



BENUTZERHANDBUCH

CISWORKS GmbH & Co. KG

ViPE



CISWORKS GMBH & CO. KG

Benutzerhandbuch

Text, Abbildungen und Gerätedaten wurden mit größter Sorgfalt erarbeitet. Für eventuell verbliebene fehlerhafte Angaben und deren Folgen kann jedoch keine Haftung übernommen werden.

Alle Rechte vorbehalten. Kein Teil dieser Anleitung darf ohne schriftliche Genehmigung von CISWORKS GmbH & Co. KG in irgendeiner Form durch Fotokopie, Mikrofilm oder andere Verfahren reproduziert oder in eine für Maschinen, insbesondere Datenverarbeitungsanlagen, verwendbare Sprache übertragen werden.

Im Text genannte Namen wie LabVIEW, VeriStand, Windows, Microsoft etc. sind eingetragene Warenzeichen und Eigentum ihrer jeweiligen Besitzer und werden im Weiteren nicht besonders gekennzeichnet.

© CISWORKS GmbH & Co. KG
Hans-Liebherr-Str. 18
88161 Lindenberg
Phone 08381 88983-70 • Fax 08381 88983-71

Inhaltsverzeichnis

INHALTSVERZEICHNIS	I
ÜBER DIESES HANDBUCH	1
VIPE	2
Konzept	2
Installation	2
Lizenzierung	3
Voraussetzungen	5
Bildschirmauflösung	5
Projektmanager	6
Sequenzeditor	8
Menü.....	9
Datei	9
Ausführung	9
Projekt	9
Globale Sequenzen	10
Sonstige	11
Hilfe	11
Toolbar.....	11
Überblick.....	11
Variablenmanager	13
TDMS Viewer	13
Stimulus Editor	14
Blockdiagramm	14
Palette.....	15
Buttons	15
Funtionsblöcke	18
Steuerung	18
Bedingung	18
Bedingung Ende	19
Wiederholung.....	19
Wiederholung Ende.....	19
Ausführen Solange	20
Ausführen Solange Ende	20
Stimuligenerierung	21
Rampen	21
Konstante	22
Formel	23
Funktionsgenerator	24
Multitasking	25
Multitask	25

Sync	25
Subtask Ende	26
Multitask Ende	26
Timing	26
Warten	26
Warte bis	26
Warte bis Ende	27
Datenaufzeichnung	28
Datenaufzeichnung	28
Datenaufzeichnung Pause	29
Datenaufzeichnung Fortsetzen	29
Datenaufzeichnung Ende	29
Grenzwerte	30
Grenzwertüberwachung	30
Grenzwertüberwachung Ende	31
Unterprogramme	32
Unterprogramm	32
Stopsequenz	32
Stimulus File	32
Stimulus File	32
Stimulus File Pause	34
Stimulus File Fortsetzen	34
Stimulus File Ende	34
Favoriten	34
System Status	35
Remotesteuerung	36
Grundlegende Funktionsweise	36
Befehlssatz	37
LabVIEW Beispiel	38

Über dieses Handbuch

Dieses Handbuch beschreibt den grafischen Echtzeitsequenzer **CISWORKS ViPE**. Der **CISWORKS ViPE** ist eine Abstrahierungsschicht für NI VeriStand Echtzeitsequenzen. Er ermöglicht Nutzern ohne fundierte Programmierkenntnisse das Erstellen von Echtzeitsequenzen, in dem es für die Syntax eine grafische Darstellung wählt.

KONVENTION Warnhinweis Wichtiger Hinweis

Konventionen – In diesem Handbuch werden zur Visualisierung folgende Konventionen verwendet um auf Hinweise und Gefahren hinzudeuten. Funktionsblöcke werden Kursiv dargestellt (Vgl. *Formel*).

ViPE

Visual Programming Editor

Konzept

Mittels Drag & Drop werden Blöcke aus der Funktionspalette zu einem Flussdiagramm zusammengesetzt. Ein Code-Generator übersetzt das Flussdiagramm in ausführbaren Echtzeit-Code, der auf einem NI VeriStand Target ausgeführt wird. Dies soll Nutzern ohne fundierte Programmierkenntnisse das Erstellen von Testsequenzen ermöglichen.

Während der Ausführung ist für den Bediener ersichtlich an welcher Stelle sich die Sequenz momentan befindet. Es ist auch möglich Sequenzen, welche im NI VeriStand Stimulus Profile Editor erstellt wurden, als Untersequenzen einzubinden und zu parametrieren.

Installation

Die Installation gestaltet sich ähnlich wie bei anderen Windows-Programmen. Starten Sie das Installationsprogramm und folgen Sie den angezeigten Anweisungen.



Der **CISWORKS ViPE** dient lediglich zur Automatisierung eines bestehenden VeriStand-Projekts. Bevor Sie den **CISWORKS ViPE** installieren, muss **NI VeriStand** installiert worden sein. Der **ViPE** funktioniert lediglich in Kombination mit den VeriStand-Versionen 2014 und 2016.

Je nach verwendeter VeriStand-Version muss die entsprechende ViPE-Version installiert werden.

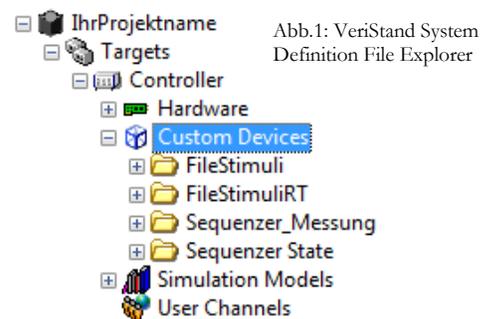
Um mit dem **CISWORKS ViPE** ein VeriStand-Projekt automatisieren zu können, müssen folgende Anforderungen erfüllt sein:

1. Es muss ein **VeriStand-Projekt** angelegt werden.
2. Im VeriStand-Projekt müssen folgende vier **Custom Devices** eingehängt werden: **FileStimuli**, **FileStimuliRT**, **Sequencer_Messung** und **Sequencer State**. Die Custom Devices und die Mapping-Datei sind mit der Installation des **CISWORKS ViPE** mit installiert worden.

Dazu gehen Sie wie folgt vor:

- a. Wählen Sie im VeriStand-Projekt die **System Definition File (*.nivssdf)** aus. Der System Explorer öffnet sich.

- b. Im System Explorer finden Sie unter *IhrProjektname/Targets/Controller* den Punkt **Custom Devices**. Klicken Sie mit der rechten Maustaste auf **Custom Devices** und wählen Sie nacheinander **FileStimuli**, **FileStimuliRT**, **Sequencer_Messung** und **Sequencer State** aus.



3. Das **VeriStand-Projekt (*.nivssdf)** muss laufen bzw. deployed werden.

Lizenzierung

Wenn Sie den **CISWORKS ViPE** zum ersten Mal starten, werden Sie informiert, dass Sie eine 30 Tage gültige Trial-Version benutzen. Bevor dieser Zeitraum abgelaufen ist, müssen Sie den **CISWORKS ViPE** aktivieren um das Programm weiter nutzen zu können. In der Trial-Version steht wie auch in der aktivierten Version der volle Funktionsumfang zur Verfügung.

Um den **CISWORKS ViPE** zu aktivieren müssen Sie zuerst den **CISWORKS License Manager** starten. Dieser wird mit dem **CISWORKS ViPE** mit installiert. Danach sind vier kleine Schritte auszuführen um die Aktivierung abzuschließen.

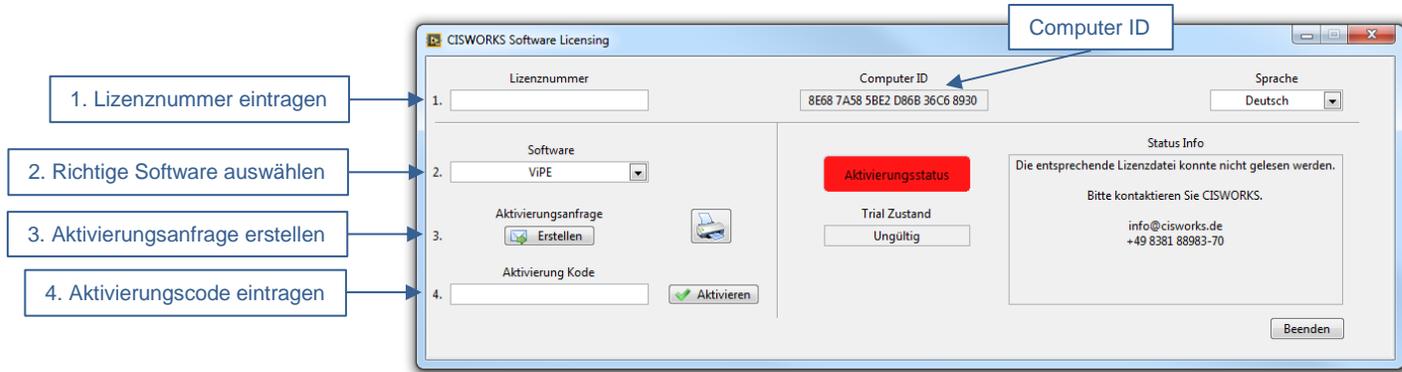
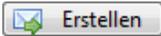
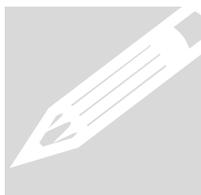


Abb.2: CISWORKS License Manager

1. Mit dem Erwerb des **CISWORKS ViPE** haben sie eine **Lizenznummer** erhalten. Tragen Sie diese in das dafür vorgesehene Feld oben links ein.
2. Wählen Sie unter **Software ViPE** aus.
3. Klicken Sie auf  **Erstellen**
Dadurch wird eine E-Mail in ihrem lokalen E-Mail Dienst erzeugt, welche die Informationen enthält, die wir benötigen um einen Aktivierungscode zu generieren. Senden Sie diese **E-Mail** ab.



Haben Sie keinen lokalen E-Mail Dienst konfiguriert, senden Sie uns einfach eine E-Mail an activate@cisworks.de mit Ihrer Lizenznummer als Betreff und Ihrer Computer ID und „ViPE“ als Inhalt.



Die Computer ID befindet sich oben in der Mitte des **CISWORKS License Managers** und ist einzigartig. Deswegen wird der Aktivierungscode, den wir Ihnen schicken nur auf diesem Computer gültig sein. Wenn Sie das Produkt auf mehreren Rechnern aktivieren möchten, benötigen Sie von jedem dieser Rechner die Computer ID.

4. Den **Aktivierungscode** erhalten Sie innerhalb von zwei Arbeitstagen. Diesen geben Sie unten links ein und klicken dann auf  **Aktivieren**
Bis dahin haben Sie weiterhin die Möglichkeit den **CISWORKS ViPE** als Trial-Version zu nutzen.

Damit haben Sie den **CISWORKS ViPE** erfolgreich aktiviert.

Voraussetzungen

Bildschirmauflösung

Folgende Auflösungen wurden erfolgreich getestet:

- 1920 x 1080
- 1680 x 1050
- 1600 x 900
- 1280 x 1024

Diese Auflösungen können zu Problemen führen:

- 1280 x 800
- 1280 x 720
- 1024 x 768
- 800 x 600
- 640 x 480

Projektmanager

Nach dem Start des **CISWORKS ViPE** erscheint der Projektmanager. Dieser dient der Verwaltung von Projekten und dem Verknüpfen von **CISWORKS ViPE** und NI VeriStand.

