CISWORKS
software company

Volume
1

USER MANUAL

CISWORKS GmbH & Co. KG

ViPE

CISWORKS GMBH & CO. KG

User Manual

Hans-Liebherr-Str. 18
88161 Lindenberg
Deutschland
Phone 0049 08381 88983-70 • Fax 0049 08381 88983-71

Table of Contents

Chapter 1

About this manual

THis manual describes the graphical real-time sequencer **CISWORKS ViPE**. The **CISWORKS ViPE** is an abstraction layer for NI VeriStand real-time sequences. It allows users to create real-time sequences without the need for profound programming skills by selecting a graphical representation for the syntax.

CONVENTIONS
Warning
Important NOTE

Conventions – In this manual the following conventions are used for visualization to indicate hints and dangers. Function Blocks are displayed in italics (see *Formula*).

Chapter

2

ViPE

Visual **P**rogramming **E**ditor.

Concept

Drag and drop blocks from the functions palette to form a flow chart. A code generator translates the flowchart into executable real-time code that runs on a NI VeriStand target. This is to enable users to create test sequences without having a profound programming experience.

During the execution, the operator can see at which point the sequence is present. It is also possible to integrate and parameterize sequences as sub-sequences created in the NI VeriStand Stimulus Profile Editor.

Installation

The installation is similar to other Windows programs. Start the installation program and follow the on-screen instructions.

The **CISWORKS ViPE** is only used to automate an existing VeriStand project. Before installing the **CISWORKS ViPE**, you must have installed **NI VeriStand.** The **ViPE** only works in combination with the VeriStand versions 2014 and 2016.

Depending on the VeriStand version used, the corresponding **ViPE** version must be installed.

To be able to automate a VeriStand project with the **CISWORKS ViPE**, the following requirements must be fulfilled:

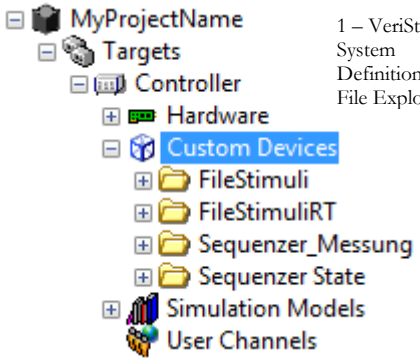
1. A **VeriStand project** must be created.

2. In the VeriStand project, the following four **Custom Devices** must be attached: **FileStimuli**, **FileStimuliRT**, **Sequenzer_Messung** and **Sequenzer State.** Furthermore the **Mapping** has to be configured. The Custom Devices and the Mapping file have been installed with the installation of the ViPE.

   To do this, proceed as follows:

   a. In the VeriStand project, select the **System Definition File** (* .nivssdf). The system explorer opens.

   b. In the System Explorer, you can find the **Custom Devices** under *MyProjectName*/Targets/Controller. Right-click on **Custom Devices** and select **FileStimuli**, **FileStimuliRT**, **Sequenzer_Messung** and **Sequenzer State**.



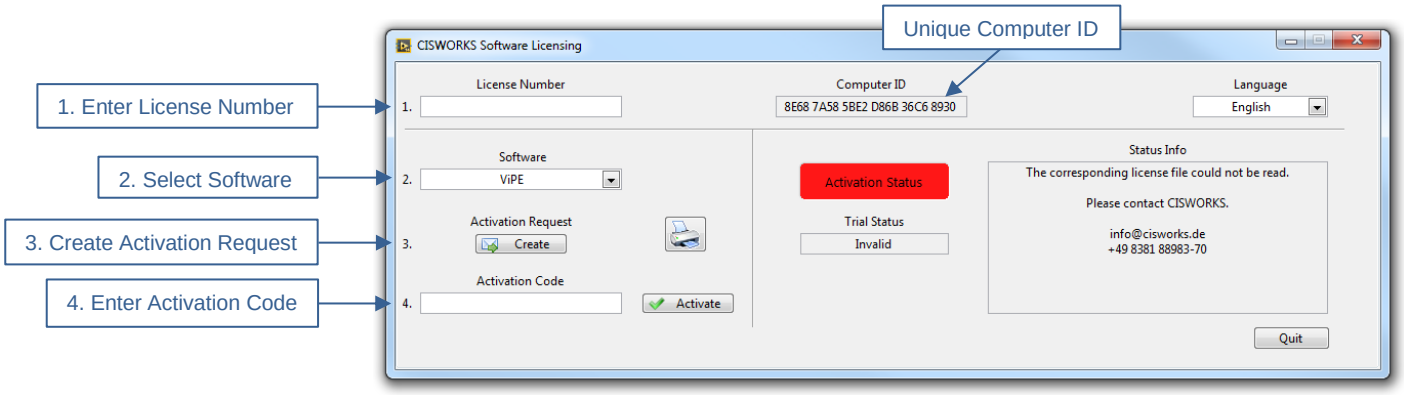
1 – VeriStand System Definition File Explorer

