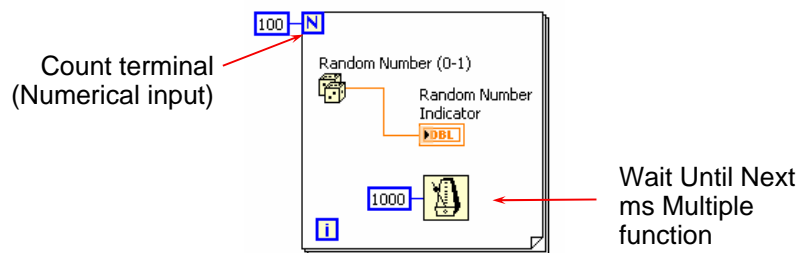
```
N=100;
i=0;
Until i=N:
    Repeat (code; i=i+1);
End;
```

Pseudo Code

52

For Loops

- In Structures Subpalette der Funktionspalette
- Führt Struktureninhalt eine fest gelegte Anzahl von Iterationen aus



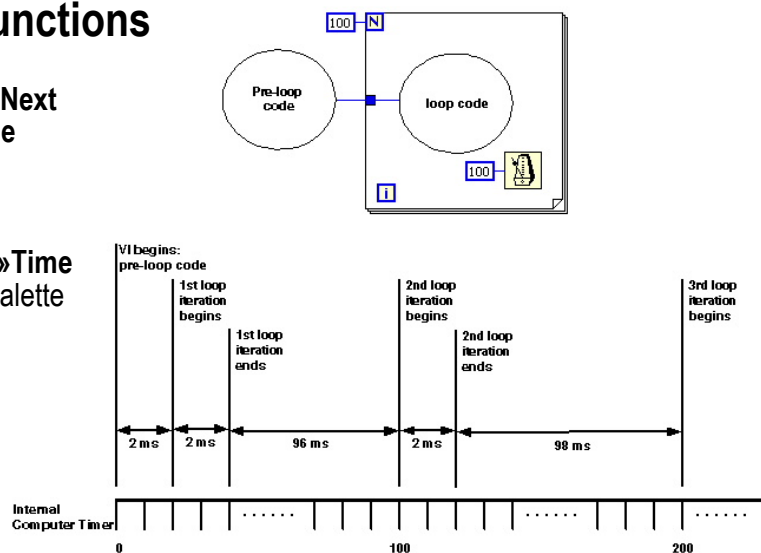
53

Wait Functions

Wait Until Next
ms Multiple



Functions»Time
& Dialog palette



54

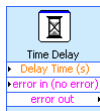
Wait Functions

Wait (ms)

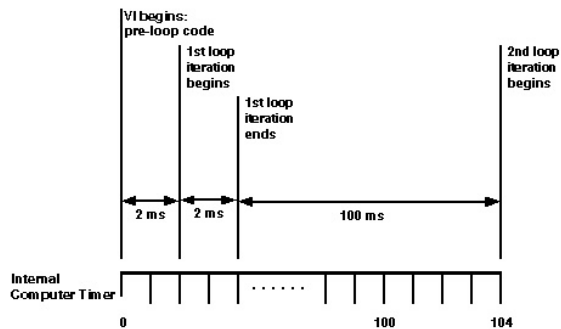
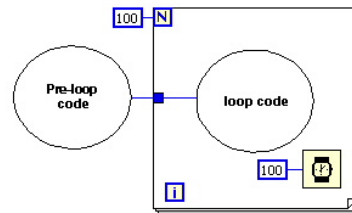


Functions»Time
& Dialog palette

Time Delay



Functions»Time
& Dialog palette



55

Numeric Conversion

- Numerics haben standardmäßig double-precision (8 bytes) oder long integer (4 bytes) Repräsentationen
- LabVIEW konvertiert automatisch (falls möglich)
- For Loop count terminal konvertiert immer in Long Integer
- Gray *coercion* dot oder roter Punkt (ab LV 8.0) zeigt Konvertierung an

Double-Precision,
Floating-Point Numeric



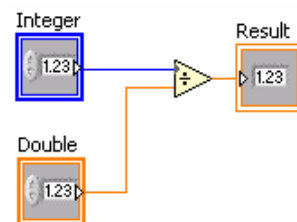
Long Integer



56

Numerische Konvertierung

- LabVIEW verwendet die hochauflösendere Repräsentation
- Mit Rechtsklick kann man die Repräsentation auswählen.
- Wenn LabVIEW Fließkommawerte in Ganzzahlwerte wandelt, rundet es auf den nächstliegenden Integerwert (z.B. rundet LabVIEW 2.4 auf 2 und 3.6 auf 4)

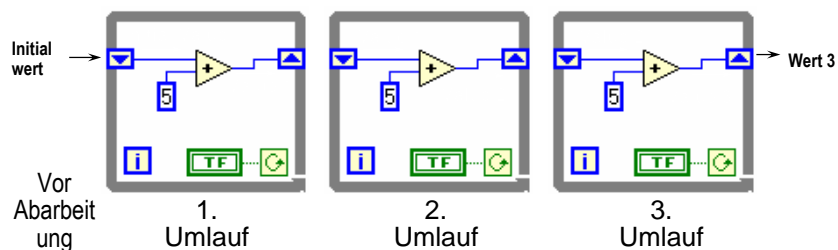


57

Zugriff auf zurückliegende Schleifendaten – Shift Register (Schieberegister)

Verfügbar am linken Schleifenrand

- Rechtsklick am Schleifenrand und Auswahl "Add Shift Register"
- Rechtes Terminal speichert Daten nachg Umlauf
- Linkes Terminal übernimmt gespeicherte Daten für den nächsten Umlauf



58

Zusätzliche Shift Register Elemente

Rechtsklick
und
Aufziehen
ermöglicht
mehr
Terminals



Rechtsklick
Ermöglicht
weitere
Schieberegister

59

Feedback Nodes

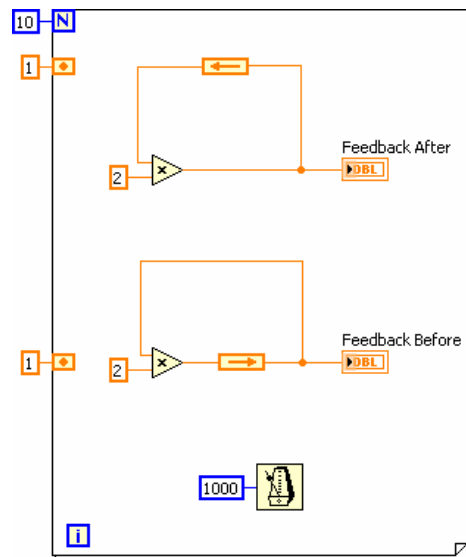


- Andere Darstellung für Schieberegister

60

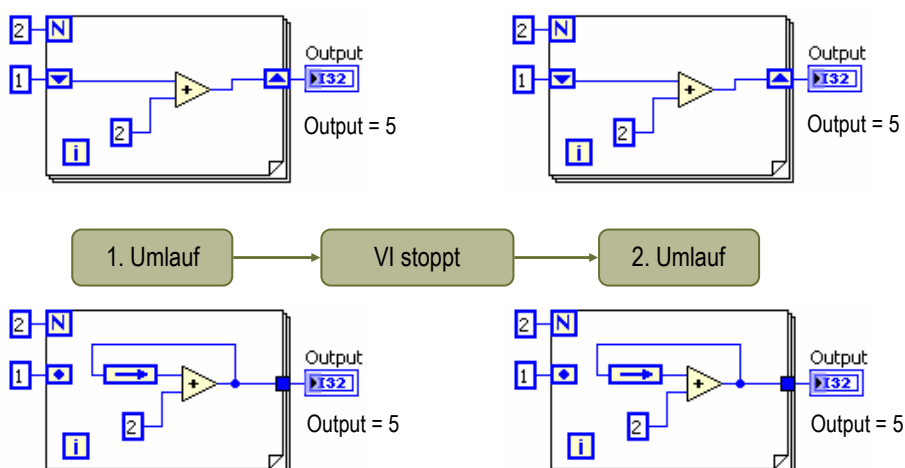
Feedback Nodes

- Erzeugt durch Verbindung von Outputs zu Inputs
<Oder>
- Platzieren über **Functions»Structures** Palette
Analog zu Schieberegistern
<Oder>
- Durch Konvertierung der Schieberegister



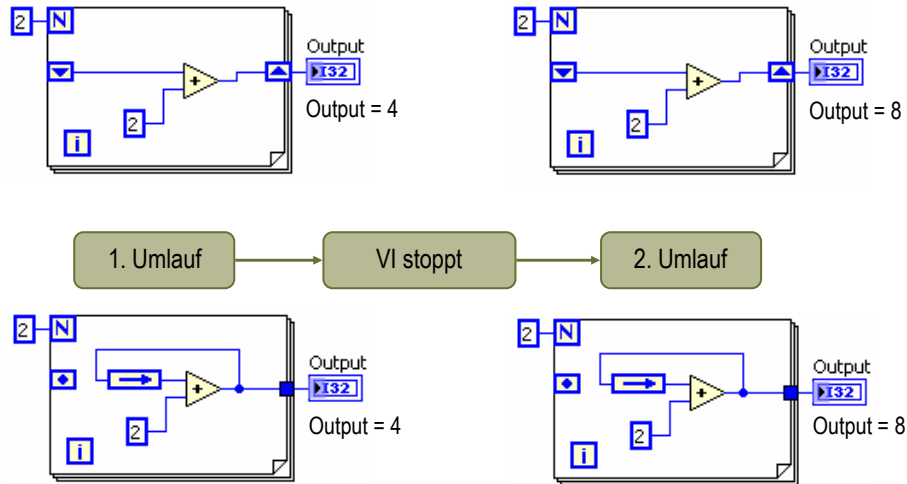
61

Initialisierte Shift Registers & Feedback Nodes



62

Nicht initialisierte Shift Registers & Feedback Nodes



63

Lektion 4 Arrays



Themen

Arrays - Einführung
Auto Indexing Arrays
Array Funktionen
Polymorphie

64

Arrays

- Ansammlung von Daten gleichen Typs
- Ein/mehrdimensional, max. 2^{31} Elemente pro Dimension
- Zugriff auf Elemente über Index; erstes Element hat immer Index 0

index	0	1	2	3	4	5	6	7	8	9
10-Element Array										

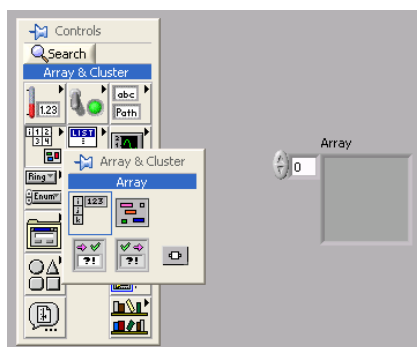
	0	1	2	3	4	5	6
0							
1							
2							
3							
4							

2-dim. Array mit 7 Spalten und 5 Reihen (35 Elemente)

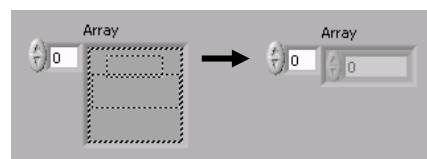
65

Array Controls und Indicators

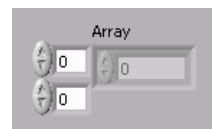
1. Wähle **Array** Shell (Controls Palette)



2. Ziehe Datentyp in Shell



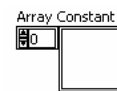
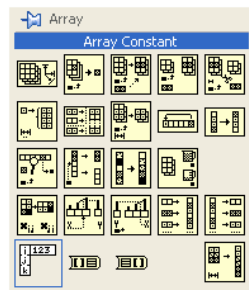
Erweitere ggf.
Dimension
(nD Arrays)