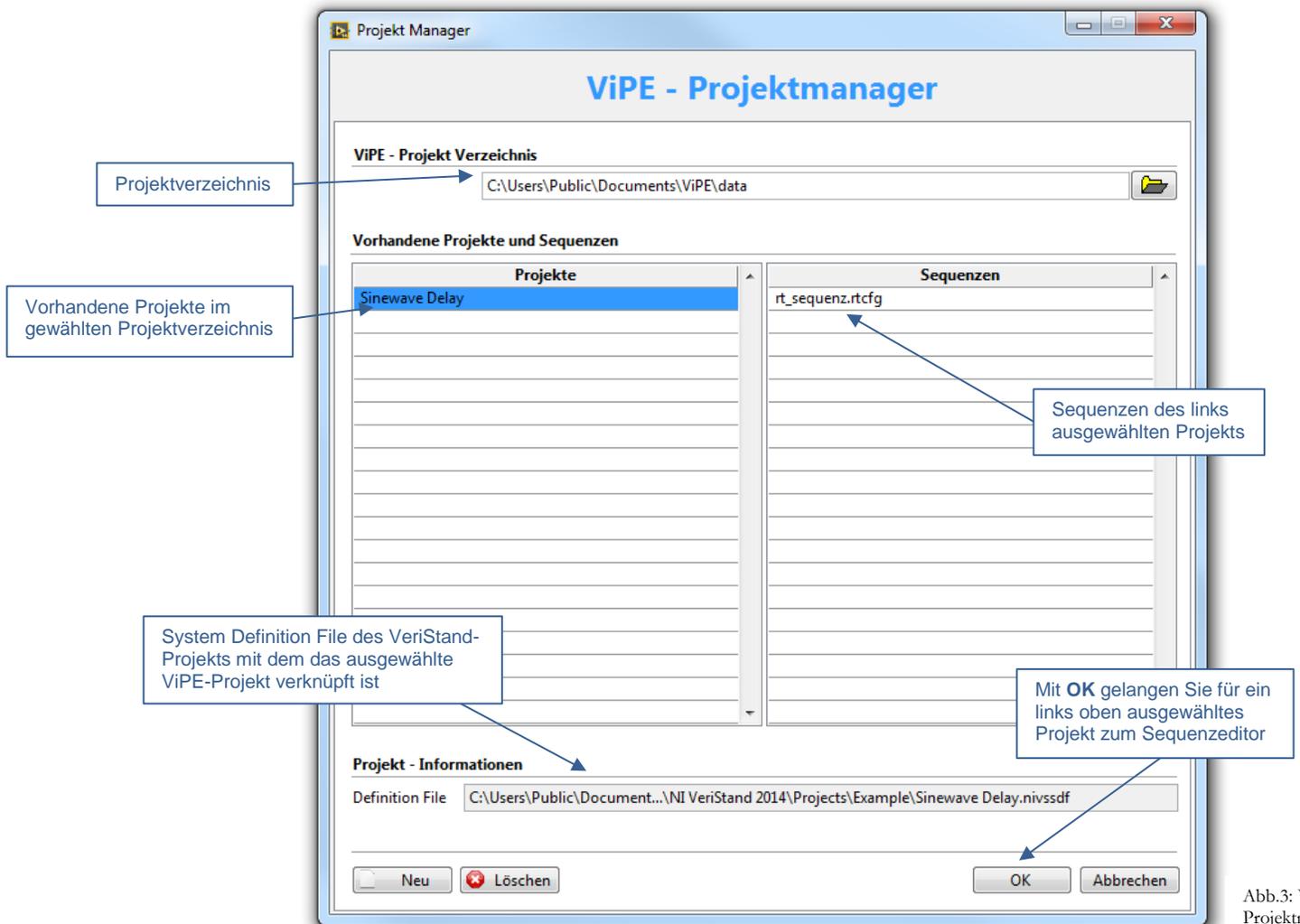
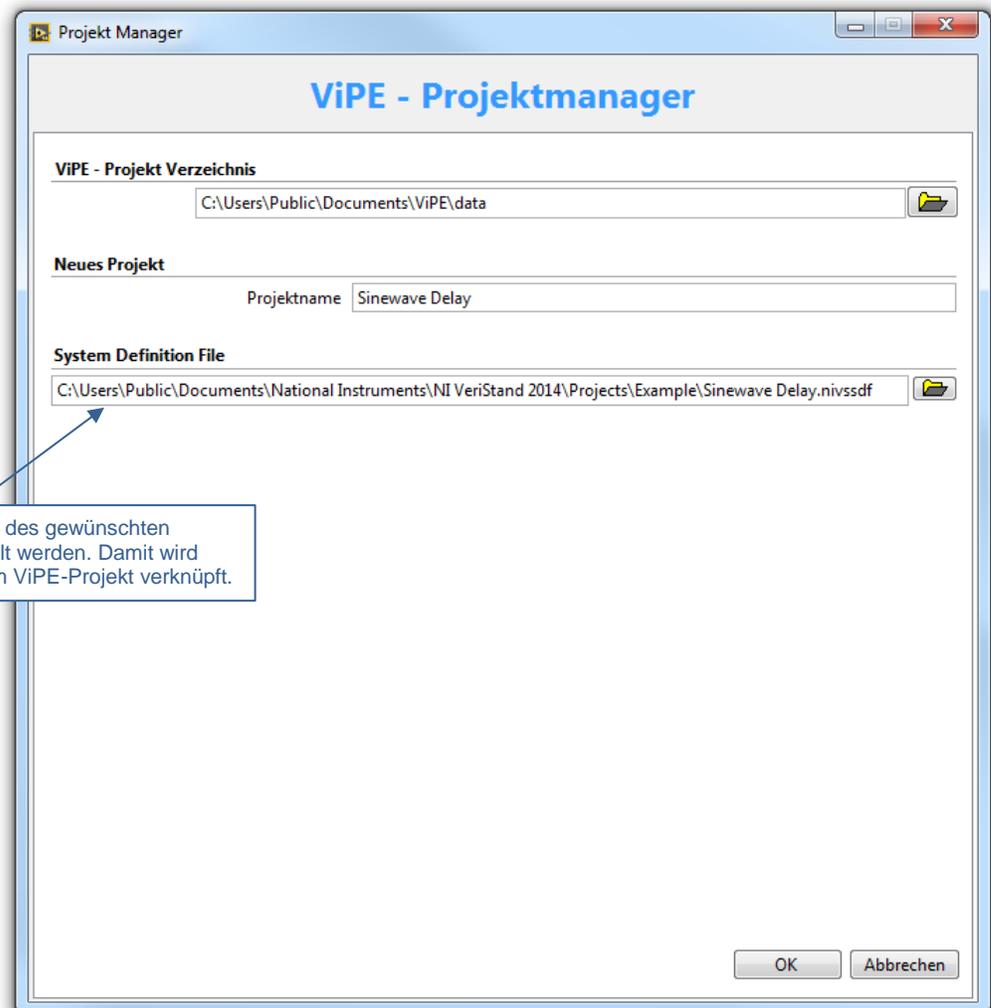


Abb.3: ViPE Projektmanager



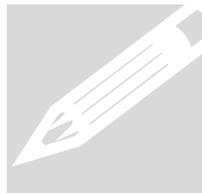
Das Anlegen eines Projekts im **CISWORKS ViPE** ist ein notwendiger Schritt um Echtzeitsequenzen erstellen und ablaufen lassen zu können. Einzelne Sequenzen müssen immer einem Projekt zugeordnet sein.

Um ein neues Projekt zu erstellen, klicken Sie im Projekt Manager links unten auf **Neu**. Hierbei öffnet sich folgender Dialog.



Hier muss die *.nivssdf- Datei des gewünschten VeriStand-Projekts ausgewählt werden. Damit wird das VeriStand Projekt mit dem ViPE-Projekt verknüpft.

Abb.4: ViPE Projektmanager Neues Projekt



Die System Definition File (*.nivssdf) kann nachträglich **nicht** mehr abgeändert werden. Daher ist darauf zu achten, gleich die korrekte Datei auszuwählen.

Sequenzeditor

Der Sequenzeditor ist in fünf unterschiedliche Bereiche aufgeteilt und stellt die Funktionalität bereit, die benötigt wird, um Sequenzen zu erstellen und bearbeiten zu können.

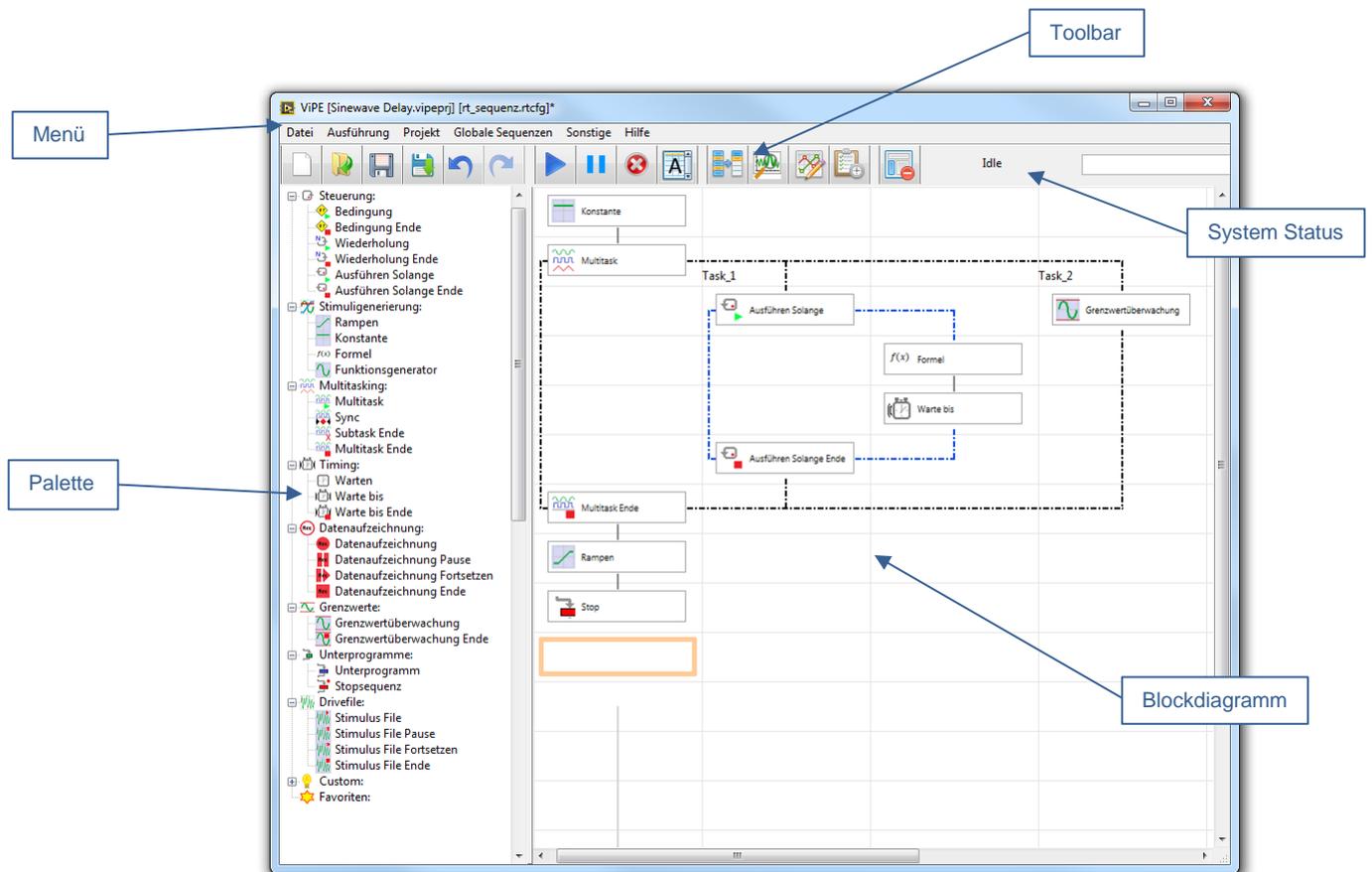


Abb.5: ViPE Sequenzeditor

Die fünf Bereiche **Menü**, **Toolbar**, **Palette**, **Blockdiagramm** und **System Status** werden im Folgenden näher erklärt.

Menü

Das Menü dient zur allgemeinen Steuerung und Verwaltung von Sequenzen und Projekten. Es gliedert sich in die sechs Unterpunkte: Datei, Ausführung, Projekt, Globale Sequenzen, Sonstige und Hilfe.

Im Folgenden wird die Funktionalität des Menüs näher erläutert.