3. The **VeriStand project** (* .nivssdf) must be deployed.

Licensing

When you start the **CISWORKS ViPE** for the first time, you will be informed that you are using a 30-day trial version. Before this period has elapsed, you must activate the **CISWORKS ViPE** to continue using the program. In the trial version, as well as in the activated version, the full range of functions is available.

To begin with the activation process you have to start the **CISWORKS License Manager**. The **CISWORKS License Manager** is a seperate program, which has been installed with the **CISWORKS ViPE**. After that you have to go through four minor steps to complete the activation.



2 – CISWORKS License Manager

1. Please enter the received **license number** in the corresponding window at the top left.

2. Select the **software** you would like to activate.

3. Click on Create
This will create an **email** in your local client that contains all the information we need to generate an activation code. Please hit send now.

If you do not have a local client configured, send an email to activate@cisworks.de with your license number as the subject and your Computer ID and the software's name in the body.

Your Computer ID is located at the top of the **CISWORKS License Manager**. Due to the uniqueness of the Computer ID, the activation code you will receive will only be valid for this same computer. If you would like to activate the software on multiple computers, you will need to retrieve the computer ID from each machine on which you would like the software to run.

4. You will receive the **activation code** within two business days. Enter this code in the corresponding window and press Activate

You have successfully activated the software.

Requirements

Resolution

The following resolutions have been successfully tested:

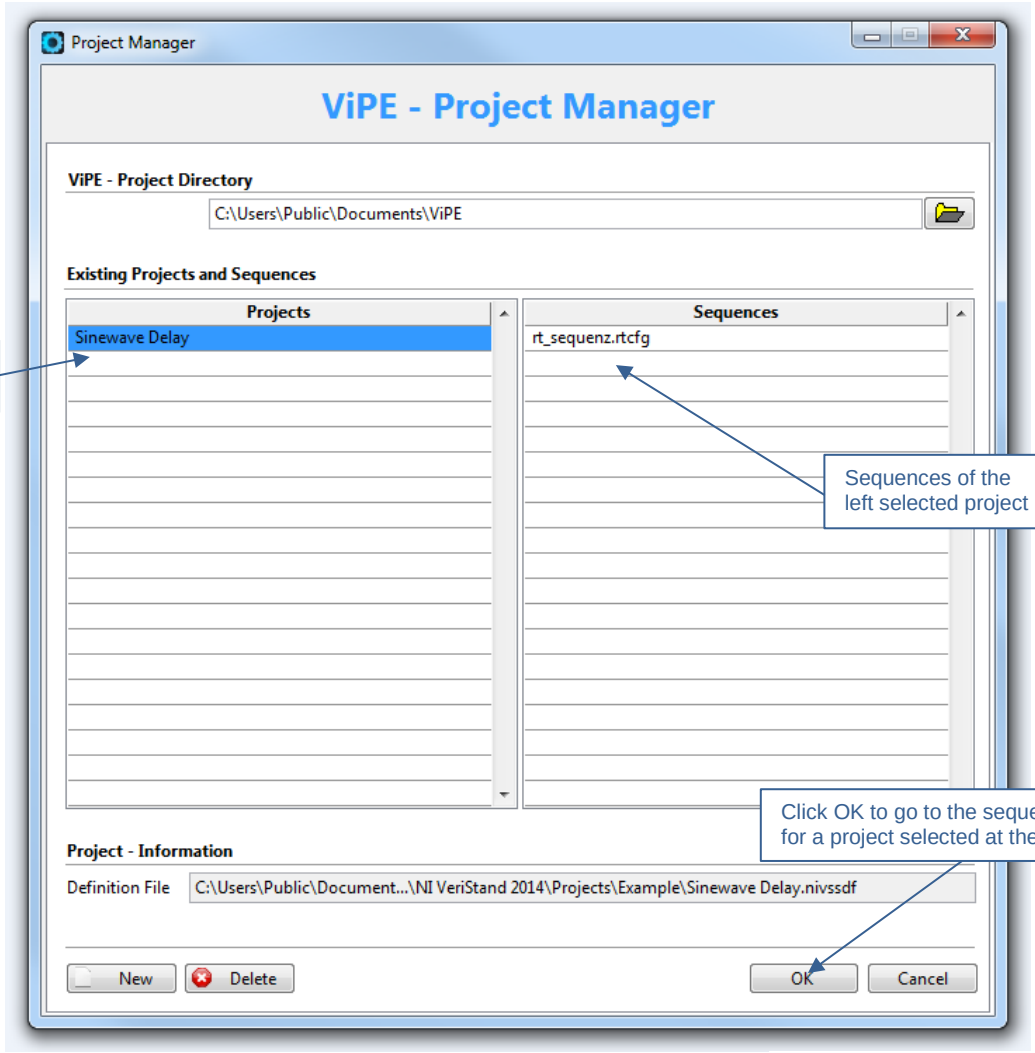
- 1920 x 1080
- 1680 x 1050
- 1600 x 900
- 1280 x 1024

These resolutions can lead to problems:

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

Project Manager

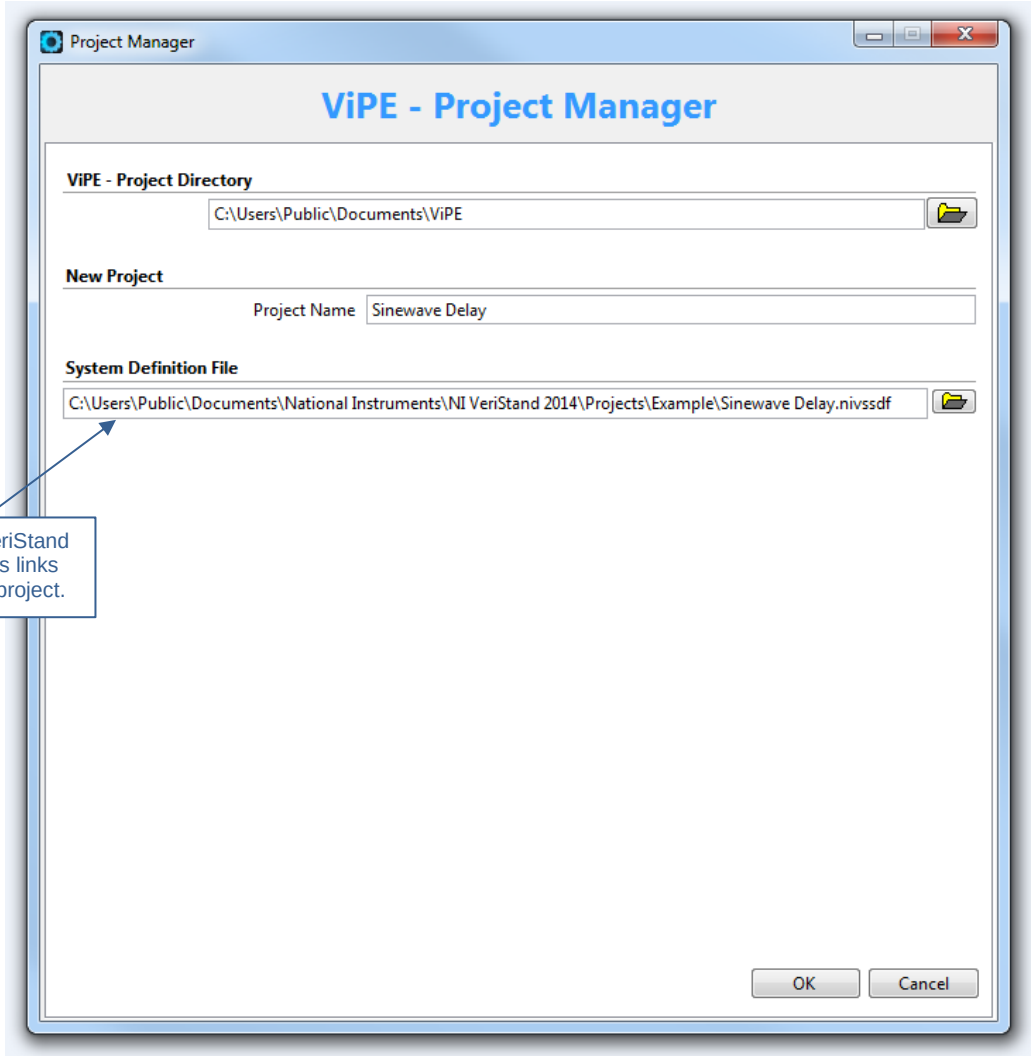
After the start of the **CISWORKS ViPE**, the project manager appears. This project manager is used for managing projects and linking **CISWORKS ViPE** and NI VeriStand.