66

Zugriff auf Array Konstanten

1. Wähle **Array** Shell (Functions Palette)



2. Ziehe Datentyp in Shell und initialisiere

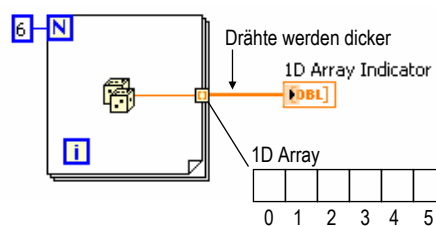


67

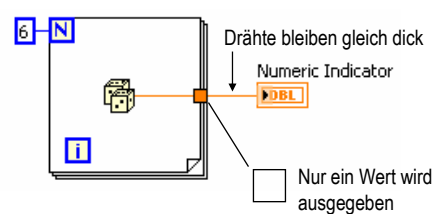
Auto-Indexing

- Mit Autoindexierung können Arrays erzeugt werden (For Loops auto-index standardmäßig)
- While Loops (While Loops geben standardmäßig letzten Wert aus)
- Rechtsklick auf Tunnel ermöglicht Veränderung der Indexingfunktion

Auto-Indexing Enabled

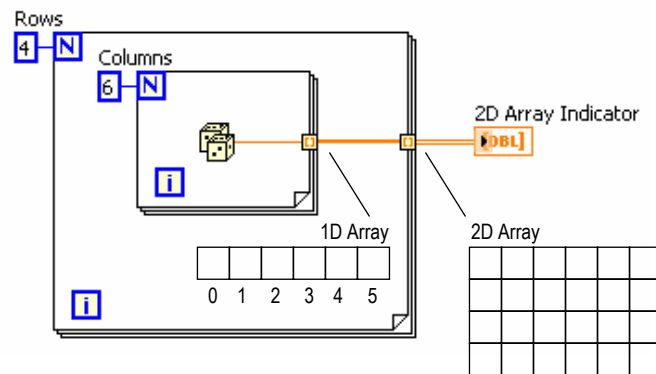


Auto-Indexing Disabled



68

Erzeugen von 2D Arrays

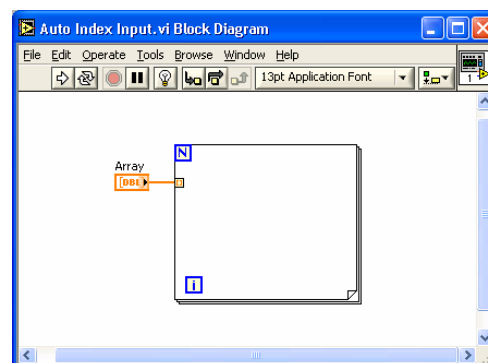


- Innere Schleife erzeugt Spaltenelemente
- Äußere Schleife setzt diese Elemente in Zeilen

69

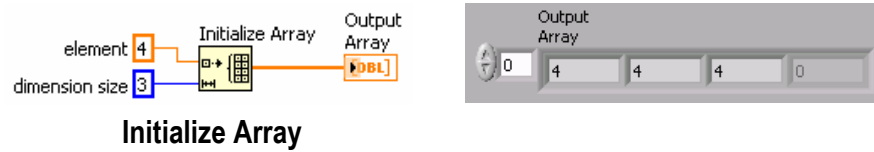
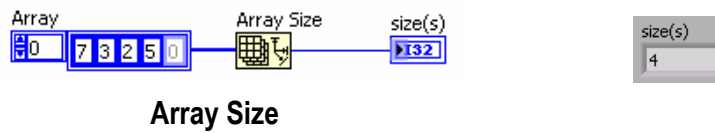
Auto-Index Input

- Array Input kann als Ersatz für die Verwendung des Loop Terminals dienen
- Anzahl der Terminalelemente entspricht Anzahl der Umläufe
- Run Arrow nicht gebrochen



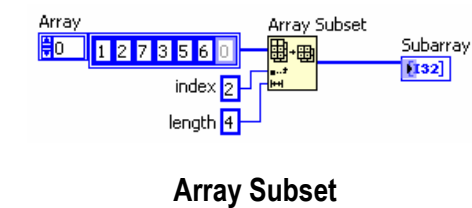
70

Gemeinsame Array Funktionen



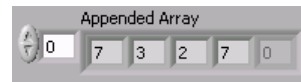
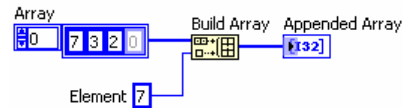
71

Gemeinsame Array Funktionen

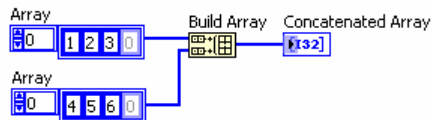


72

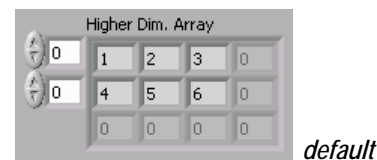
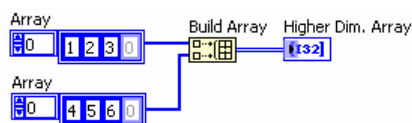
Die Build Array Funktion



Element anhängen (append)



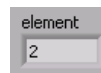
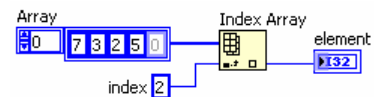
Concatenate Inputs



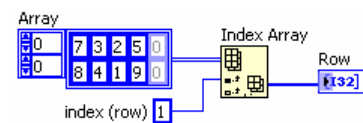
Dimension erhöhen

73

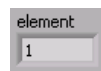
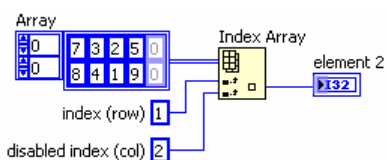
Die Index Array Funktion



Element extrahieren



Reihe extrahieren



Ein Element der Reihe extrahieren

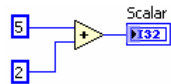
74

Polymorphie

Funktionseingänge können verschieden Datentypen besitzen
Alle arithmetischen Funktionen LabVIEWs sind polymorph

Kombination

Scalar + Scalar

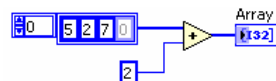


Ergebnis

Scalar



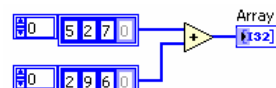
Array + Scalar



Array



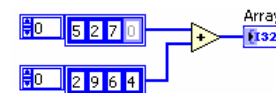
Array + Array



Array



Array + Array



Array



75

Lektion 5 Cluster



TOPICS

Clusters - Einführung
Cluster Funktionen
Error Cluster

76

Cluster

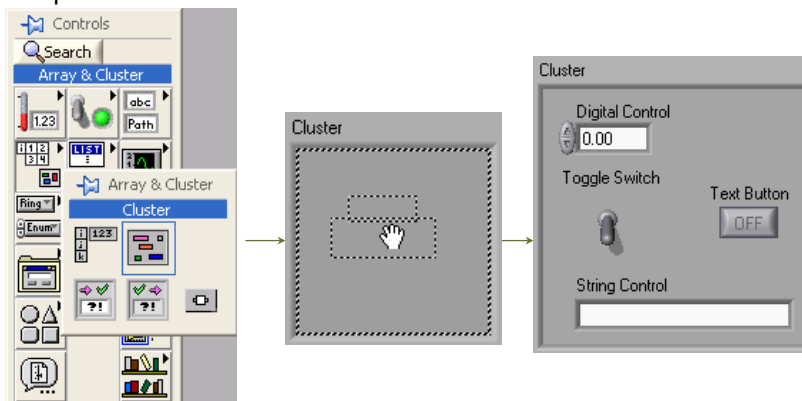
- Datenstrukturen die Daten gruppieren (wie *struct* in C oder *record* in Pascal)
- Elements müssen entweder als Controls oder Indicators angelegt werden
- Ähnlich wie Drähte und Litzen in einem Kabel



77

Cluster Controls und Indicators

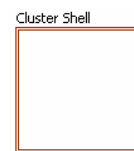
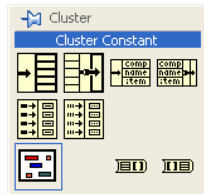
1. Wähle **Cluster** shell aus **Array & Cluster** subpalette
2. Ziehe Objekte in Shell



78

Creating Cluster Constants

1. Wähle **Cluster Constant** shell aus der **Cluster** subpalette



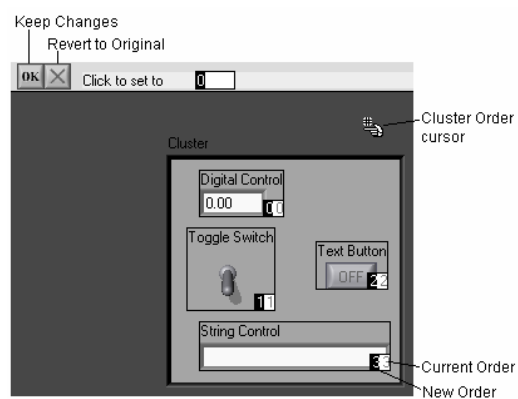
2. Ziehe Objekte in Cluster shell



79

Cluster Order

- Elemente besitzen Logische (Start mit 0)
- Wähle **Reorder Controls in Cluster...** um Reihenfolge zu ändern



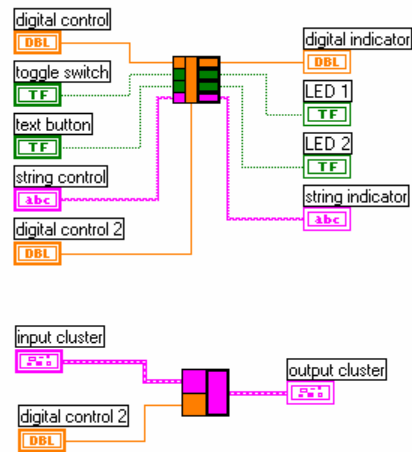
80

Verwendung von Clustern um Daten in SubVIs zu bringen

Cluster geeignet um komplexe Informationen an Terminals zu bringen

Eliminiert 28-Terminal Limit

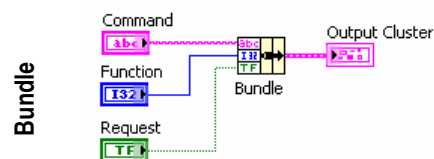
Vereinfacht Verdrahtung



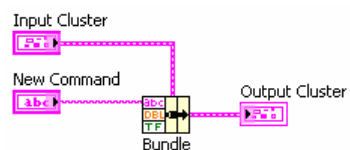
81

Cluster-Funktionen - Bundle

Neuen Cluster erzeugen

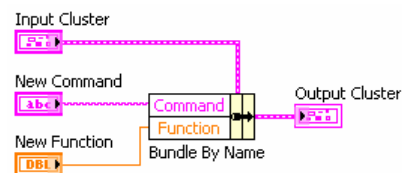


Existierenden Cluster modifizieren



Bundle
By Name

Braucht existierenden Cluster



82

Cluster Funktionen - Unbundle

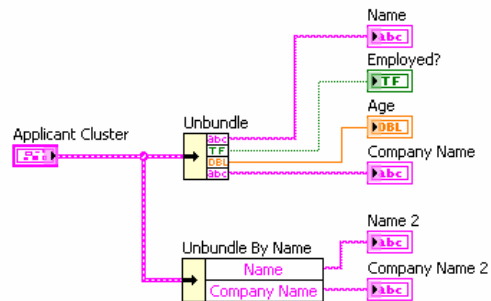
Applicant Cluster

Name Age

Employed?