DATEI

Neu	Legt eine neue Sequenz an
Öffnen ...	Öffnet eine bereits bestehende Sequenz
Zum Projekt hinzufügen	Fügt aktuelle Sequenz einem beliebigen Projekt hinzu
Speichern	Speichert aktuelle Sequenz
Speichern unter ...	Speichert aktuelle Sequenz in beliebigem Verzeichnis
Beenden	ViPE beenden

AUSFÜHRUNG

Compile	Kompiliert Sequenz
Run	Startet Sequenz
Pause	Pausiert ablaufende Sequenz
Stop	Stoppt ablaufende Sequenz

PROJEKT

Neu	Neues Projekt erstellen
Öffnen ...	Bestehendes Projekt öffnen
Speichern	Aktuelles Projekt speichern
Speichern als ...	Aktuelles Projekt in beliebigem Verzeichnis speichern

Globale Sequenzen

Die Absicht von globalen Stopp- und Alarmsequenzen ist der Schutz der Geräte. Sollten Sequenzen abgebrochen oder bestimmte Grenzwerte überschritten werden, so soll ein kontrolliertes Beenden bzw. Herunterfahren erfolgen, um keine Geräte zu beschädigen.

- Globale Stoppsequenzen** Öffnet das Globale Stoppsequenzen-Fenster
- Globale Alarmsequenzen** Öffnet das Globale Alarmsequenzen-Fenster
- Stoppsequenz speichern** Speichert aktuelle Sequenz als globale Stoppsequenz
- Alarmsequenz speichern** Speichert aktuelle Sequenz als globale Alarmsequenz



Globale Stopp- und Alarmsequenzen müssen über den Menüpunkt Globale Sequenzen abgespeichert werden um diese später auswählen zu können.

- Stoppsequenz öffnen** Bestehende Stoppsequenz öffnen
- Alarmsequenz öffnen** Bestehende Alarmsequenz öffnen



Der Geltungsbereich von Pre 1, Pre 2, Post 1 und Post 2 **Stoppsequenzen** ist global. Das bedeutet, dass diese Sequenzen bei allen lokalen Stoppsequenzen davor bzw. danach ausgeführt werden. **Ausführungsreihenfolge:** Pre 1 – Pre 2 – lokale Stoppsequenz – Post 1 – Post 2.



Die Ausführung von globalen **Alarmsequenzen** hängt vom ausgewählten Modus der Grenzwertüberwachung ab. Je nach Modus werden bei einer lokalen Alarmsequenz entweder keine, nur Pre 1 und Pre 2 oder alle globalen Alarmsequenzen ausgeführt. (siehe Grenzwertüberwachung)

SONSTIGE

Sprache	Deutsch oder Englisch
Shortcuts	Liste der verfügbaren Abkürzungen

HILFE

Handbuch	Ausführliches Benutzerhandbuch
Einleitungsvideos	Links zu Youtube-Videotutorials

Toolbar

Zum Steuern und Arbeiten mit dem **CISWORKS ViPE** werden eine Reihe von Buttons verwendet, die den Nutzer bei seinem Anliegen unterstützen. Diese Buttons sind mit selbsterklärenden Icons versehen, welche die wichtigsten Funktionen direkt bereitstellen.

ÜBERBLICK

	Neu	Legt eine neue Sequenz an
	Öffnen	Öffnet eine vorhandene Sequenz
	Speichern	Speichert die aktuelle Sequenz ab

	Speichern unter ...	Speichert die aktuelle Sequenz unter einem bestimmten Pfad ab
	Rückgängig	Macht die letzte Aktion rückgängig
	Wiederherstellen	Die zuletzt rückgängig gemachte Aktion wiederherstellen
	Start	Startet die Sequenz
	Pause	Pausiert die Sequenz
	Stopp	Beendet die Sequenz
	Autoscroll	Fokus auf aktuell ausgeführtem Schritt einer ablaufenden Sequenz
	Variablenmanager	Erstellen bzw. bearbeiten von lokalen Variablen
	TDMS Viewer	TDMS-Dateien anzeigen lassen
	Stimulus Editor	Stimulus File erzeugen und als CSV-Datei abspeichern
	Status Info	Zeigt Dauer, Restliche Zeit, Anzahl der abgearbeiteten Schleifen und Anzahl der abgearbeiteten Blöcke einer ablaufenden Sequenz an.
	Funktionspalette ein-/ausblenden	Palette anzeigen bzw. ausblenden

Die Toolbarfunktionen **Variablenmanager**, **TDMS Viewer** und **Stimulus Editor** werden im Folgenden näher erläutert.

VARIABLENMANAGER

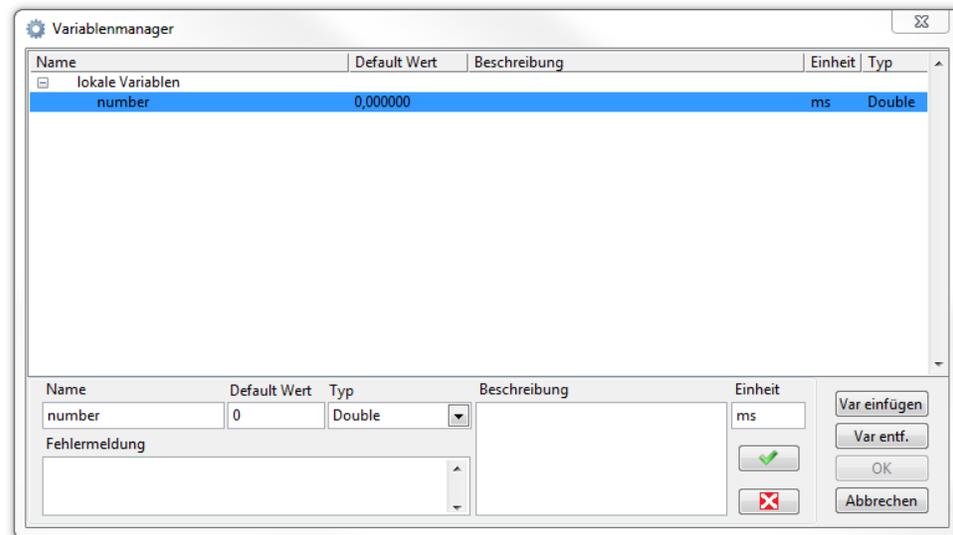
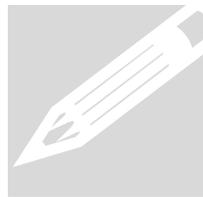


Abb.6: ViPE Variablenmanager

Mit Hilfe des **Variablenmanagers** können lokale Variablen angelegt und mit einem Default Wert sowie einer Einheit versehen werden. Auf lokale Variablen kann von Funktionsblöcken aus zugegriffen werden.



Lokale Variablen haben einen **lokalen Geltungsbereich**, was bedeutet, dass sie nur in derjenigen Sequenz verwendet werden können, in welcher sie erstellt wurden. Außerdem muss der Name einer Variablen eindeutig sein und darf keine der folgenden Zeichen enthalten: `__ . , - ; ! /`

TDMS VIEWER

Die Datenaufzeichnung wird im TDMS-Format abgespeichert. Mit dem **TDMS Viewer** können diese Dateien geöffnet und als Graph betrachtet werden.

TDMS-Dateien auswählen



Hierzu wird die gewünschte TDMS-Datei aus dem *TDMS Dateien auswählen* Menü in das rechte Fenster verschoben.

Dateiinhalte anschauen



Danach wird in das *Datei Inhalte anschauen* Menü gewechselt und das gewünschte Messkriterium von der rechten Seite in den Graphen hineingezogen.

STIMULUS EDITOR

Im **Stimulus Editor** können Kurven erzeugt und als CSV-Datei zur weiteren Verwendung abgespeichert werden. Die Parameter Frequenz, Amplitude und Offset spezifizieren den Verlauf der Kurve. Dateilänge und Target Rate spezifizieren die Anzahl der Punkte. Es stehen die drei Kurvenarten Sinus, Rechteck und Lärm zur Verfügung.



Zu beachten ist, dass die Target Rate der erzeugten Kurven mit der Target Rate des VeriStand-Targets übereinstimmen muss.

Blockdiagramm

Grafische Darstellung der Sequenz, wie in einem Flussdiagramm. In das Blockdiagramm können die Funktionsblöcke der Palette hineingezogen werden. Der Datenfluss bewegt sich von oben nach. Während der Ausführung ist für den Benutzer ersichtlich, an welcher Stelle sich die Sequenz befindet.

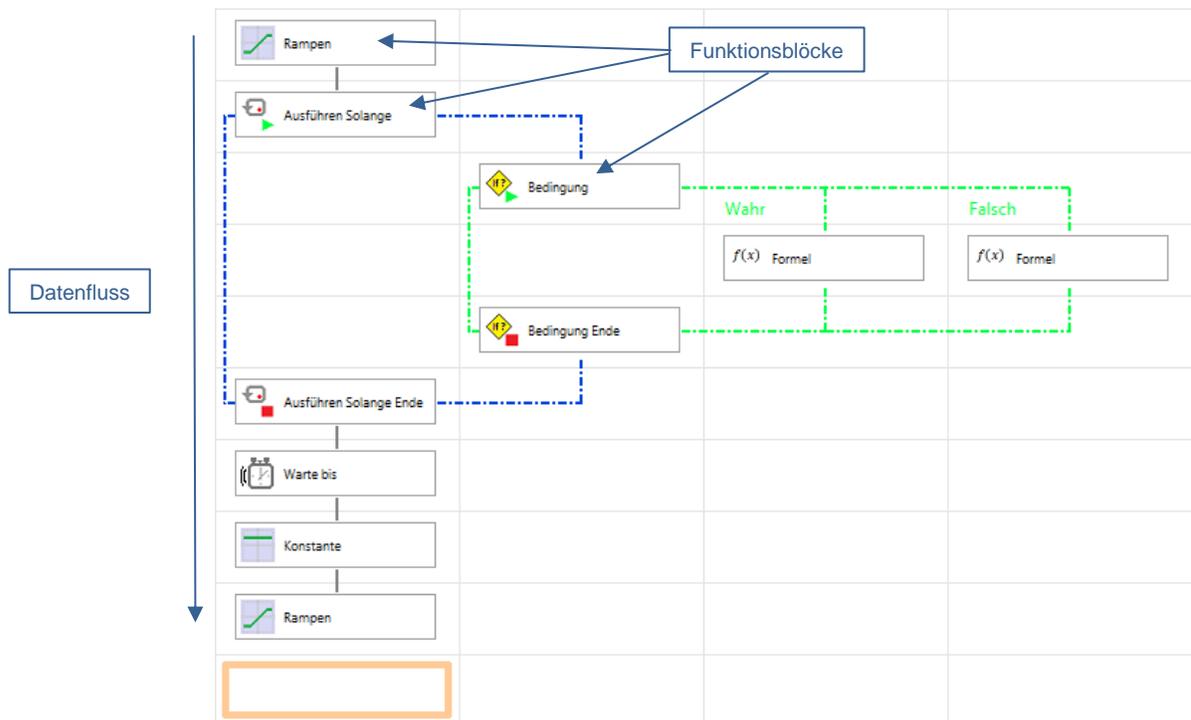


Abb.7: ViPE Blockdiagramm

Bereits hinzugefügte Blöcke können per Drag & Drop verschoben werden, ein Doppelklick darauf öffnet das Konfigurationsfenster. Rechtsklick auf einen Block stellt weitere Funktionalität zur Verfügung. So kann zum Beispiel ein Block deaktiviert bzw. wieder aktiviert oder zu den Favoriten hinzugefügt werden.

Palette

Gewünschte Funktionsblöcke können ganz bequem per Drag & Drop dem Blockdiagramm hinzugefügt werden. In dem danach öffnenden Konfigurationsfenster, kann dieser Block seiner jeweiligen Funktion entsprechend spezifisch konfiguriert werden. Neben grundlegenden Funktionen zur Sequenzsteuerung stellt die Palette auch Blöcke zur Verfügung, um Messdaten zu erfassen sowie Alarmsequenzen zu definieren, welche bei der Überschreitung von Grenzwerten ausgeführt werden können.