3 – ViPE Project Manager

Creating a project in **CISWORKS ViPE** is a necessary step to create and run real-time sequences. Individual sequences must always be assigned to a project.

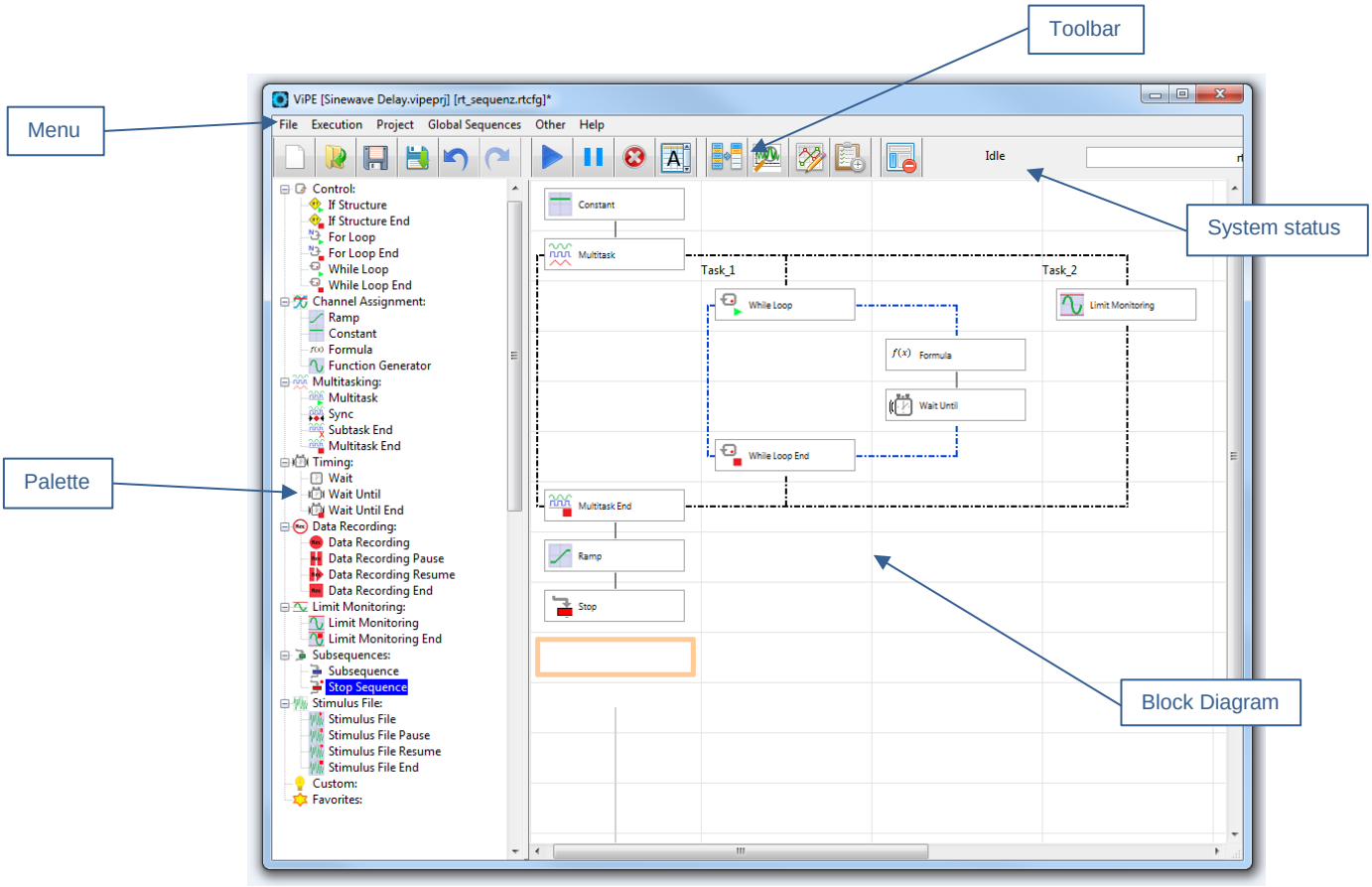
To create a new project, click **New** in the lower left corner of the project manager. The following dialog opens.