Yes ☐ No ☐

Company Name



Unbundle



Unbundle By Name



83

Error Cluster

Error In und Error Out Elemente sind essentiell für das Fehlermanagement in VIs

(zu finden in **Controls»Array & Cluster Palette**)

error in (no error)

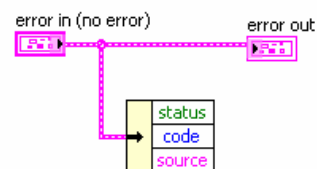
status ☒ code

source

error out

status ☒ code

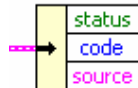
source



84

Error Cluster Details

- **Status** Boolescher Wert "TRUE" bedeutet Fehler
- **Code** 32-bit Integer mit Vorzeichen der die Fehlernummer zurückgibt.



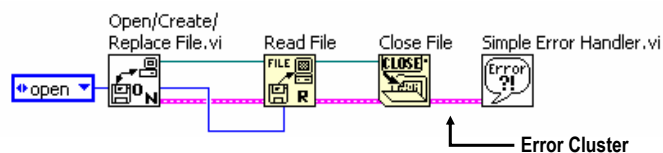
- Code gleich 0 und Status= False Kein Fehler
- Code ungleich 0 und Status= False Warnung
- Code ungleich 0 und Status= True Fehler

- **Source** String mit Angabe über Fehler und Fehlerort.

85

Error Handling with Clusters

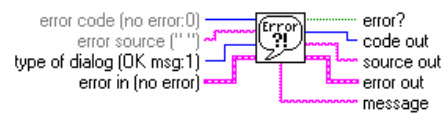
- LabVIEW handelt Fehler nicht automatisch
- LabVIEW Fehlerbehandlung funktioniert nach Datenflussprinzipien)
- Daisy Chain Architektur (Cluster wird durch alle SubVIs geführt; VIs die einen Fehler am Eingang haben werden nicht ausgeführt!)



86

Simple Error Handler

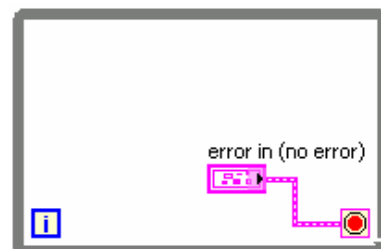
Simple Error Handler ist geeignet für Fehlerbehandlung nach dem Datenfluss (in **Functions»All Functions»Time and Dialog** palette)



87

Verwendung von While Loops für das Error Handling

Error Cluster direkt mit Abbruchbedingung verdrahtbar (Schleife bricht bei Fehler=wahr ab – aut. Selektion des Booleans "Error")



88

Lektion 6

Daten plotten



Themen

Waveform Charts

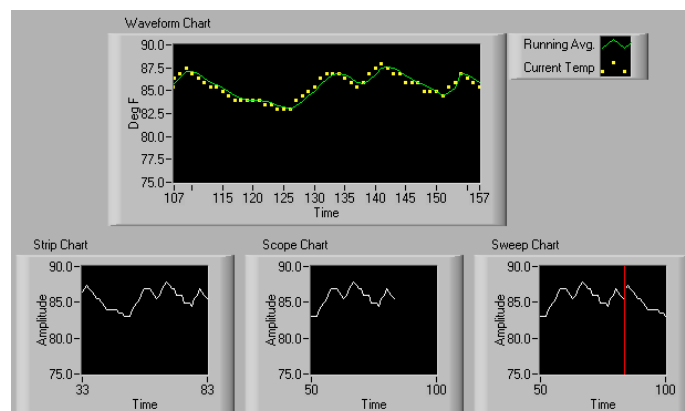
Waveform und XY Graphs

Intensity Graphs

89

Waveform Charts

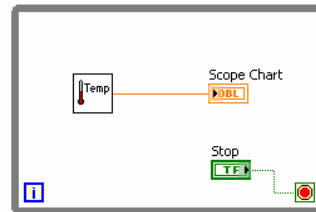
Zu wählen in **Controls»Graphs and Charts Palette**



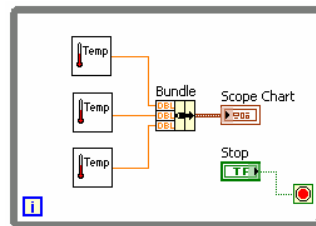
90

Charts verdrahten

Single-Plot Chart

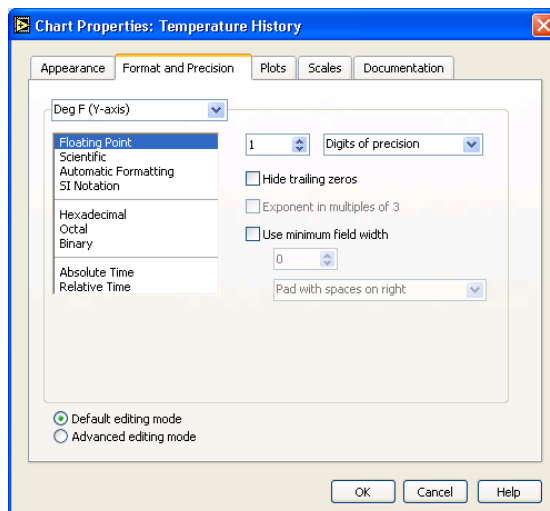


Multiple-Plot Chart



91

Chart Properties anpassen



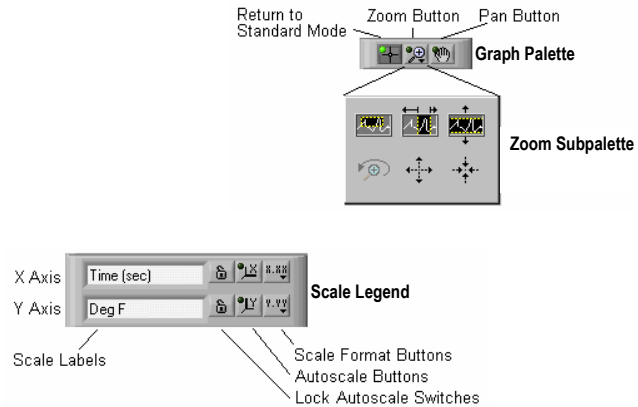
- Erscheinungsbild ändern
- Format und Auflösungen der Achsen verändern
- Plottyp wählen
- Skalen ändern
- Dokumentation/Tip Strip
- Etc.

92

Charts und Graphs anpassen

Visible Items anwählen ermöglicht Ein- und Ausblenden der Elemente:

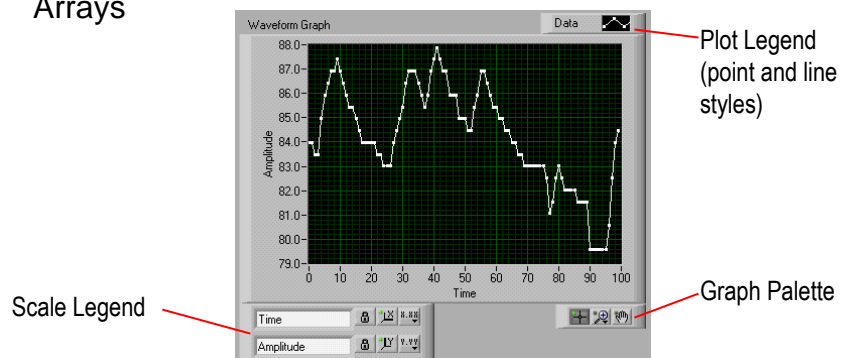
- Plot Legend
- Digital Display
- Scrollbar
- X and Y Scale
- Graph Palette
- Scale Legend



93

Graphs

- Wählbar aus Graph subpalette
- Waveform Graph – Plot eines Arrays in Abh. von Einzelindizes
- XY Graph – Plot eines Arrays in Abh. von eines zweiten Arrays

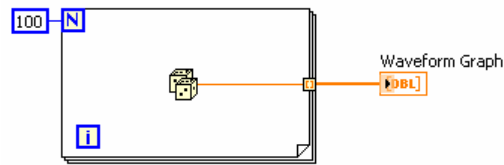


94

Single-Plot Waveform Graphs

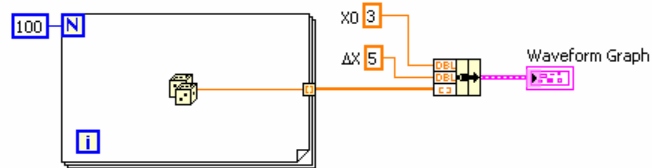
$X_0 = 0.0$

Delta X = 1.0



X_0 und

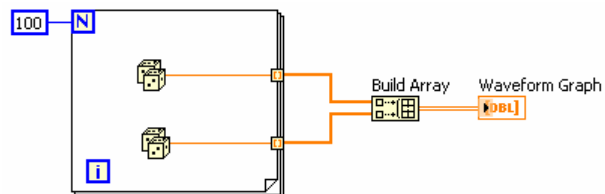
Delta X wählbar



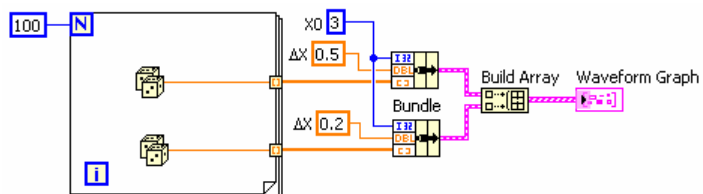
95

Multiple-Plot Waveform Graphs

Jede Reihe
erzeugt eigenen
Plot:
Initial X = 0
Delta X = 1



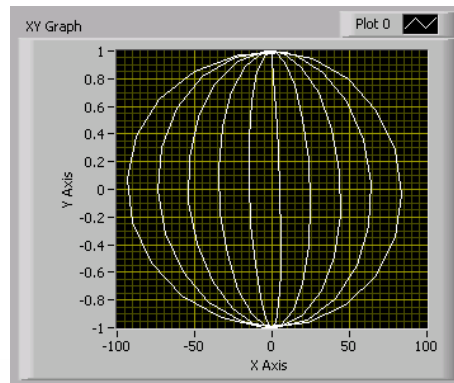
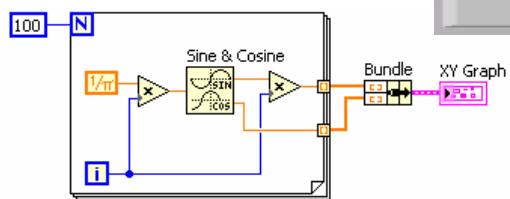
Jede Reihe
erzeugt eigenen
Plot mit
unterschiedlichen
Startwerten und
Schrittweiten



96

XY Graphen

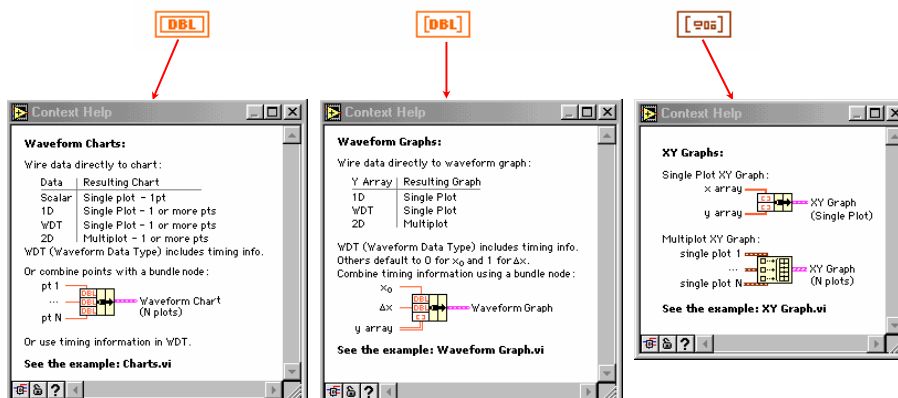
- Zwei unabhängige Arrays erzeugen Datenpunkte
- Relationen möglich (mehr als ein Wert pro X-Wert)