BUTTONS

In manchen Funktionsblöcken wird anhand von Buttons mathematische Funktionalität bereitgestellt. Die Verwendung dieser Buttons wird im Folgenden näher erläutert.



Klammer auf

Funktionsparameter werden mit runden Klammern eingeschlossen



Klammer zu



Klammer auf

Variablen, Kanäle und Aliase werden mit eckigen Klammern eingeschlossen



Klammer zu



Zuweisung

Zuweisungsoperator



Größer

Vergleichsoperator

	Kleiner	Vergleichsoperator
	Größer gleich	Vergleichsoperator
	Kleiner gleich	Vergleichsoperator
	Oder	Logische Oder-Operation
	Und	Logische Und-Operation
	Gleich	Vergleichsoperator
	Ungleich	Vergleichsoperator
	Sinus	Sinusfunktion, Verwendung: $\sin(x)$ mit $x =$ Winkel in Grad
	Kosinus	Kosinusfunktion Verwendung: $\cos(x)$ mit $x =$ Winkel in Grad
	Tangens	Tangensfunktion Verwendung: $\tan(x)$ mit $x =$ Winkel in Grad
	Wurzel	$\text{sqrt}(x)$ liefert Quadratwurzel von x
	Exponentialfunktion	Exponentialfunktion mit eulerschen Zahl e als Basis Verwendung: $\text{exp}(x)$
	Potenz	Verwendung: $\text{pow}(x,y)$ mit x als Basis und y als Exponent
	Logarithmus	Verwendung: $\log(x,y)$ mit y als Basis
	Natürlicher Logarithmus	Logarithmus zur Basis e , Schreibweise: $\ln(x)$ mit $x = \text{exp}(\ln(x))$

isnan	IsNotANumber	Liefert TRUE wenn der Wert keine Zahl ist
abs	Betrag	$\text{abs}(x)$ liefert Betrag von x
acos	Arkuskosinus	Arkuskosinusfunktion Verwendung: $\text{acos}(x)$ mit x zwischen -1 und 1 liefert Winkel in Bogenmaß
asin	Arkussinus	Arkussinusfunktion Verwendung: $\text{asin}(x)$ mit x zwischen -1 und 1 liefert Winkel in Bogenmaß
atan	Arkustangens	Arkustangensfunktion Verwendung: $\text{atan}(x)$ liefert Winkel in Bogenmaß
mod	Modulo	Modulofunktion Verwendung: $\text{mod}(x,y)$ liefert Rest von $x \div y$
rand	Zufallszahl	Zufallszahlgenerator Verwendung: $\text{rand}(x)$ liefert Zufallszahl zwischen 0 und x
sign	Vorzeichen	Verwendung: $\text{sign}(x)$ liefert 1 bei $x > 0$, -1 bei $x < 0$, 0 bei $x = 0$

FUNTIONSBLÖCKE

Steuerung

Bedingung

Nachdem eine **Bedingung** in das Blockdiagramm gezogen wird, öffnet sich das Konfigurationsfenster, in welchem die Bedingung festgelegt werden kann.

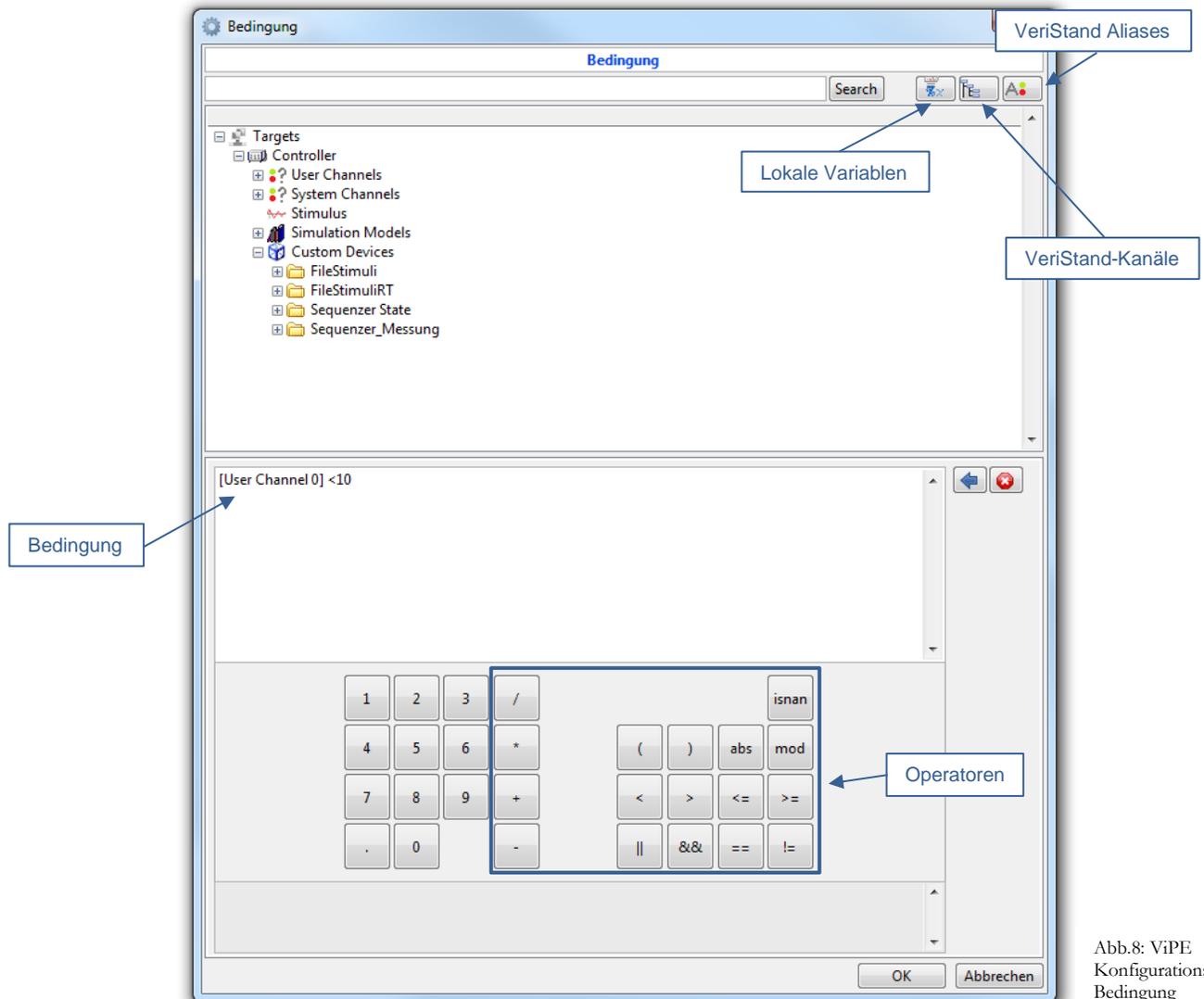


Abb.8: ViPE Konfigurationsfenster Bedingung

Hierbei können Lokale Variablen, VeriStand Kanäle oder VeriStand Aliases verwendet werden. Mit Hilfe der Suchfunktion lassen sich bestimmte Variablen, Kanäle und Aliases finden. Um die Bedingung festzulegen oder zu bearbeiten, können neben der Tastatur auch die Zahlen- bzw. Operatorenbuttons verwendet werden.

Bei der Ausführung der Sequenz wird die Bedingung geprüft. An einer Verzweigung wird entweder der Wahr- oder der Falsch-Pfad ausgeführt, abhängig davon, ob die Bedingung erfüllt ist oder nicht.

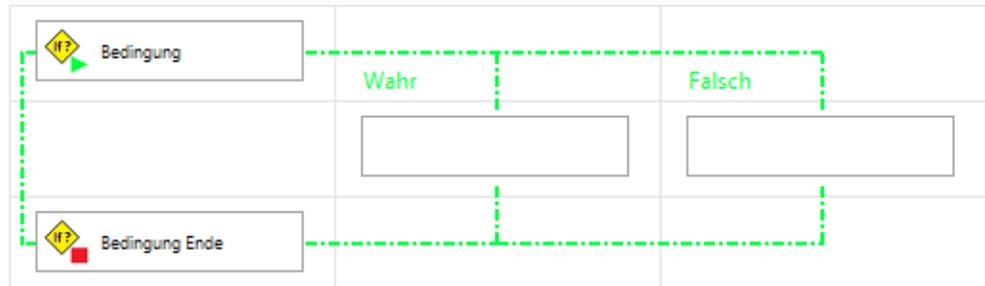


Abb.9: ViPE Bedingung

Bedingung Ende

Der Anweisungsblock innerhalb *Bedingung* muss immer mit *Bedingung Ende* abgeschlossen werden. Wurde *Bedingung Ende* ausversehen gelöscht, so kann es per Drag & Drop dem Blockdiagramm wieder hinzugefügt werden.

Wiederholung

Der Anweisungsblock innerhalb einer *Wiederholung* wird mit einer bestimmten Anzahl an Wiederholungen ausgeführt. Im unteren Beispiel werden 12 Wiederholungen ausgeführt.



Abb.10: ViPE Wiederholung

Wiederholung Ende

Der Anweisungsblock innerhalb *Wiederholung* muss immer mit *Wiederholung Ende* abgeschlossen werden. Wurde *Wiederholung Ende* ausversehen gelöscht, so kann es per Drag & Drop dem Blockdiagramm wieder hinzugefügt werden.

Ausführen Solange

Wie beim *Bedingung* Block ist auch hier eine Bedingung zu definieren. Der Anweisungsblock innerhalb der *Ausführen Solange* Funktion wird solange wiederholt, solange die Bedingung gültig ist.



Abb.11: ViPE Ausführen Solange

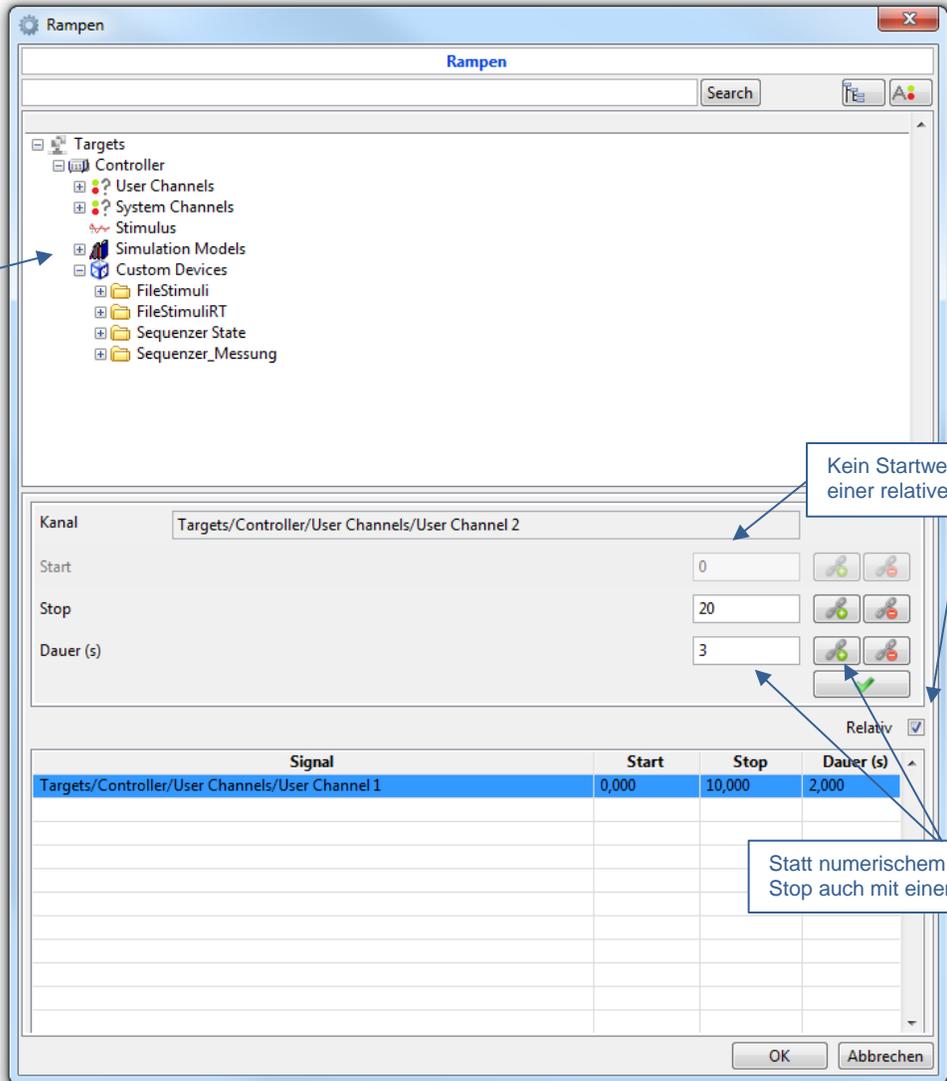
Ausführen Solange Ende

Der Anweisungsblock innerhalb *Ausführen Solange* muss immer mit *Ausführen Solange Ende* abgeschlossen werden. Wurde *Ausführen Solange Ende* ausversehen gelöscht, so kann es per Drag & Drop dem Blockdiagramm wieder hinzugefügt werden.