4 – ViPE Project Manager New Project

Sequence Editor

The sequence editor consists of five different parts and provides the functionality required to create and edit sequences.



5 – ViPE Sequence Editor

Menu

The menu is used for general control and administration of sequences and projects. It is divided into the six sub-items: File, Execution, Project, Global Sequences, Other and Help.

The functionality of the menu is explained in more detail below.

FILE

New	Create a new file
Open …	Open an existing file
Add to the project	Add current sequence to any project
Save	Save current file
Save as …	Save current file to any directory
Exit	Exit ViPE

EXECUTION

Compile	Compile sequence
Run	Start sequence
Pause	Pause sequence
Stop	Stop sequence

PROJECT

New	Create new project
Open …	Open existing project
Save	Save current project
Save as …	Save current project to any directory

GLOBAL SEQUENCES

The intention of global stop and alarm sequences is the protection of the devices. If sequences are interrupted or certain limit values are exceeded, a controlled termination or shutdown should be carried out in order not to damage any devices.

Configure Global Stop Sequences	Open the global Configure Stop Sequences window
Configure Global Alarm Sequences	Open the global Configure Alarm Sequences window
Save Stop Sequence	Saves the current sequence as a global stop sequence
Save Alarm Sequence	Saves the current sequence as a global alarm sequence

Global stop and alarm sequences must be saved in the default directory in order to select them later.

Open Stop Sequence	Open existing stop sequence
Open Alarm Sequence	Open existing alarm sequence

The scope of Pre 1, Pre 2, Post 1 and Post 2 **Stop Sequences** is global. This means that these sequences are executed before and after all local Stop Sequences. Running **Order sequence**: Pre 1 - Pre 2 - local Stop Sequence - Post 1 - Post 2.

The execution of global Alarm Sequences depends on the selected Limit Monitoring Mode. Depending on the mode either none, Pre 1 and Pre 2 or all Global Alarm Sequences are executed together with the local Alarm Sequence. (see Limit Monitoring)

OTHER

Language	English oder german
Shortcuts	List of available shortcuts

HELP

Manual	Detailed user manual
Training Videos	Youtube links

Toolbar

To control and work with the **CISWORKS ViPE**, a series of buttons are used to support the user in his or her request. These buttons are presented with self-explaining icons, which provide the most important functions directly.