97

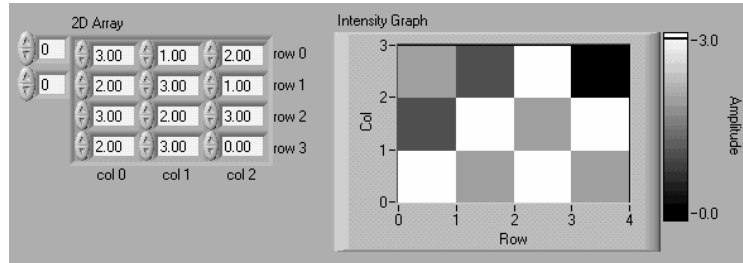
Chart und Graph Anwendung - Sonstiges

Kontexthilfe ist sehr nützlich in Zusammenhang mit Charts und Graphs



98

Intensity Plots und Graphs



- Useful in displaying terrain, temperature patterns, spectrum analysis, and image processing
- Data type is a 2D array of numbers; each number represents a color
- Use these options to set and display color mapping scheme
- Cursor also adds a third dimension

99

Lektion 7 Entscheidungsfindung in VIs



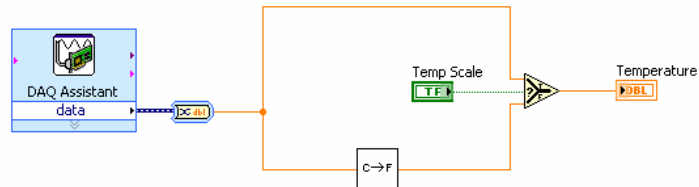
Themen

Die Select Funktion
Case Strukturen
Formelknoten

100

Einfache Entscheidung: Die Select Funktion

- Wenn Temp Scale TRUE ist, wird der obere Zweig durchgeleitet
- Wenn Temp Scale FALSE ist, wird der untere Zweig durchgeleitet



- Bei komplexeren Operationen mit mehr als zwei Optionen ist die CASE Struktur zu verwenden

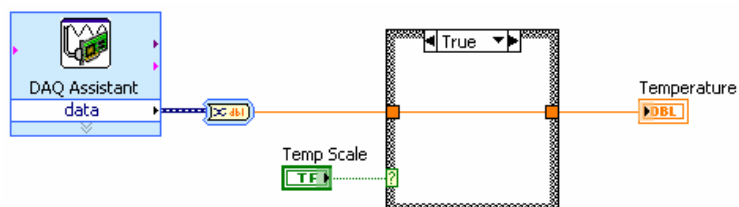
101

Case Structures

Boolean Case Structure Beispiel:

Wenn Temp Scale TRUE ist, wird die TRUE Case Instanz abgearbeitet

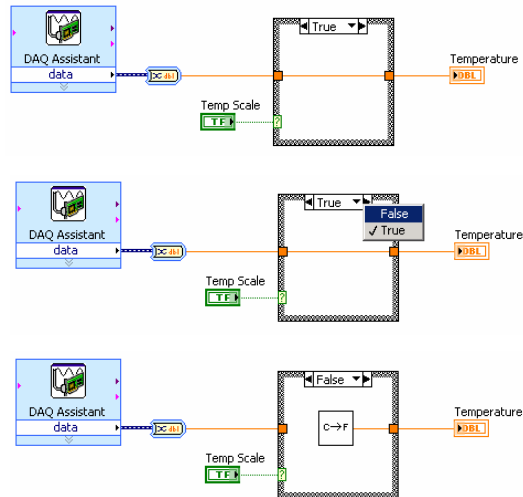
Wenn Temp Scale FALSE ist, wird die FALSE Case Instanz abgearbeitet



102

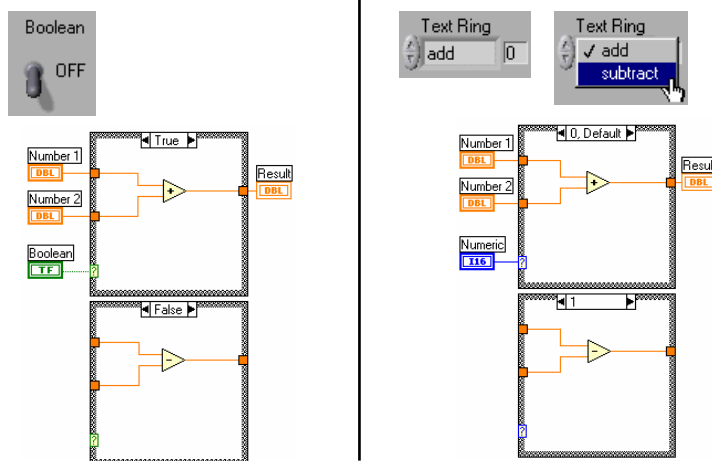
Case Structures

- In der Structuren subpalette der Funktionspalette
- UmschlieÙe Knoten oder ziehe sie in Struktur
- Wie Karten gestapelt – eine Instanz pro Fall



103

Boolean und Numeric Cases



Verdarhte alle möglichen Ausgänge der Case Struktur

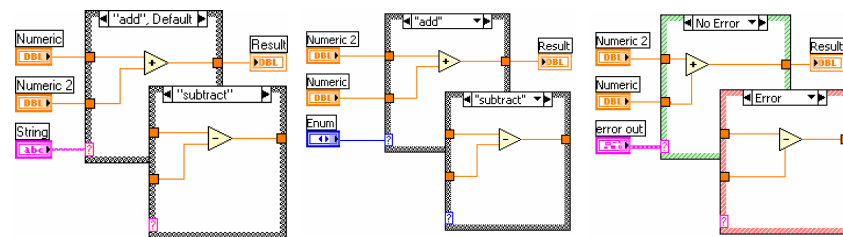
104

String, Enum und Error Cases

String Case

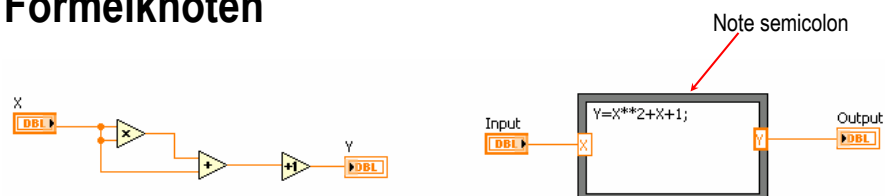
Enum Case

Error Case



105

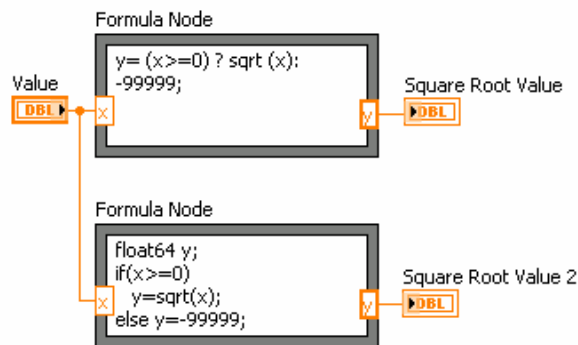
Formelknoten



- In der Struktursubpalette wählbar
- Möglichkeit zur Erzeugung komplexer Gleichungen
- Variablen werden am Rand erzeugt
- Variablennamen sind casesensitiv (Groß- und Kleinschreibung beachten)
- Semicolon (;) erforderlich als Abschluss
- Context Help Fester zeigt Optionen an

106

Entscheidungsfindung mit Formelknoten



Zwei unterschiedliche Wege, die zum Ziel führen

107

Lektion 8 Strings und File I/O



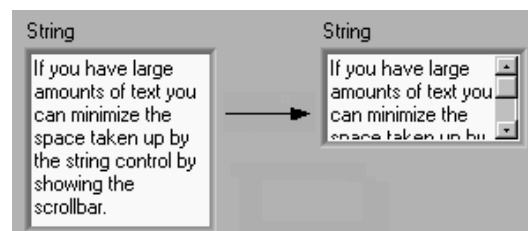
Themen

- Wie man String Controls und –Indicators erzeugt
- Bei String Funktionen an Hand von Einzelbeispielen
- Wie mach man Dateiewin- und –ausgabe?
- Wie benutzt man High- und Low Level VIs
- Wie formatiert man Text um ihn dann in Tabellendateien verwenden zu können

108

Strings

- Sequenz aus (nicht) anzeigbaren Zeichen (ASCII)
- Hinweise anzeigen, Kommunikation mit Messgeräten und Dateien, etc.
- String control/indicator befinden sich in **Controls»String** Subpalette