Stimuli- generierung

Rampen

Der Wert eines auszuwählenden Kanals verändert sich innerhalb einer Zeitspanne von einem Start- in einen Stoppwert linear. Es besteht die Möglichkeit Dauer sowie Start- und Stoppwert numerisch festzulegen oder diese mit einem Kanal zu verknüpfen. Als Kanal können Kanäle und Aliase verwendet werden. Des Weiteren kann man den Startwert absolut oder relativ festlegen. Ein relativer Startwert bedeutet, dass der aktuelle Wert des Kanals als Startwert fungiert.



Baumstruktur der VeriStand-Kanäle

Kein Startwert bei einer relativen Rampe

Statt numerischem Wert könnte man Stop auch mit einem Kanal verknüpfen

Signal	Start	Stop	Dauer (s)
Targets/Controller/User Channels/User Channel 1	0,000	10,000	2,000

Abb.12: ViPE Konfigurationsfenster Rampen



Konstante

Auszuwählenden Kanälen kann jeweils ein bestimmter Wert zugewiesen werden. Diese Werte werden direkt bei der Ausführung des Konstante-Funktionsblocks übernommen. Mit der Dauer lässt sich festlegen, wie lange gewartet werden soll bis die Funktion beendet wird.

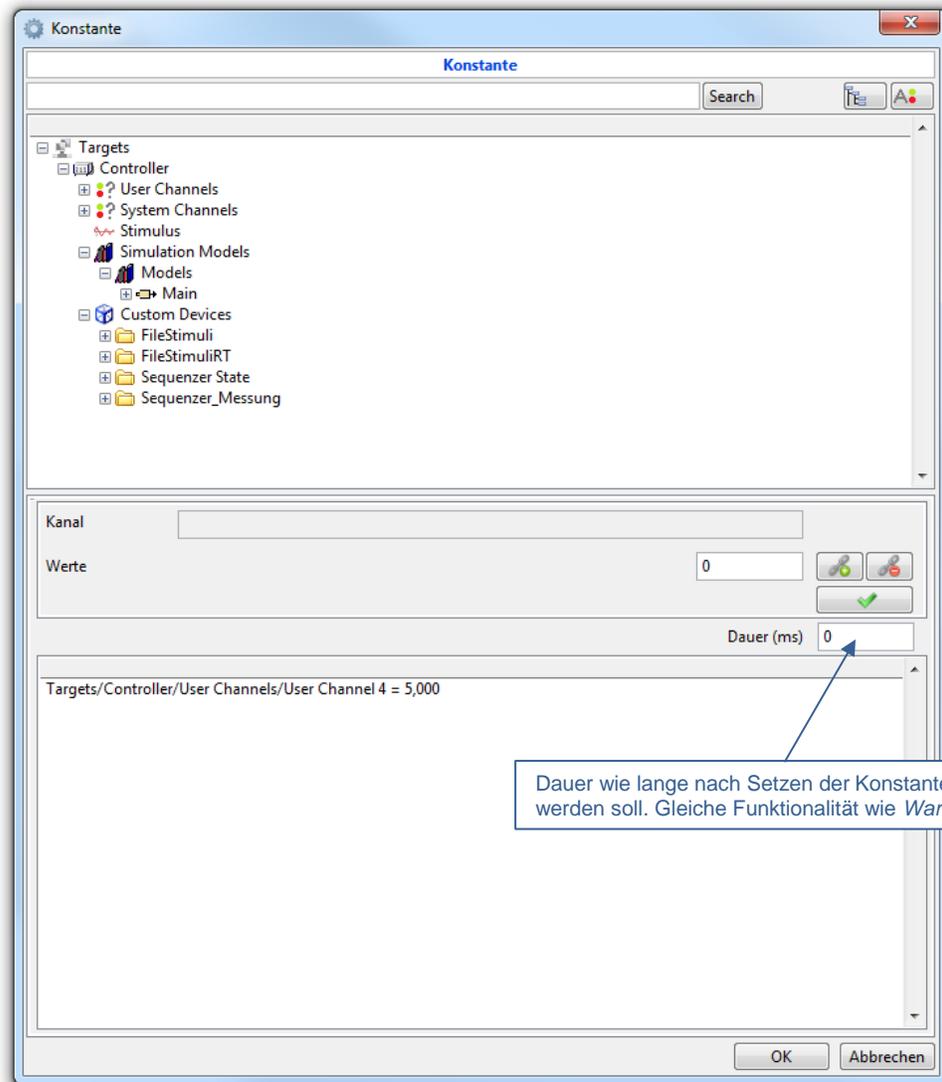


Abb.13: ViPE Konfigurationsfenster Konstante

$f(x)$ Formel

Eine **Formel** lässt sich vielseitig anwenden. Die Werte von Variablen, Kanälen oder Aliassen lassen sich mit Hilfe von Operatoren und Funktionen manipulieren.

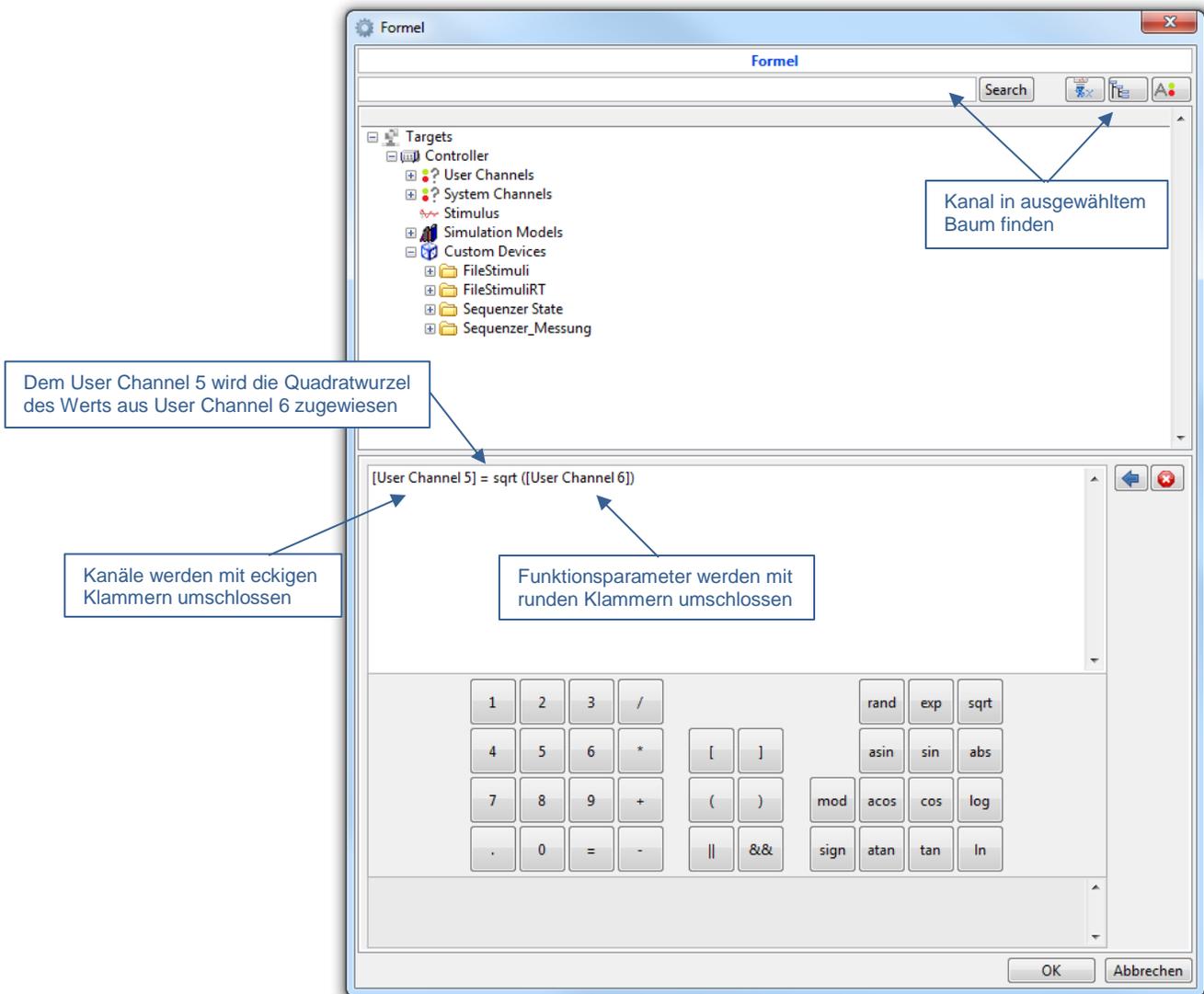


Abb.14: ViPE Konfigurationsfenster Formel



Funktionsgenerator

Mit dem **Funktionsgenerator** lassen sich Funktionen auf einen auszuwählenden Kanal legen. Es stehen vier Funktionen zur Auswahl:

Sinus, Sägezahn, PWM (Pulsweitenmodulation), Dreieck

Für jede Funktion können folgende Einstellungen getroffen werden:

- Amplitude** Maximale Auslenkung in y-Richtung (positiver und negativer Bereich) ausgehend von der x-Achse
- Bias** Versatz der zu y addiert wird. x-Achse ist per Default auf $y=0$
- Frequenz** Schnelligkeit der aufeinanderfolgenden Wiederholungen in Hertz
- Phase** Startposition der Funktion in Winkel (Grad)
- Dauer** Dauer der Funktion in Sekunden
- Tastgrad** gibt Verhältnis von $Amplitude - Bias$ zu $Amplitude + Bias$ in Prozent an (**Nur bei PWM**)

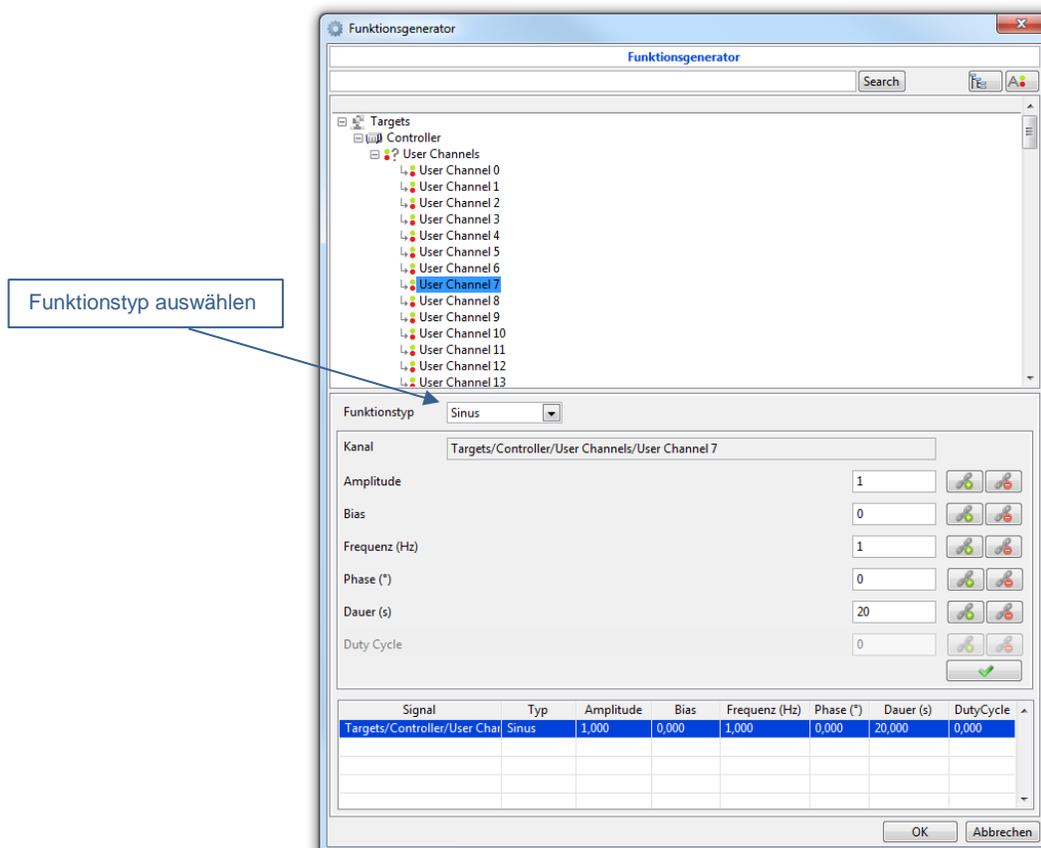


Abb.15: ViPE Konfigurationsfenster Funktionsgenerator

Multitasking



Es können bis zu 8 Anweisungsblöcke in sogenannten Tasks parallel abgearbeitet werden.