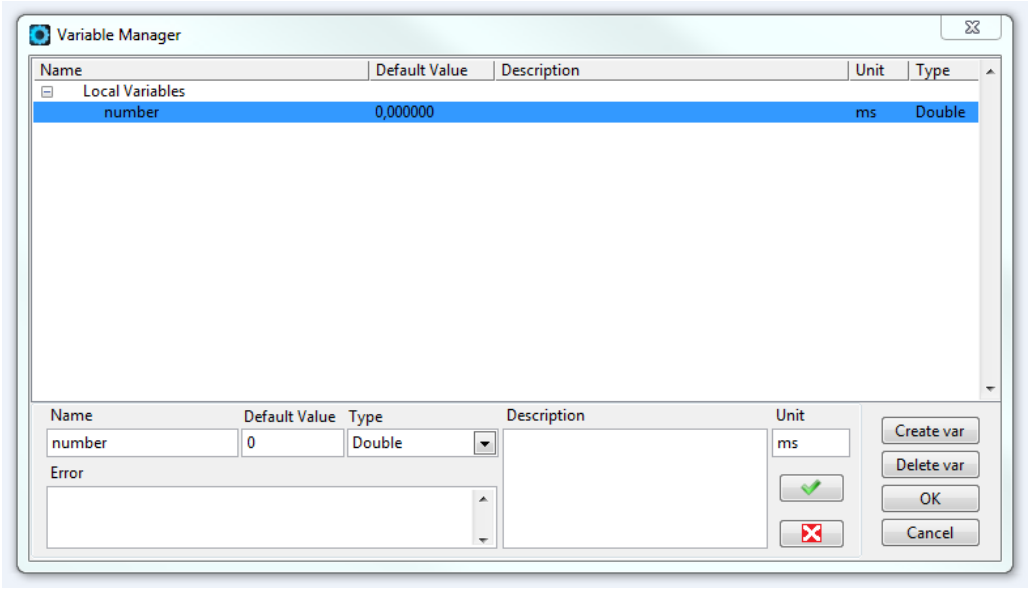
OVERVIEW

	**Create new file**	Create a new sequence
	**Open file**	Open an existing sequence
	**Save file**	Save current sequence
	**File – Save as**	Save current sequence under a specific path

 **Undo**	Undo the last action
 **Redo**	Restore an undo action
 **Start Sequence**	
 **Pause Sequence**	
 **Stop Sequence**	
 **View follows sequence execution**	Focus on the currently executed step of a running sequence
 **Variable Manager**	Create and edit local variables
 **TDMS Viewer**	View TDMS-Files
 **Stimulus File Editor**	Create Stimulus File and save as a *.csv file
 **Status Info**	Status information about the running sequence
 **Show/Hide Palette**	

The toolbar functions **Variable Manager**, **TDMS Viewer** and **Stimulus Editor** are explained in more detail below.

VARIABLE MANAGER



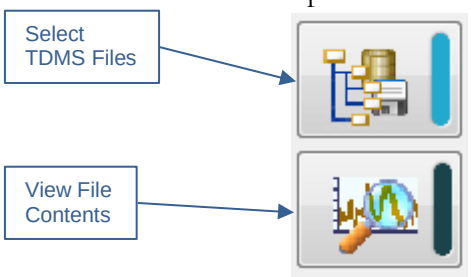
6 – ViPE Variable Manager

With the help of the variable manager local variables can be created and provided with a default value as well as a unit. Local variables can be accessed from function blocks.

Local variables have a local scope, which means that they can only be used in the sequence in which they were created. In addition, the name of a variable must be unique and must not contain any of the following characters: __ . , - ; : ! /

TDMS VIEWER

Recorded data is stored in TDMS format. With the TDMS Viewer, these files can be opened and viewed as graphs.



To do this, you have to move the desired TDMS file from the *Select TDMS Files* menu into the right window.

Afterwards, you switch to the *View File Contents* menu and draw the desired measurement criterion from the right side into the graph.