109

String Display Modi

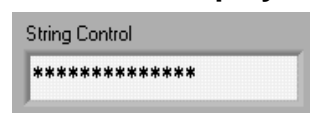
Normal display



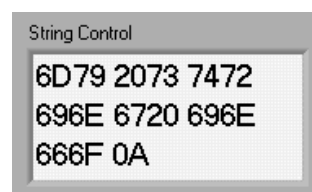
\ code display



Password display



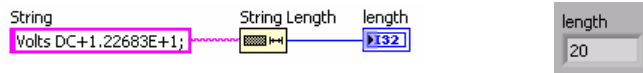
Hex display



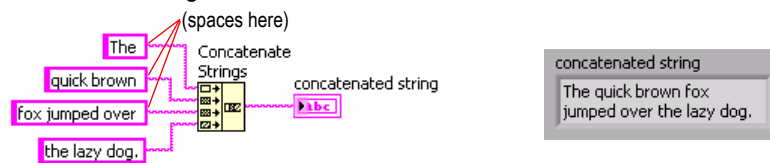
110

String Functions

String Length



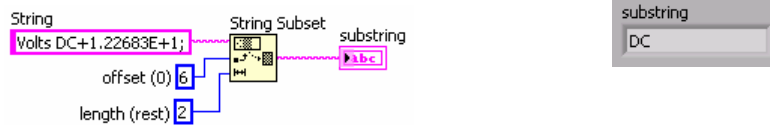
Concatenate Strings



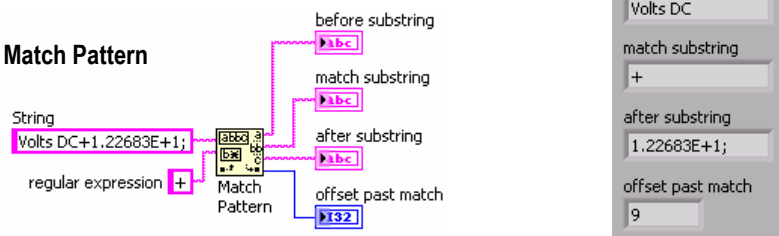
111

String Functions

String Subset

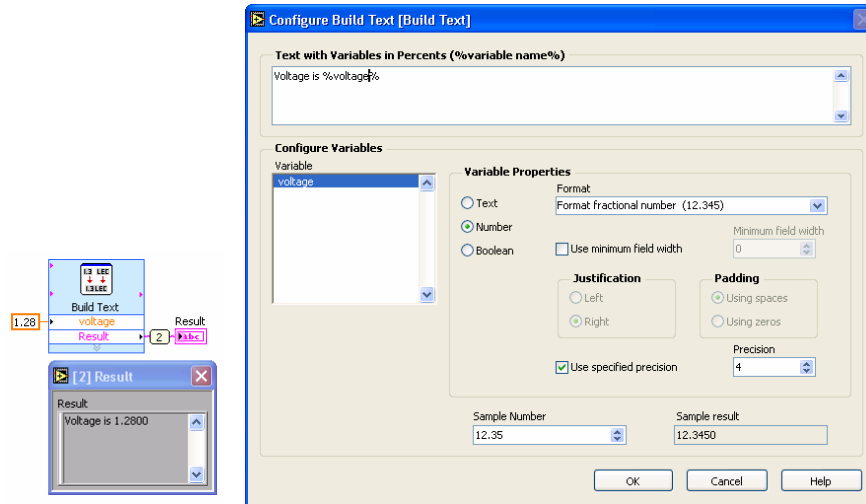


Match Pattern



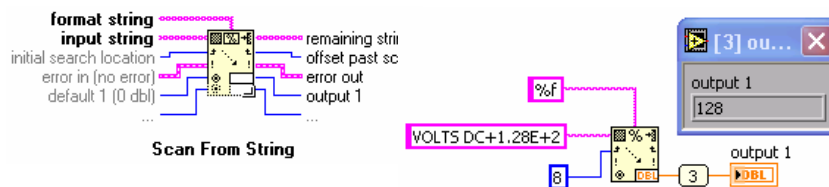
112

Wandeln von Numerics in Strings: Build String



113

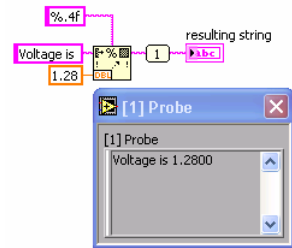
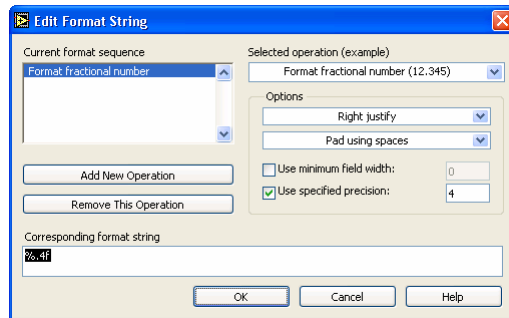
Wandeln von Strings in Numerics: Scan From String



114

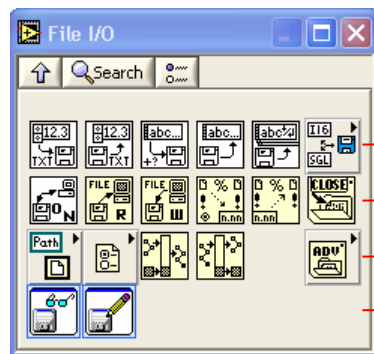
Edit Format String

Scan From String Function



115

File Input und Output



Vier Hierarchieebenen:

High-level File VIs

Intermediate File VIs and Functions

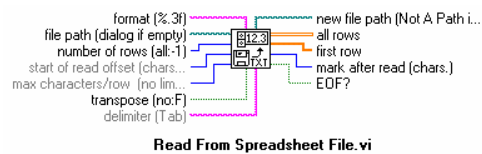
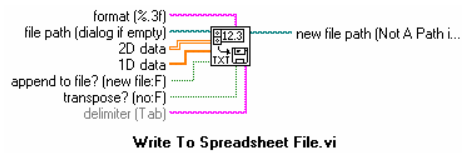
Advanced File Functions subpalette

Express VIs

116

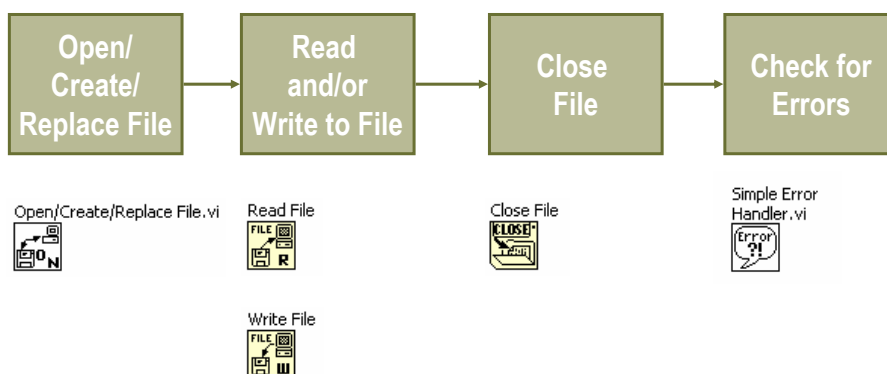
High-level File I/O VIs

- Write to Spreadsheet File
- Read from Spreadsheet File
- Write Characters to File
- Read Characters from File
- Read Lines from File



117

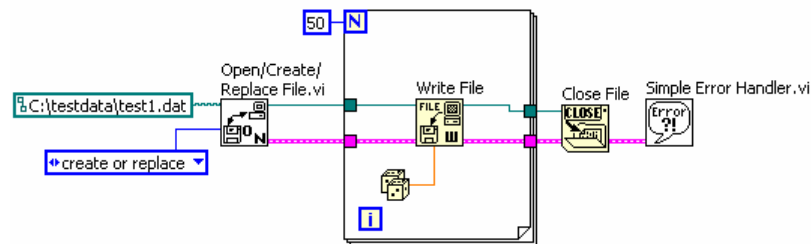
File I/O Programmiermodell - Intermediate



118

Daten in Dateien schreiben

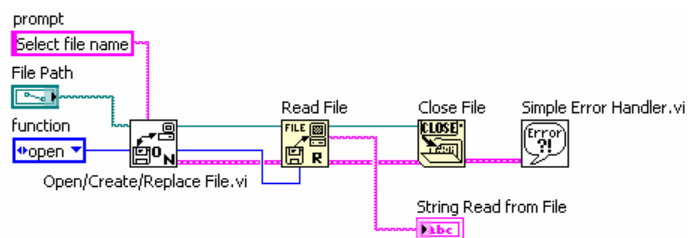
- Open/Create/Replace öffnet file TEST1 . DAT und erzeugt refnum und error cluster
- Write File schreibt Daten
- Close File schließt Datei
- Simple Error Handler überprüft ob Fehler erzeugt wurden



119

Daten aus Dateien lesen

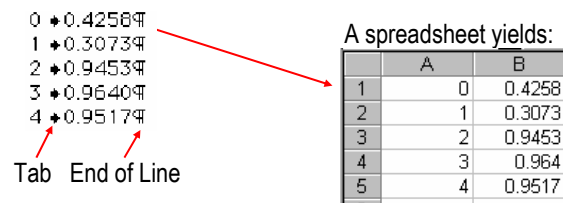
- Open/Create/Replace öffnet file in Abhängigkeit von der Benutzerauswahl und erzeugt refnum und error cluster
- Read File liest Daten
- Close File schließt Datei
- Simple Error Handler überprüft ob Fehler erzeugt wurden



120

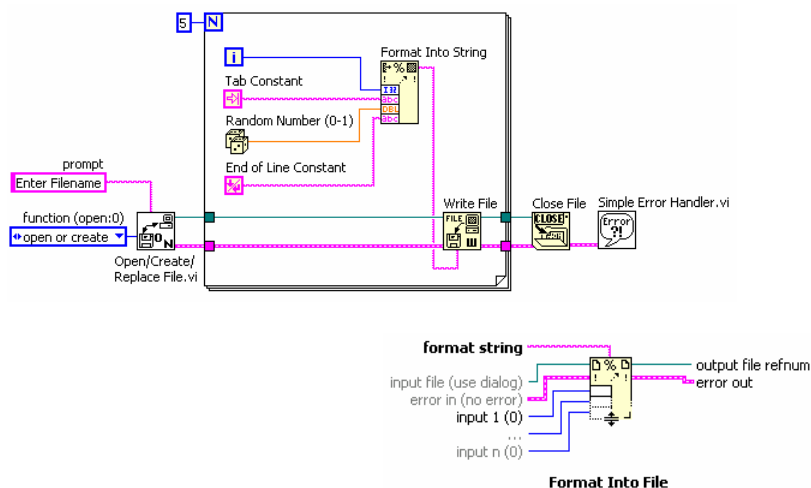
Spreadsheet Strings editieren

- Spreadsheets sind populäre Tools für das Datenhandling und die -analyse
- Die *tab-delimited spreadsheets* sind die gebräuchlichsten einer Vielzahl von Tabledateienformaten
 - Spalten sind durch Tabs getrennt
 - Zeilen sind durch End of Line Zeichen getrennt



121

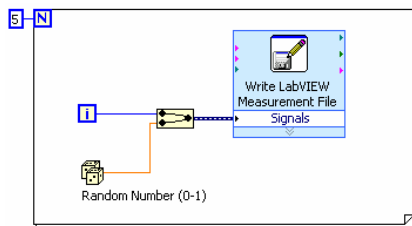
Erzeugen von Spreadsheet Files



122

Write LabVIEW Measurement File

- Besitzt open, write, close und error handling Funktionen
- Tab- or Komma-Trennzeichen verwendbar
- Merge Signals function wird benutzt um Dynamische Datentypen zu erzeugen



	A	B	C	D
1		0	0.385055	
2		1	0.23516	
3		2	0.985184	
4		3	0.177893	
5		4	0.935815	
6				
7				

123

Lektion 9

Datenerfassung und Waveforms

Themen

Plug-in DAQ Geräte
Datenerfassung in LabVIEW
Analog Input
Data Logging
Analog Output
Counter
Digital I/O



124

Übersicht und Konfiguration

Aufgabe der DAQ System ist die Umsetzung von "Real-World"-Signalen in maschinenlesbare Werte

DAQ Systeme bestehen aus:

- Sensoren
- Signalkonditionierung
- Plug-in DAQ Geräte
- Treiber
- Software

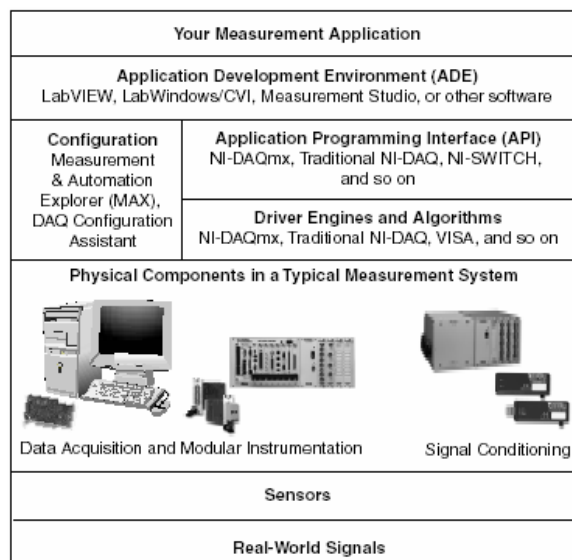


125

Measurement Software Framework

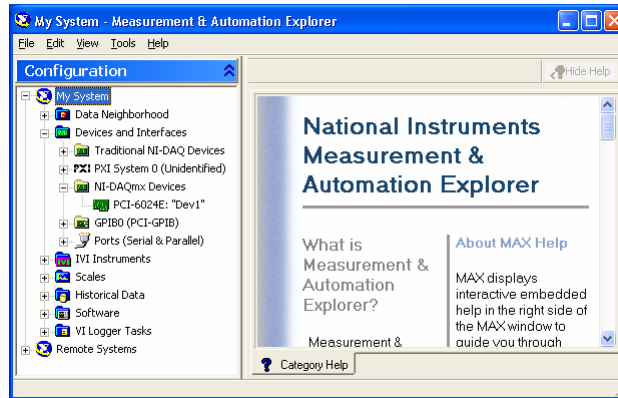
NI-DAQ enthält

- Traditional NI-DAQ
- NI-DAQmx



126

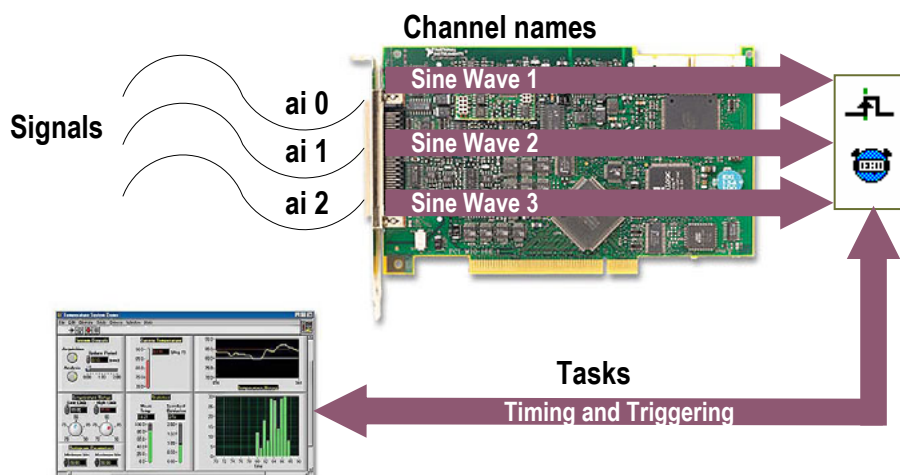
DAQ Hardware Konfiguration



Measurement & Automation Explorer (MAX)