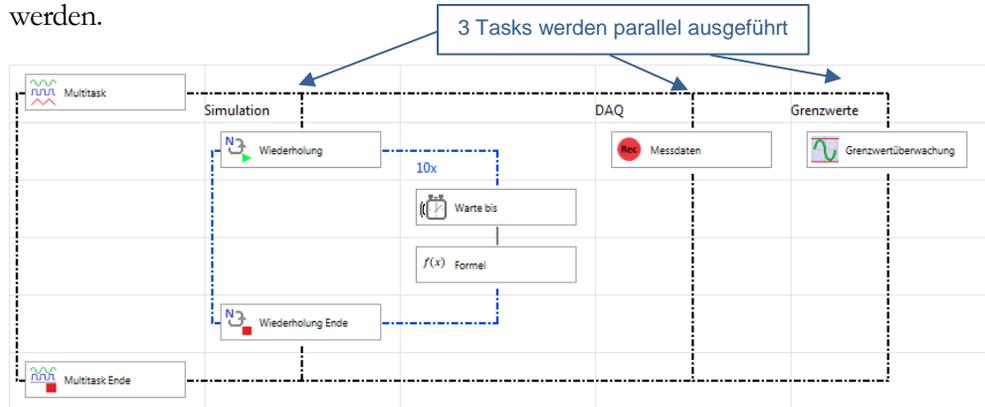


Abb.16: ViPE Multitask



Innerhalb von *Multitask*-Anweisungsblöcken lässt sich mit Hilfe von *Sync* festlegen, dass alle Tasks zuerst ihren *Sync*-Block erreicht haben müssen, bevor nachfolgende Funktionsblöcke ausgeführt werden. So wird im unteren Beispiel *Konstante* in Task2 erst ausgeführt, wenn der *Wiederholung* aus Task1 beendet ist.

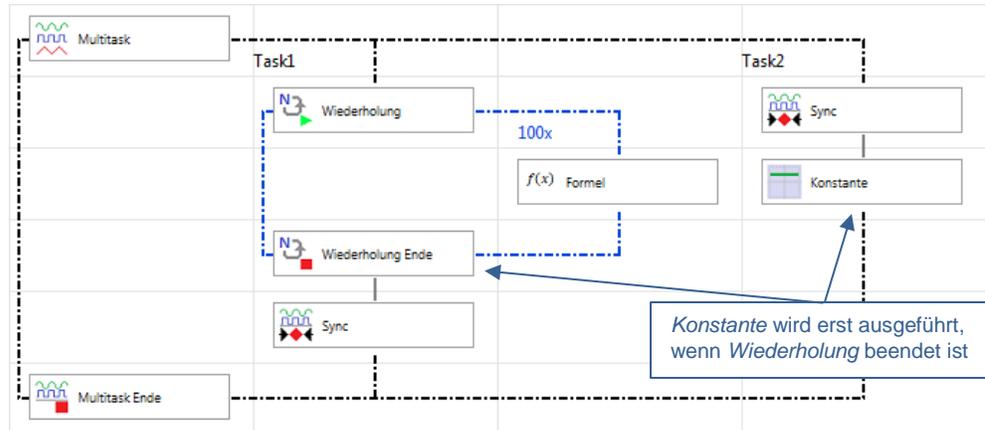


Abb.17: ViPE Sync

Subtask Ende

Innerhalb von *Multitask* lässt sich aus einer Task heraus ein anderer Task beenden.

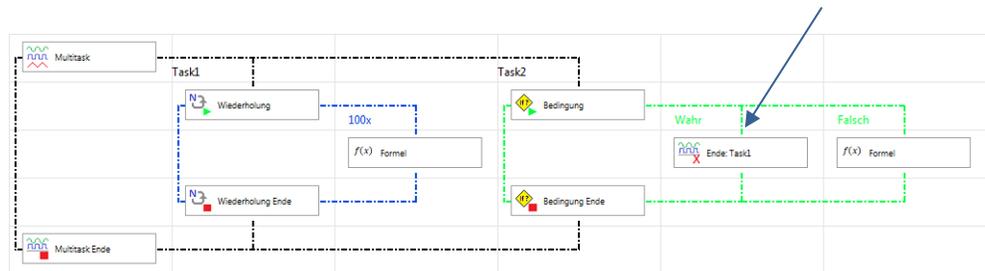


Abb.18: ViPE Subtask Ende

Multitask Ende

Der Anweisungsblock innerhalb *Multitask* muss immer mit *Multitask Ende* abgeschlossen werden. Wurde *Multitask Ende* ausversehen gelöscht, so kann es per Drag & Drop dem Blockdiagramm wieder hinzugefügt werden.

Timing

Warten

Wartet eine zu bestimmende Zeit. Im unteren Block wird 10 s gewartet, bis der nächste Funktionsblock ausgeführt wird.

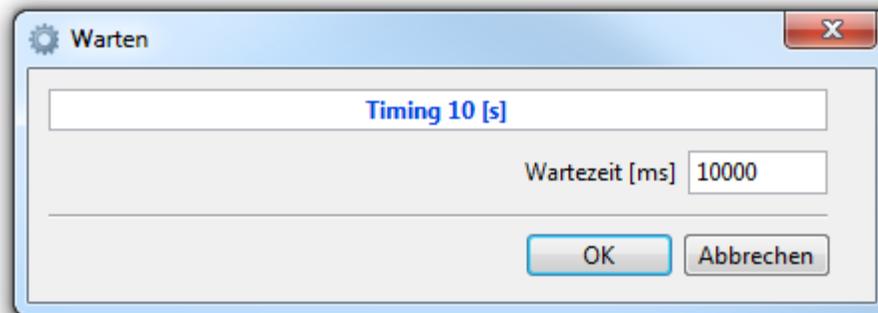


Abb.19: ViPE Warten

Warte bis

Wartet bis die festgelegte Bedingung erfüllt ist. Es lässt sich ein Timeout definieren, der die Funktion nach einer gewissen Zeit abbricht, wenn die Bedingung nicht erfüllt ist. Die Wartezeit bis zum Timeout wird in Millisekunden angegeben.



Warte bis Ende

Ist ein Timeout in *Warte bis* festgelegt, so ergibt sich eine Verzweigung in Standard und Timeout. Diese Verzweigung muss am Ende immer durch *Warte bis Ende* abgeschlossen werden. Tritt die Bedingung innerhalb des definierten Zeitraums auf, wird der Block im Zweig *Standard* ausgeführt. Tritt diese Bedingung innerhalb des definierten Zeitraums nicht auf, so kommt es zum Timeout und dann wird der Block im Zweig *Timeout* ausgeführt.

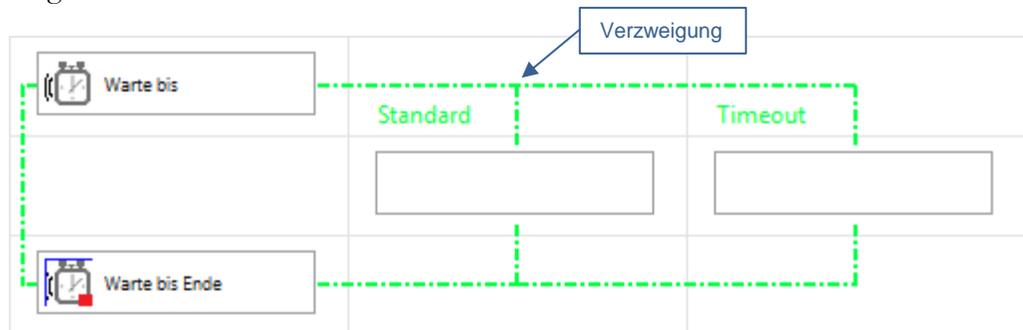


Abb.20: ViPE Warte bis Ende

Daten- aufzeichnung

Datenaufzeichnung

Werte von Variablen, Kanälen und Aliasen können aufgezeichnet und in einer TDMS-Datei abgespeichert werden. TDMS ist ein Datenformat von National Instruments, welches mit dem TDMS Viewer direkt im **CISWORKS ViPE** geöffnet werden kann. Die aufzuzeichnenden Kanäle werden per Doppelklick hinzugefügt.

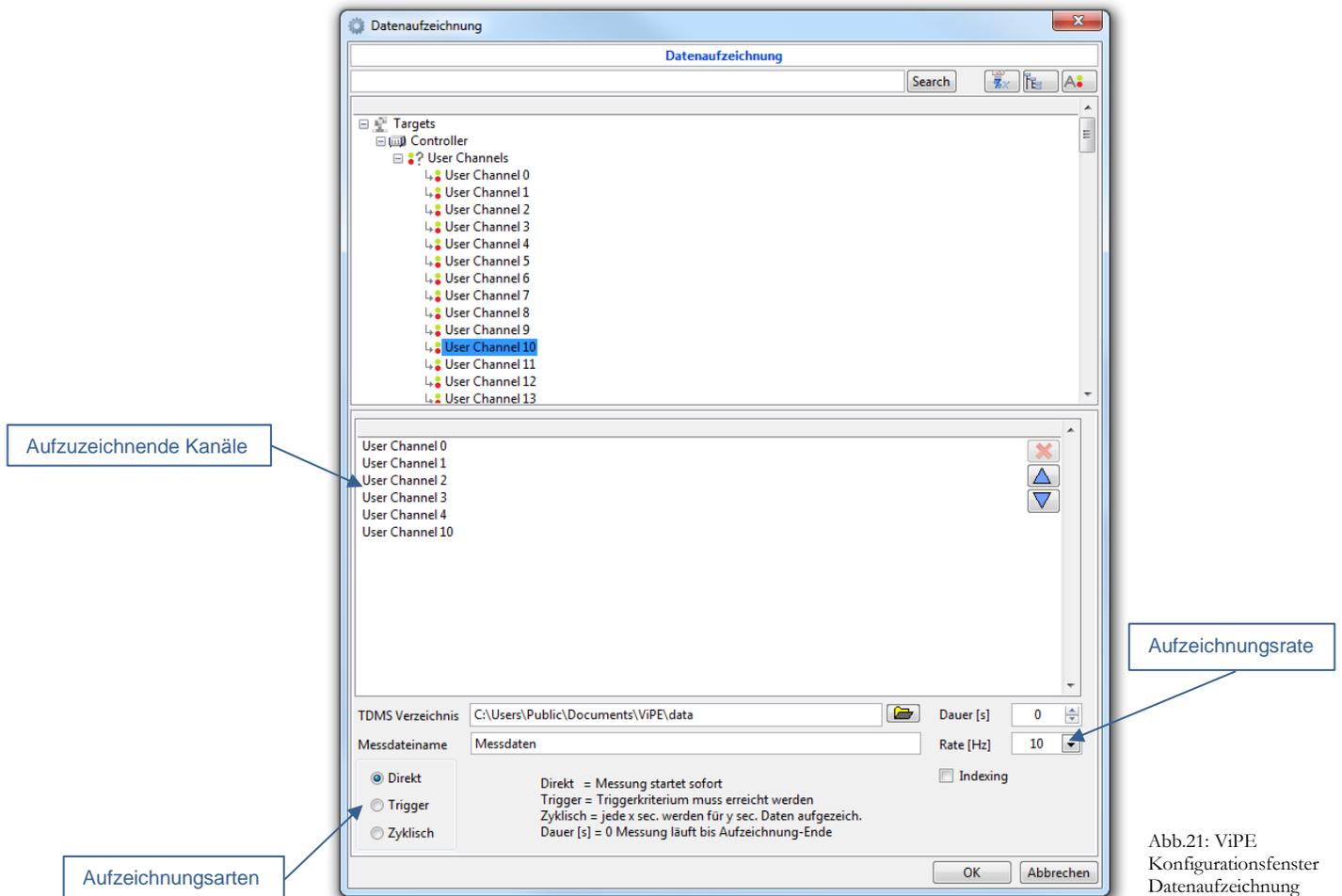


Abb.21: ViPE
Konfigurationsfenster
Datenaufzeichnung

Konfigurationsmöglichkeiten

Dauer Dauer [s] = 0 bedeutet, dass die Messung bis Aufzeichnungsende oder bis zum Ende der Sequenz läuft