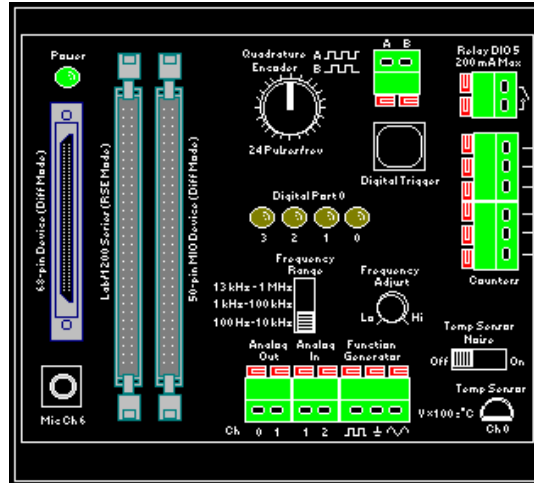
127

Channels und Tasks



128

The DAQ Signal Accessory



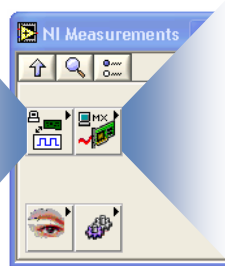
129

Datenerfassung in LabVIEW

Traditional NI-DAQ

Spezifische VIs für

- Analog Input
- Analog Output
- Digital I/O
- Counter operations



NI-DAQmx

Next Generation
Treiber:

- VIs um Tasks abzuarbeiten
- Set von VIs für alle Aufgaben

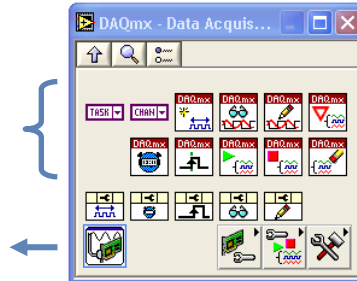
130

NI-DAQmx Datenerfassung

Satz an VIs die
Analog I/O, Digital I/O und Counter
Operationen erlauben

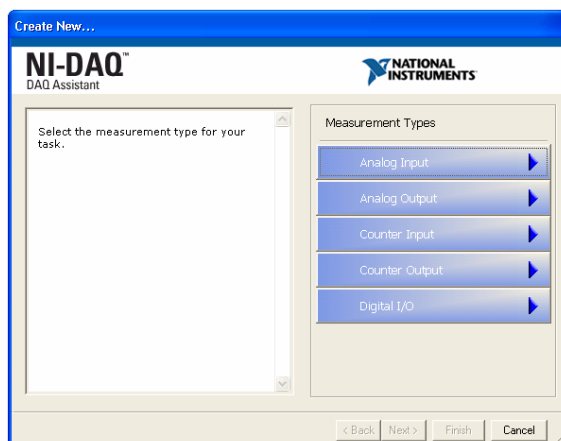
DAQ Assistant Express VI

- Einfache Programmerstellung
- Erzeugt lokalen Task
- Die meisten Applikationen können DAQ Assistant Express VI verwenden



131

NI-DAQmx Data Acquisition Task Types

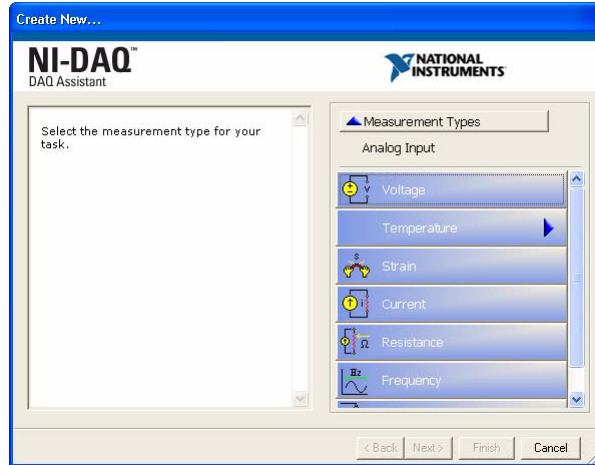


Measurement types:

- Analog Input
- Analog Output
- Counter Input
- Counter Output
- Digital I/O

132

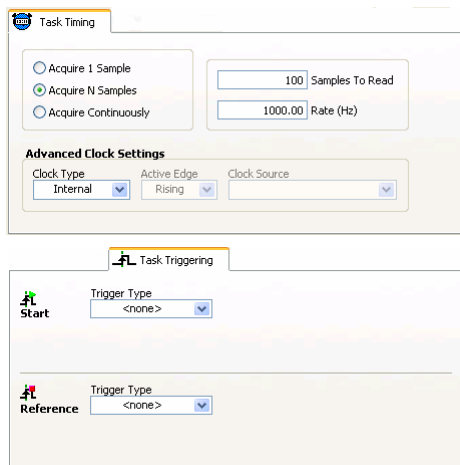
Analog Input



Analog Input task ist
messgerätespezifisch

133

Analog Input Task Timing and Triggering



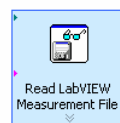
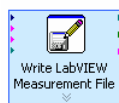
Konfiguriert Abtastrate und
Anz. An Abtastungen

Konfiguriert Start- und
Referenztrigger

134

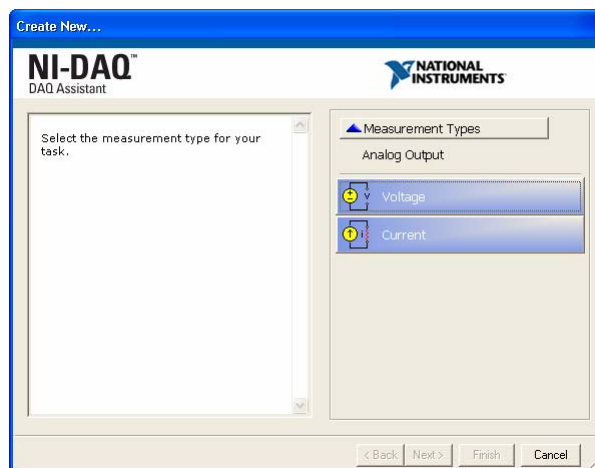
Data Logging

- Oft ist permanente Speicherung von Messdaten erforderlich
- LabVIEW kann Measurement Files lesen und schreiben
- LabVIEW Measurement Files sind ASCII Textdateien



7.35

Analog Output



7.36

Analog Output Task Timing and Triggering

Konfiguriert Abtastrate und Anz. An Abtastungen

Konfiguriert Start- und Referenztrigger

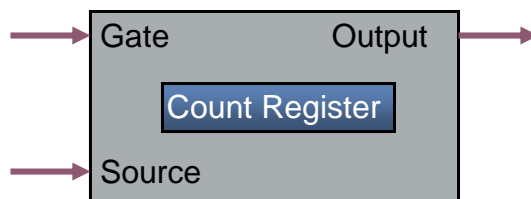
137

Counters

Ein Counter (Zähler) ist eine digitales Timinggerät

Counteranwendungen:

- Event counting
- Frequency measurement
- Period measurement
- Position measurement
- Pulse generation



Count register – Speichert den aktuellen Zählerstand

Source – Eingang der den Counter zum Weiterzählen bringt beim Toggeln am

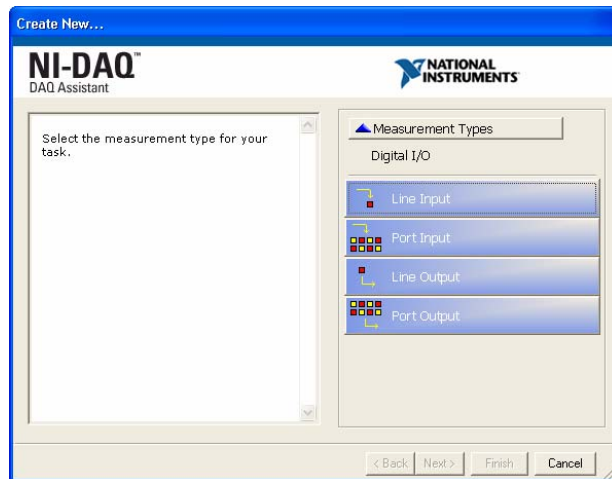
Gate – Eingang mit Enable- und Disable-Funktion

Output – Signal, das Pulse oder Muster ausgibt

138

Digital Input und Output

- Digital I/O können Leitungen und Ports lesen und schreiben
- Ein digitaler Port besteht aus mehreren Leitungen



139

Lektion 10 Instrument Control



Themen

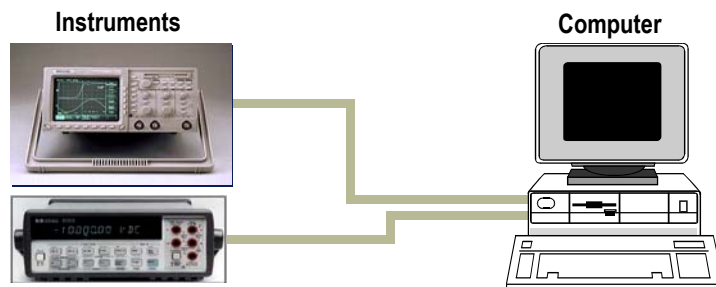
Instrument Control - Übersicht
GPIB Communication und Configuration
Instrument I/O Assistant
Virtual Instrument Software Architecture (VISA)
Instrument Drivers
Serial Port Communication
Waveform Transfers

140

Instrument Control - Überblick

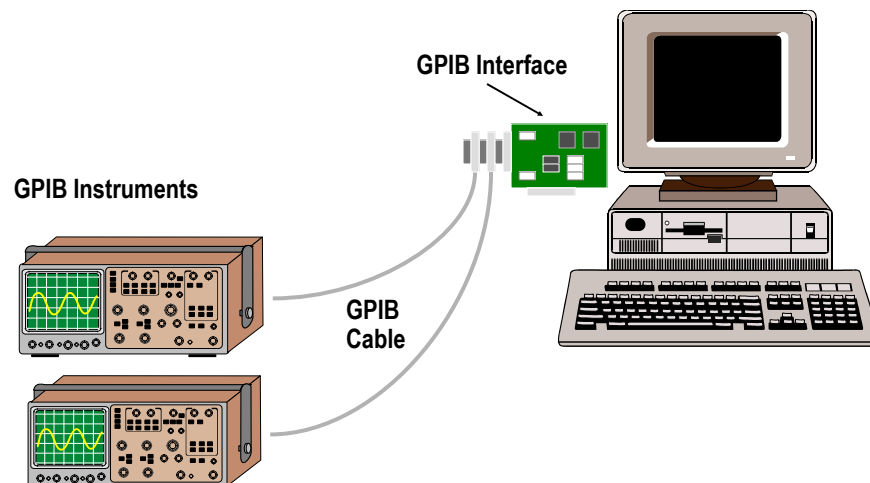
Wenn folgendes bekannt ist kann jedes Gerät angesprochen werden:

- Steckertyp
- Kabeltyp
- Elektrische Eigenschaften des Gerätes
- Kommunikationsprotokolle
- Software Treiber verfügbar

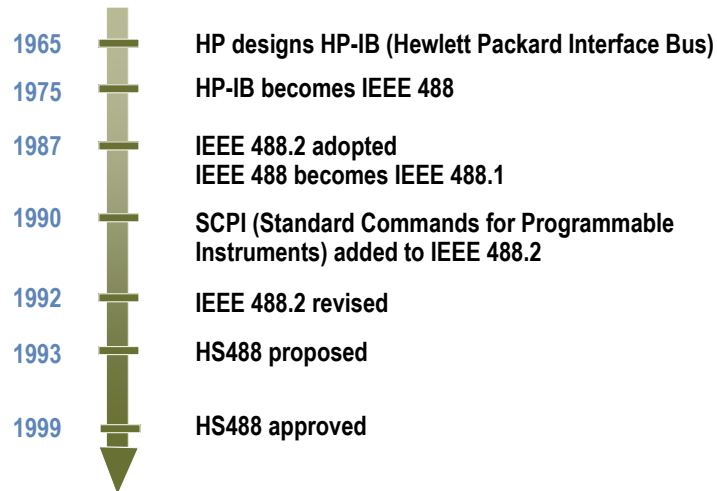


141

GPIB Kommunikation

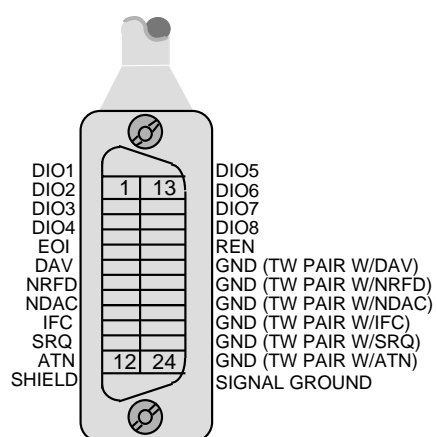


GPIB Standardisierung



143

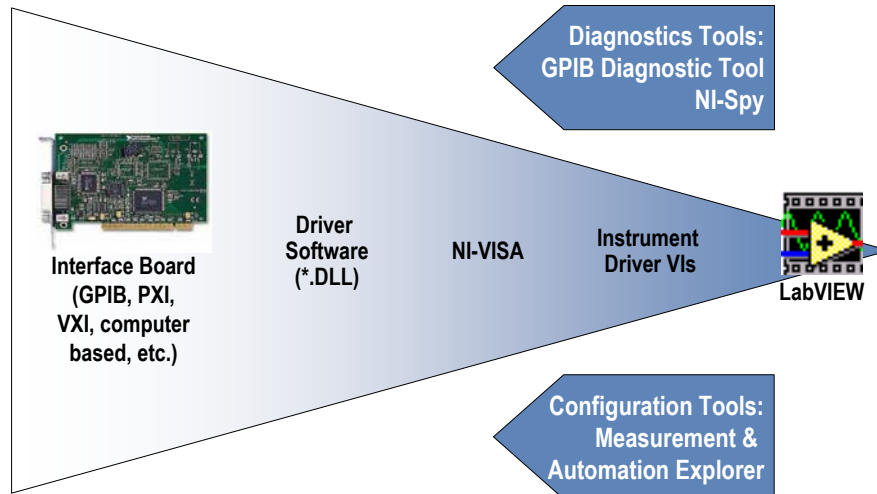
GPIB Hardware Spezifikationen



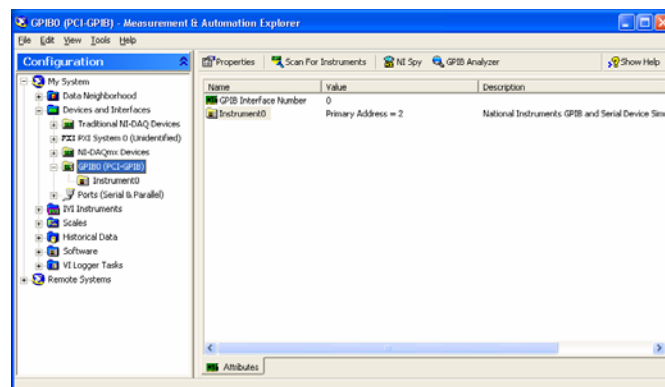
- Max. Kabellänge zw. Geräten = 4 m (2 m Durchschnitt)
- Max. Kabellänge = 20 m
- Max. Anz. An Geräten = 15 (max. 2/3 angeschaltet)

144

GPIB Software Architecture — Windows



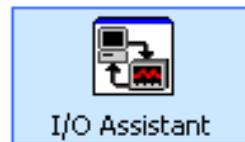
Konfiguration von GPIB Boards und Instrumenten



Measurement & Automation Explorer (MAX)

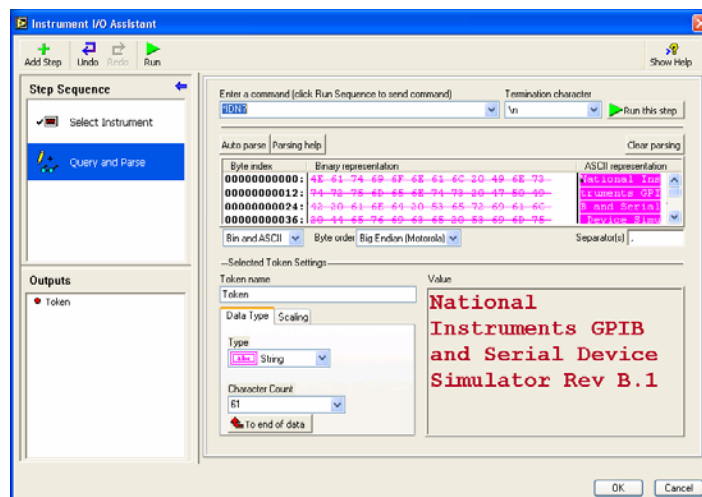
Was versteht man unter dem Instrument I/O Assistant?

- Zugriff mittels LabVIEW Express VI
- Geräte Sets Up
- Kommunikation und schrittweises Datenparsing durch Kommunikationsinterface



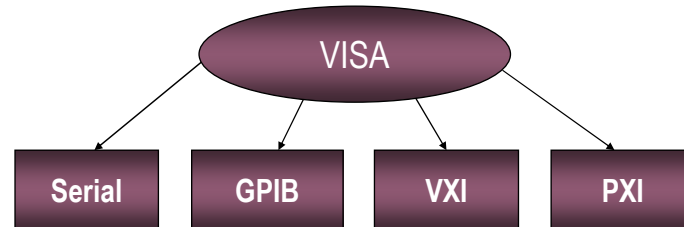
147

Kommunikation mit einem Instrument



148

Virtual Instrument Software Architecture



- Plattformunabhängig
- VISA ist das Rückgrat der IVI and Plug & Play Instrument Drivers
- Interfaceunabhängig
- SCPI command set muss bekannt sein um mit VISA zu interagieren

149

VISA Terminology

- **Resource**—Instrument, Serial Port oder Parallel Port
- **Session**— Verbindung/Handle zur Ressource
- **Instrument Descriptor**—Resource location
 - Format: Interface Type::Address::INSTR
 - Beispiele:

GPIB0::1::INSTR
GPIB0::4::INSTR
GPIB0::10::INSTR
ASRL1::INSTR
ASRL2::INSTR
ASRL3::INSTR
ASRL10::INSTR

150

Instrument Descriptor Syntax

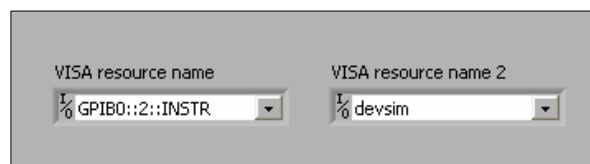
- Resource Name enthält Interface Info
- VISA Aliases funktionieren auch

Interface	Resource Name Grammar
Serial	ASRL[board][::INSTR]
GPIB	GPIB[board]:: <i>primary address</i> ::INSTR
VXI	VXI[board]:: <i>VXI logical address</i> ::INSTR
GPIB-VXI	GPIB-VXI[board][:: <i>GPIB-VXI primary address</i>]:: <i>VXI logical address</i> ::INSTR

151

VISA Resource Name

- Exakter Name und Lage des Instrumentes
- Benutzte VISA Resource Name control

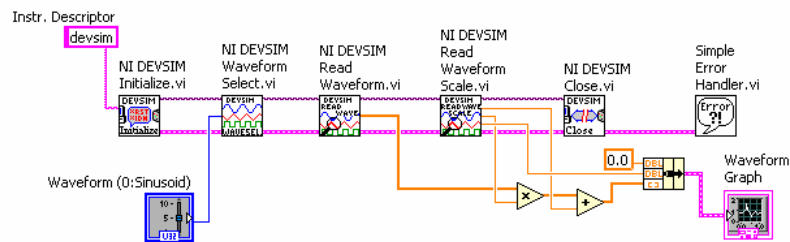


The image shows a graphical user interface for selecting VISA resource names. It contains two labels: "VISA resource name" and "VISA resource name 2". Below each label is a dropdown menu. The first dropdown menu is open, showing the selected value "GPIB0::2::INSTR". The second dropdown menu is also open, showing the selected value "devsim".

152

Instrument Drivers

- Mehr als 1600 LabVIEW Instrumententreiber
- Programmierung durch high-level API stark vereinfacht



153

Installation und Zugriff auf Instrumententreiber

- Treiber herunterladbar: ni.com/idnet
- Treiberbibliotheken sind in das **LabVIEW 8.x\instr.lib** Verzeichnis zu installieren
- Zugriff unter der **Functions»Input»Instrument Drivers** Subpalette



154

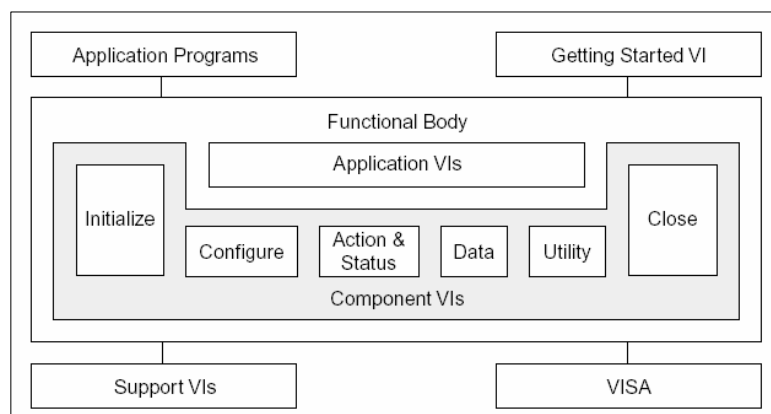
IDNET - Instrument Driver Network



- Infos über Treiber
- Hilfe zum Entwickeln von Treibern
- Möglichkeit zum Hochladen und Herunterladen von Treibern

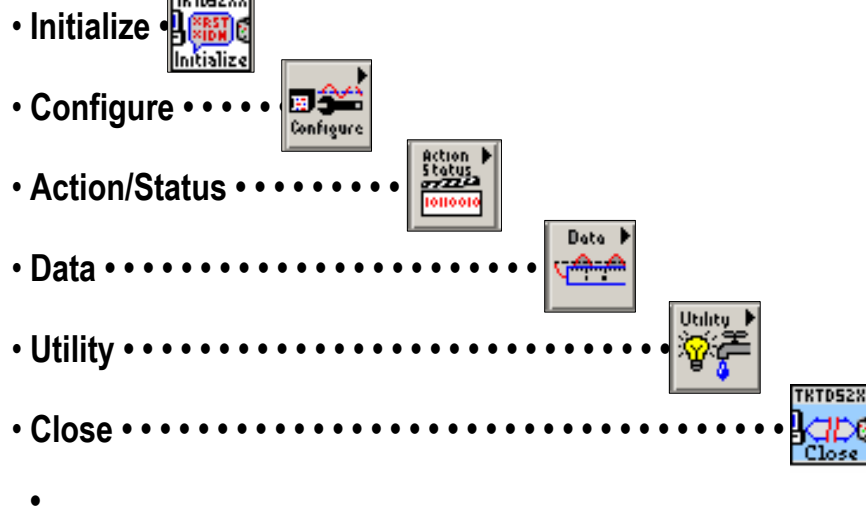
155

Instrument Driver Model



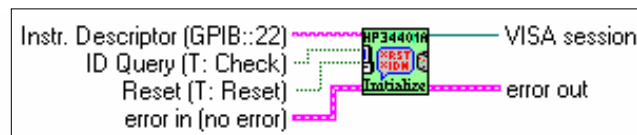
156

Instrument Driver VIs



157

Instrument Driver Inputs und Outputs

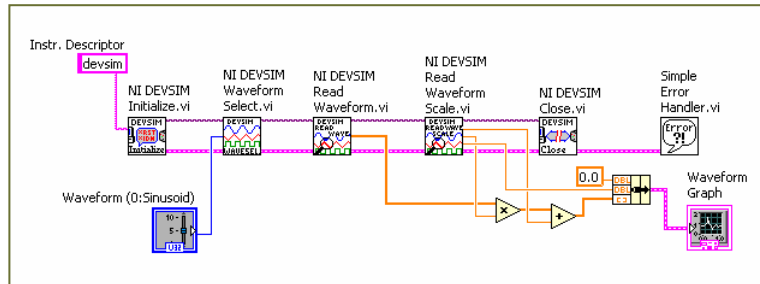


HP34401A Initialize.vi

- Instrument Descriptor
- VISA Sessions
 - Verbindung oder Link zu einem Gerät
 - Erzeugt nach Initialisierung
 - Innerhalb einer VI Instanz zu verwenden
- Error cluster

158

Putting It All Together

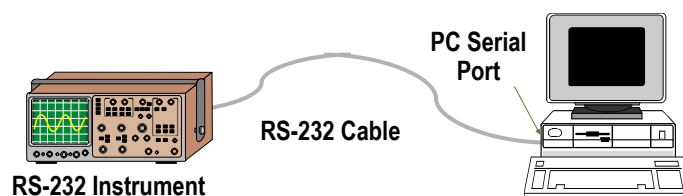


- Konstruktor (Initialize instrument)
- Methoden (Do operation(s))
- Destruktor (Close instrument)
- Check error status

159

Serielle Kommunikation

- Populäre Form der Kommunikation zwischen PC und Peripheriegeräten
- Daten werden bitweise gesendet
- Niedrige Datenraten über lange Distanzen
- Nur ein Kabel wird benötigt, jedoch "Legacy"-Technologie, die auf den modernen Mainboards immer mehr verschwindet

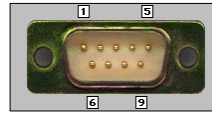


160

Serial Hardware Interfaces

- **RS-232**

- DCE oder DTE
- 9-pin or 25-pin



- **RS-422**

- DCE oder DTE
- 8-pin

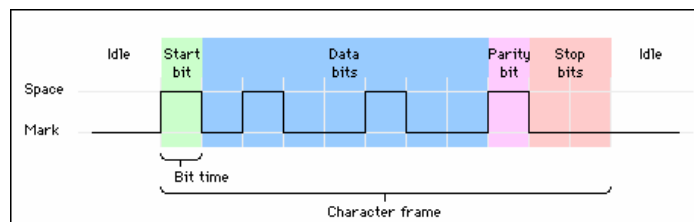
- **RS-485**

- Multidrop

Pin	DTE	DCE
1 DCD	Input	Output
2 RxD	I	O
3 TxD	O	I
4 DTR	O	I
5 Com	-	-
6 DSR	I	O
7 RTS	O	I
8 CTS	I	O
9 RI	I	O

161

Serielle Kommunikation



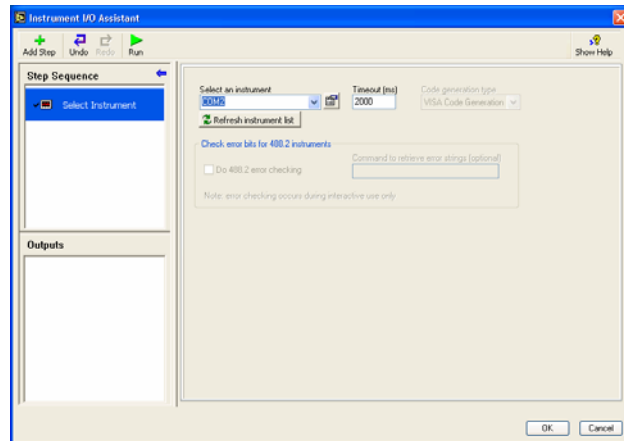
Terminologie

- Baud rate – bits per second
- Data bits – Invertierte Logik mit LSB am Anfang
- Parity – optionales error-checking bit
- Stop bits – 1, 1.5, or 2 invertierte bits am Datenende
- Flow control – Hardware und Software Handshaking möglich

162

Instrument I/O Assistant mit Seriellen Geräten

- Wähle COMX als Geräteadresse
- Benutze den I/O Assistant wie unter GPIB



163

Lektion 11 VI Anpassungen



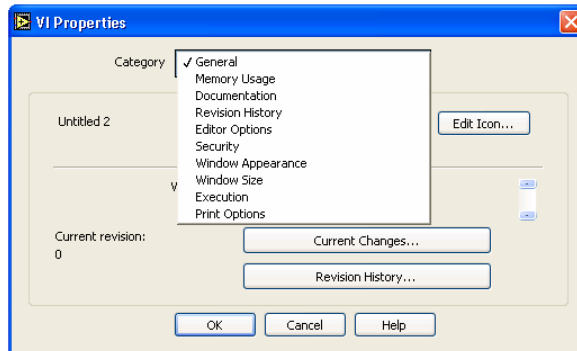
Themen

Front Panel Erscheinungsbild ändern
SubVI Front Panels
Keyboard Shortcuts
VI Properties
Paletten anpassen

164

VI Properties anpassen

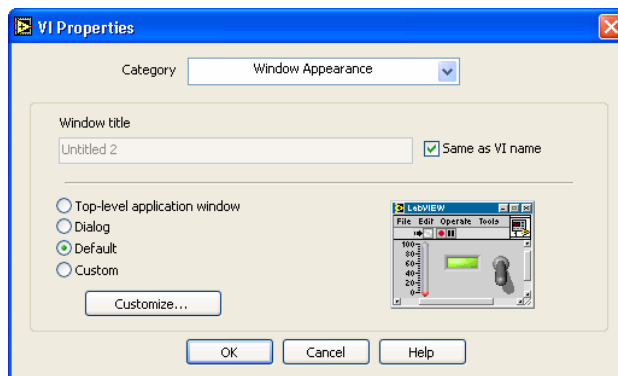
- Rechtsklick auf Icon Pane zum Ändern der VI Properties
- Hat Auswirkungen auf alle Instanzen des VIs in allen Applikationen



165

Window Appearance

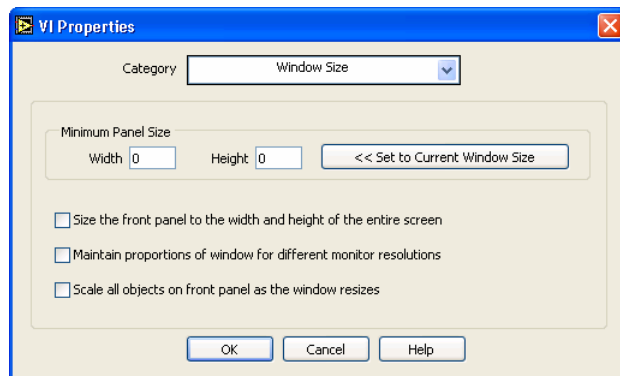
Wirkt sich nur im RUN-Mode aus



166

Window Size

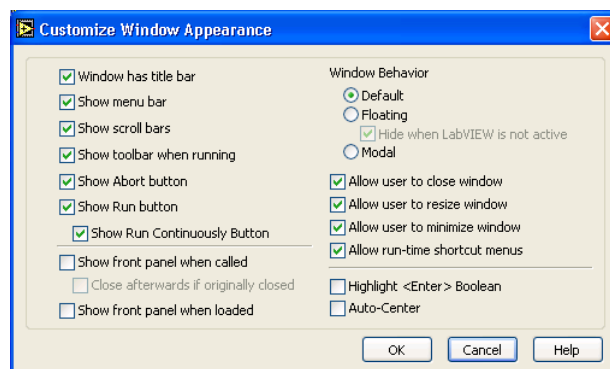
- Setzen der Minimum Panel Größe (wichtig zum Anpassen an Zielmonitor)
- Dynamische Größenanpassung mit Vorsicht zu genießen



167

Pop-Up Panels erzeugen

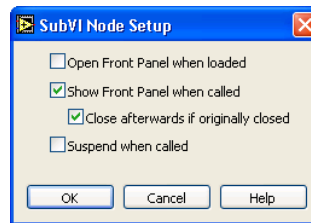
- Benutzung bei Top-Level Application Fenstern oder Dialogarten angeraten
- Kundenspezifisch anpassbar



168

Erzeugung von Pop-Up Panels – Single Instance

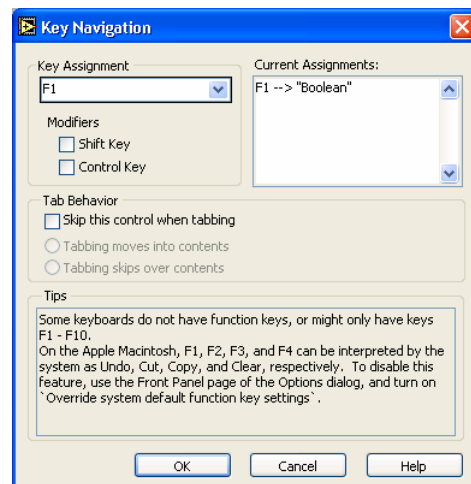
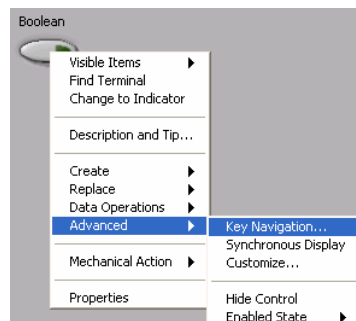
Zugriff auf SubVI Node Setup durch Rechtsklick auf subVI Icon auf aufrufendem VI-Diagramm



169

Key Navigation

Zuweisung von Shortcuts auf Frontpaelemente



170