Rate Messfrequenz in Hertz

Indexing Falls *Datenaufzeichnung* innerhalb einer *Wiederholung* oder eines *Ausführen Solange* mehrmals wiederholt wird, so wird mit einem

gesetzten Haken für jede Wiederholung eine neue Datei mit fortlaufender Nummer angelegt.

Aufzeichnungsarten

- Direkt** Messung startet sofort mit der eingestellten Frequenz
- Trigger** Messung startet erst dann, wenn die Triggerbedingung erfüllt ist. Es lässt sich für einen Kanal oder für ein Alias eine Triggerbedingung festlegen. Hierbei stehen folgende Operatoren zur Verfügung: > (größer), >= (größer gleich), < (kleiner), <= (kleiner gleich), == (gleich), -| | - (offenes Intervall), |--| (abgeschlossenes Intervall)
- Zyklisch** Die Daten werden für den Zeitraum des in Messdauer angegebenen Wertes aufgezeichnet. Danach wird die Messdatenaufzeichnung für den Zeitraum des in Zykluszeit angegebenen Wertes unterbrochen. Dieser Ablauf wird für die Dauer der Messdatenaufzeichnung wiederholt.



Datenaufzeichnung Pause

Die Messdatenaufzeichnung wird unterbrochen.



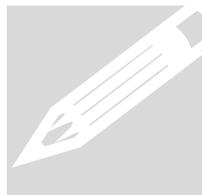
Datenaufzeichnung Fortsetzen

Eine unterbrochene Messdatenaufzeichnung wird fortgesetzt.



Datenaufzeichnung Ende

Die Messdatenaufzeichnung wird beendet.



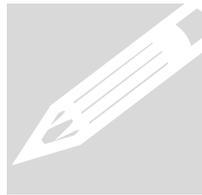
Datenaufzeichnung Pause und *Datenaufzeichnung Ende* erfolgen nicht echtzeitgenau. Das bedeutet, dass nach der Ausführung dieser Blöcke weitere Daten für mehrere Millisekunden aufgezeichnet werden können.

Grenzwerte



Grenzwertüberwachung

Es lässt sich eine Bedingung definieren. Wird diese erfüllt so startet eine auszuwählende Alarmsequenz.



Es ist darauf zu achten, Alarmsequenzen im dafür vorgesehenen Ordner (... \Sequences \Alarm Sequences) abzuspeichern. Nur dann können diese ausgewählt werden.

Bei der Ausführung dieser Alarmsequenz gibt es vier verschiedene Modi:

Kontinuierlich Immer wenn die Grenzwertbedingung erfüllt ist, wird die Alarmsequenz aufgerufen. Nach der Abarbeitung der Alarmsequenz läuft die eigentliche Sequenz weiter. Vor und nach der Alarmsequenz werden keine globalen Alarmsequenzen aufgerufen.

Pre 1 → Pre 2 → **Alarmsequenz** → Post 1 → Post 2

Einmalig Die Alarmsequenz wird lediglich bei der ersten Erfüllung der Grenzwertbedingung aufgerufen, ansonsten nicht mehr. Vor und nach der Alarmsequenz werden keine globalen Alarmsequenzen aufgerufen.

Pre 1 → Pre 2 → **Alarmsequenz** → Post 1 → Post 2

Sequenz Abbrechen Nach der Abarbeitung der Alarmsequenz wird die Sequenz abgebrochen. Vor dieser Alarmsequenz werden noch die beiden globalen Alarmsequenzen Pre Alarm 1 & 2 aufgerufen.

Pre 1 → Pre 2 → **Alarmsequenz** → Post 1 → Post 2

Sequenz Pausieren Nach der Abarbeitung der Alarmsequenz wird die Sequenz pausiert. Das Fortsetzen der Sequenz muss manuell durch den Benutzer erfolgen. Vor und nach dieser Alarmsequenz werden alle globalen Alarmsequenzen aufgerufen.

Pre 1 → Pre 2 → **Alarmsequenz** → Post 1 → Post 2

Wenn Bedingung erfüllt, dann wird Alarmsequenz ausgelöst

Alarmsequenz auswählen

Modus

Abb.22: ViPE Konfigurationsfenster Grenzwertüberwachung

Grenzwertüberwachung Ende

Beendet eine auszuwählende aktive Grenzwertüberwachung.

Den zu beendenden Grenzwert auswählen

Abb.23: ViPE Grenzwertüberwachung Ende

Unterprogramme



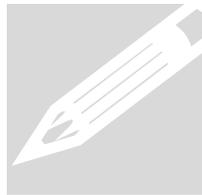
Unterprogramm

Eine vorhandene Sequenz kann als **Unterprogramm** in die aktuelle Sequenz eingebunden werden. Dabei können nicht nur **CISWORKS ViPE**-Sequenzen (*.rtcfg) sondern auch VeriStand-Sequenzen (*.nvsseq) aufgerufen werden.



Stoppsequenz

Eine **Stoppsequenz** kann an beliebiger Stelle in der aktuellen Sequenz eingebunden werden. Diese **Stoppsequenz** kommt nur dann zur Ausführung, wenn die Ausführung der aktuellen Sequenz manuell vom Benutzer abgebrochen wird.



Wird in einer Sequenz, in der eine Stoppsequenz definiert ist eine Untersequenz aufgerufen, in welcher ebenfalls eine Stoppsequenz definiert ist, so wird die Stoppsequenz aus der aufrufenden Sequenz im Falle eines Abbruchs ausgeführt.

Stimulus File



Stimulus File

Als **Stimulus Files** können Dateien (*.txt, *.csv, *.dsv) geladen werden, die zuvor im Stimulus Editor erstellt wurden. Auf die Werte der FileStimuli kann in den VeriStand-Kanälen zugegriffen werden. Der Zugriff auf die Werte der einzelnen Spalten läuft über folgende VeriStand-Kanäle (Kanäle unter Custom Devices/FileStimuliRT/Output im VeriStand System Explorer der System Definition File):

Stimulus File	1. Spalte	2. Spalte	3. Spalte	4. Spalte	...
VeriStand-Kanal	Elapsed_Time[s]	Out0	Out1	Out2	...

Folgendes Beispiel zeigt ein generiertes **Stimulus File** (*.csv) mit drei Kurven:
Spalte B = **Sinus** (Frequenz = 1, Amplitude = 1, Offset = 0)
Spalte C = **Rechteck** (Frequenz = 1, Amplitude = 1, Offset = 0)
Spalte D = **Lärm** (Frequenz = 1, Amplitude = 1, Offset = 0)

1. Spalte = Elapsed_Time[s]

Target Rate = 100 Hz
=> alle 10 ms ein Punkt

	A	B	C	D
1	0.000	0.000	1.000	-3.395
2	0.010	0.063	1.000	3.029
3	0.020	0.125	1.000	3.975
4	0.030	0.187	1.000	-24.979
5	0.040	0.249	1.000	-1.518
6	0.050	0.309	1.000	-7.481
7	0.060	0.368	1.000	25.645
8	0.070	0.426	1.000	28.757
9	0.080	0.482	1.000	-12.404
10	0.090	0.536	1.000	27.269
11	0.100	0.588	1.000	-7.813
12	0.110	0.637	1.000	-8.566
13	0.120	0.685	1.000	3.833
14	0.130	0.729	1.000	-11.347
15	0.140	0.771	1.000	1.039
16	0.150	0.809	1.000	17.938
17	0.160	0.844	1.000	9.522
18	0.170	0.876	1.000	-5.527
19	0.180	0.905	1.000	-23.989
20	0.190	0.930	1.000	-2.240
21	0.200	0.951	1.000	-1.764
22	0.210	0.969	1.000	11.844
23	0.220	0.982	1.000	45.221

Lärm:
4. Spalte = Out2

Sinus:
2. Spalte = Out0

Rechteck:
3. Spalte = Out1

Abb.24: Stimulus File

Die Abarbeitung eines **Stimulus File** läuft nach einer Wartezeit von 2 Sekunden im Hintergrund einer Sequenz ab. Die Werte werden für die in der **Stimulus File** spezifizierten Dauer gelesen. Es kann aber auch die zeitliche Start- und Stopposition manuell festgelegt werden. Ist die Abarbeitung der Sequenz beendet, so endet auch das Einlesen des **Stimulus File**. Das Einlesen der Werte kann mit den folgenden Funktionsblöcken manipuliert werden.

Stimulus File Pause

Einlesen des Stimulus File unterbrechen.

Stimulus File Fortsetzen

Einlesen des Stimulus File fortsetzen.

Stimulus File Ende

Einlesen des Stimulus File beenden.

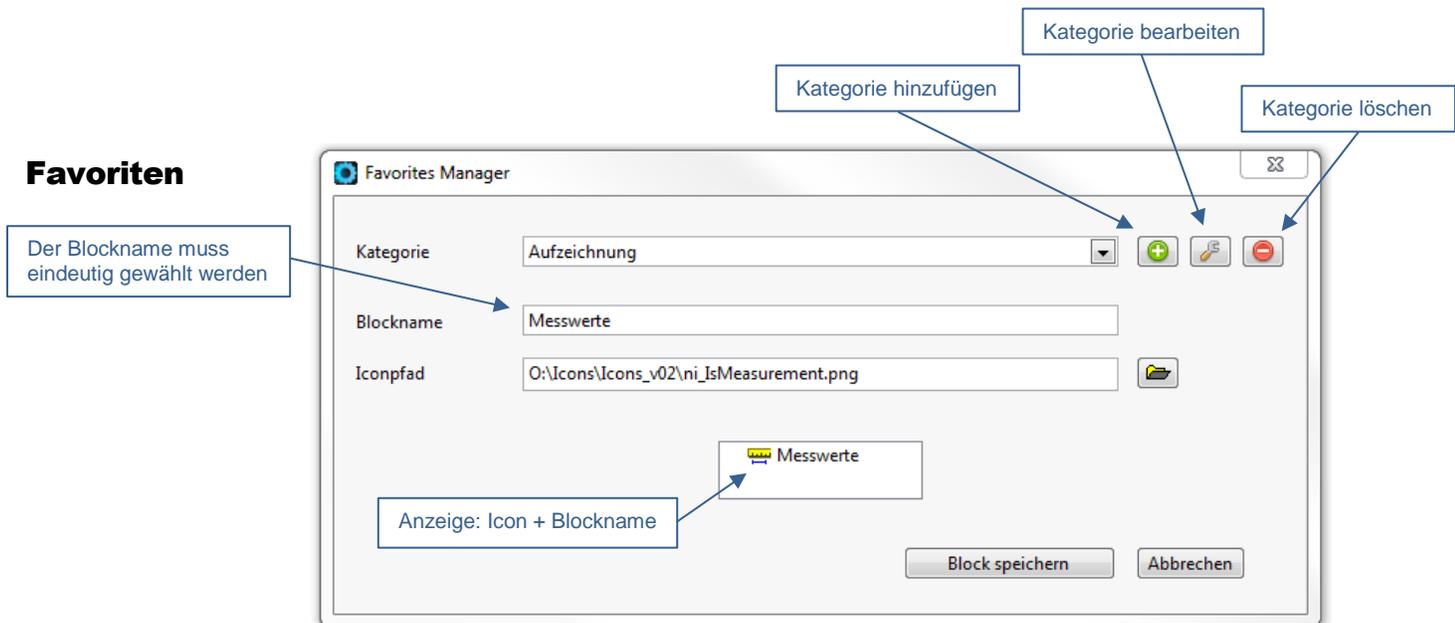


Abb.25: ViPE Favoritenmanager

Wird die Konfiguration eines Blocks mehrmals verwendet, kann dieser als Favorit abgespeichert werden. Dazu muss mit einem Rechtsklick auf den gewünschten Funktionsblock im Blockdiagramm der Unterpunkt „Zu Favoriten hinzufügen“ ausgewählt werden. Dabei kann der Favorit mit einem Icon versehen und einer Kategorie hinzugefügt werden. Ein eindeutiger Blockname muss gewählt werden. Ist ein Favorit erstellt, so wird dieser der Palette hinzugefügt.

System Status

Der System Status gibt Auskunft über den aktuellen Status der Sequenz.

Running

Running: Die Sequenz läuft gerade ab und kann in dieser Zeit nicht bearbeitet werden. Sie können den aktuellen Fortschritt im Blockdiagramm verfolgen.



Autoscroll aktivieren um immer den aktuellen Stand der Sequenz im Fokus zu haben.



Idle

Idle: Die Sequenz ist nicht aktiv und kann bearbeitet werden.

Paused

Paused: Ausführung der Sequenz ist pausiert. Erneutes Drücken auf Pause nimmt Ausführung der Sequenz wieder auf. Die Ausführung

Remotesteuerung

Um den **CISWORKS ViPE** per TCP Protokoll remote steuern zu können muss der **CISWORKS ViPE** beim Start mit den entsprechenden Parametern versehen werden. Dazu muss der **ViPE** folgendermaßen über die Windows Konsole gestartet werden:

```
<Pfad der CISWORKS ViPE.exe> Sequenzer/t"Labview-Remote" /d"<Pfad zur Veristand System Definition File (.nivssdf)>" /p"<Pfad zum ViPE-Projekt (.vipeprj)>" /m"Menu"
```

Beispiel

```
C:\>C:\Program Files (x86)\CISWORKS\ViPE\ViPE.exe Sequenzer/t"Labview-Remote" /d  
"C:\Projekte\Testprojekt\Testprojekt.nivssdf" /p"C:\Users\Public\Documents\ViPE\  
Testprojekt\Testprojekt.vipeprj" /m"Menu"
```

Wird der **CISWORKS ViPE** so gestartet, wartet ein TCP Listener im Hintergrund auf eingehende Verbindungen. Mittels der IP Adresse des Rechners, auf welchem der **CISWORKS ViPE** läuft und dem Service Name „LocalSequenzPort“ lässt sich eine TCP Verbindung herstellen.

Grundlegende Funktionsweise

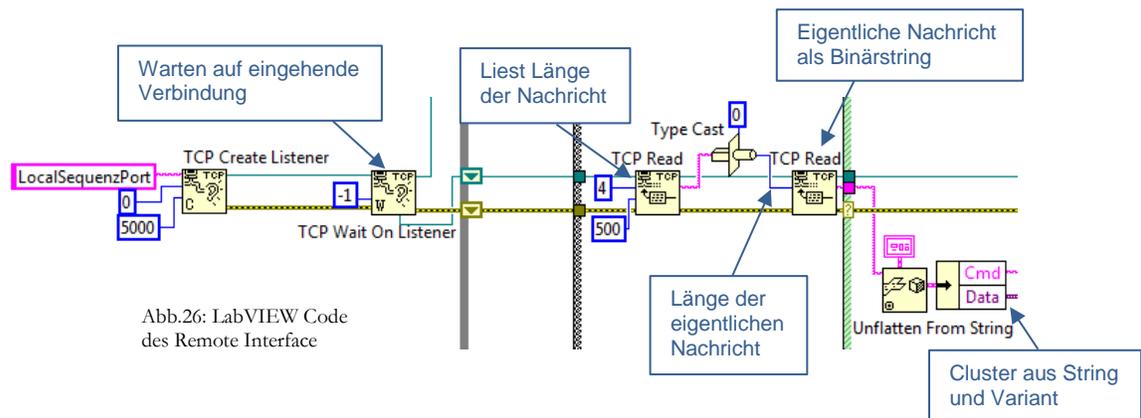
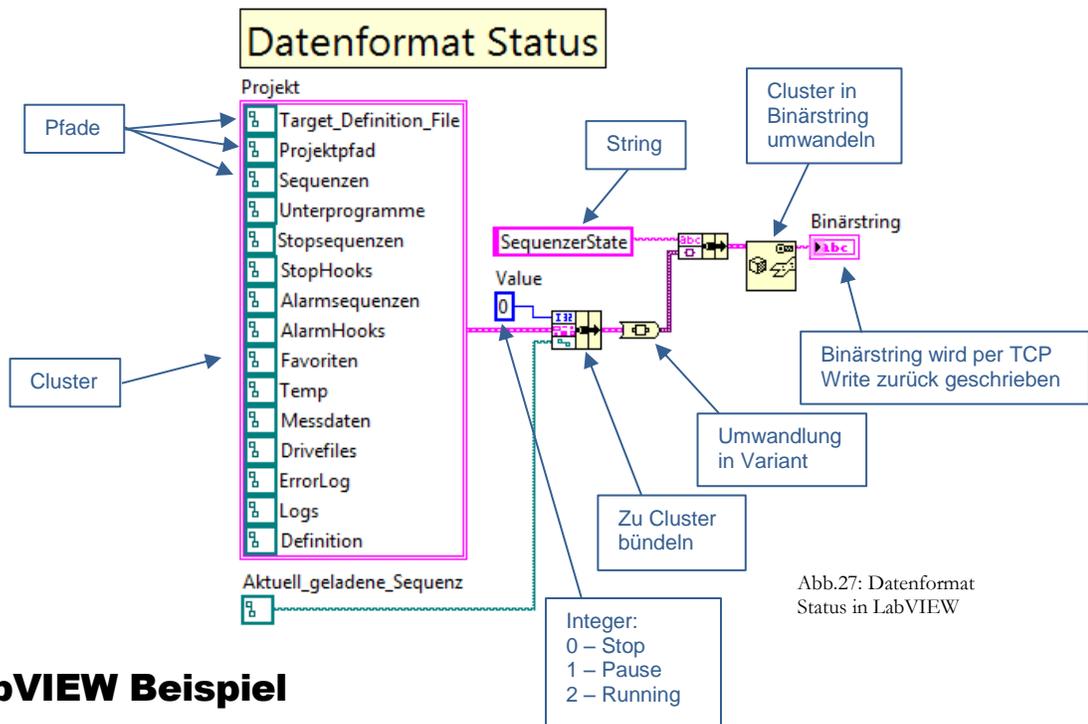


Abb.26: LabVIEW Code des Remote Interface

Ist eine TCP Verbindung hergestellt, so müssen zwei Nachrichten an das Remote Interface geschickt werden um einen Befehl auszuführen. Die erste Nachricht muss die Länge der zweiten Nachricht als Binärstring enthalten. Die zweite Nachricht enthält den Befehl und eventuelle Parameter. Das Remote Interface erwartet die zweite Nachricht ebenfalls als Binärstring. Dieser Binärstring wird im Remote Interface zu einem Cluster umgewandelt. Der Cluster besteht aus einem String- und einem Variant-Datentyp. Der String repräsentiert den Befehl und der Variant den entsprechenden Parameter. Als Parameter können Integer und Pfade auftreten.

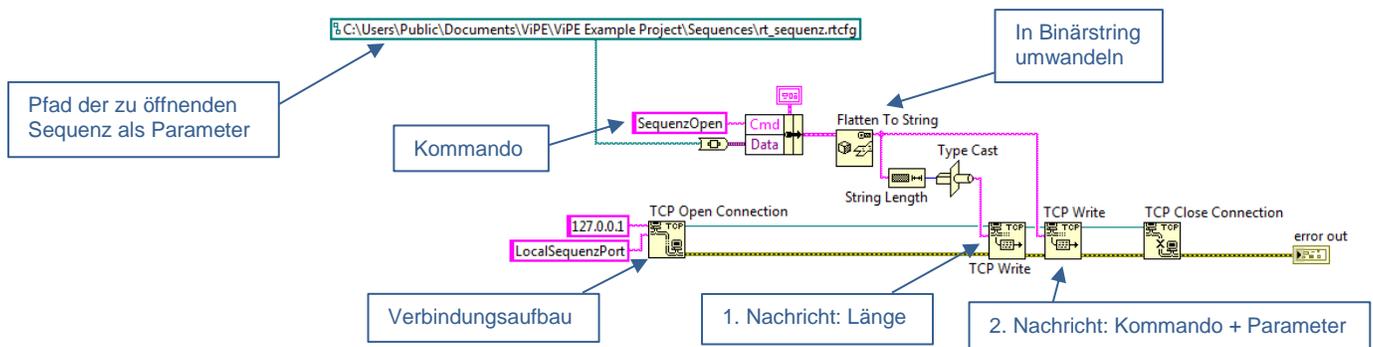
Befehlssatz

Exit	Beendet den ViPE , erwartet keinen Parameter
SequenzWindow	Maximiert ViPE -Fenster falls Hidden bzw. umgekehrt, erwartet keinen Parameter
SequenzOpen	Öffnet eine Sequenz, erwartet Pfad der zu öffnenden Sequenz als Parameter
ProjektOpen	Öffnet ein Projekt, erwartet Pfad des zu öffnenden Projekts als Parameter
ProjektNeu	Erstellt ein neues Projekt, erwartet keinen Parameter
SequenzStart	Startet aktuelle Sequenz, erwartet keinen Parameter
SequenzStop	Stoppt aktuell ablaufende Sequenz, erwartet keinen Parameter
SequenzPause	Pausiert aktuell ablaufende Sequenz, erwartet keinen Parameter
Setze_Messpfad	Setzt Messpfad, erwartet Pfad der Messdatei als Parameter
Status	Schreibt aktuellen Status per TCP Write in die TCP Verbindung, erwartet keinen Parameter. Man erhält einen Binärstring, der in einen Cluster, bestehend aus String und Variant umgewandelt werden kann. Das Variant kann wiederum in einen Cluster umgewandelt werden. Dieser Cluster besteht aus einem Integer, einem weiteren Cluster bestehend aus vielen Pfaden und einem weiteren Pfad. Im unteren Bild ist das Datenformat des Status in LabVIEW dargestellt.



LabVIEW Beispiel

Im folgenden Beispiel wird in LabVIEW das Versenden des Remote-Kommandos: „SequenzOpen“ gezeigt.



Im Beispiel wird mit der LabVIEW VI **TCP Open Connection** eine TCP Verbindung zum Remote Interface hergestellt. Das Kommando „SequenzOpen“ wird zusammen mit dem in einen Variant-Datentyp umgewandelten Pfad als Cluster weiterverarbeitet. Dieser Cluster wird in zu einem Binärstring konvertiert. Die Länge dieses Strings wird in einer ersten Nachricht per **TCP Write** an das Remote Interface geschickt. Danach wird der eigentliche Inhalt in der zweiten Nachricht auch per **TCP Write** verschickt.