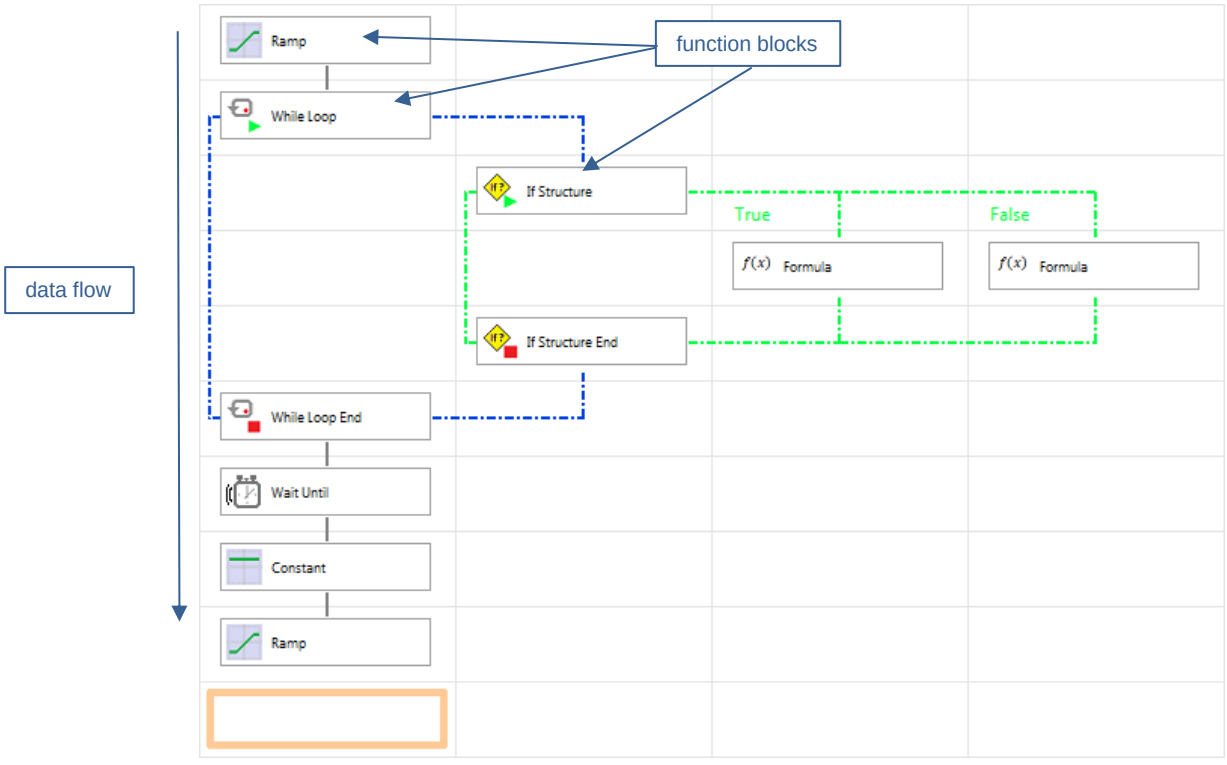
STIMULUS EDITOR

In the **stimulus editor**, curves can be generated and saved as a CSV file for further use. The frequency, amplitude, and offset parameters specify the waveform of the curve. File length and target rate specify the number of points. There are three types of curves available: sine, rectangle and noise.

Note that the target rate of the generated curves must match the target rate of the VeriStand target.

Block Diagram

Graphical representation of the sequence, as in a flowchart. The function blocks of the pallet can be dragged into the block diagram. The data flow moves from top to bottom. During execution, the user can see which block of the sequence is currently executed.



7 – ViPE Block Diagram

Palette

Function blocks can be easily added to the block diagram by drag & drop. In the subsequent configuration window, this block can be specifically configured according to its function. In addition to basic sequence control functions, the palette also provides functionality to capture measurement data and to define alarm sequences that are executed when limits are exceeded.

BUTTONS

In some function blocks mathematical functionality is provided by buttons. The use of these buttons is explained in more detail below.

	Round brackets	Function parameters are enclosed with round brackets
		
	Square brackets	Variables, channels and aliases are enclosed with square brackets
		
	Assignment	Assignment operator
	Greater than	Relational operator
	Less than	Relational operator
	Greather than or equal to	Relational operator
	Less than or equal to	Relational operator

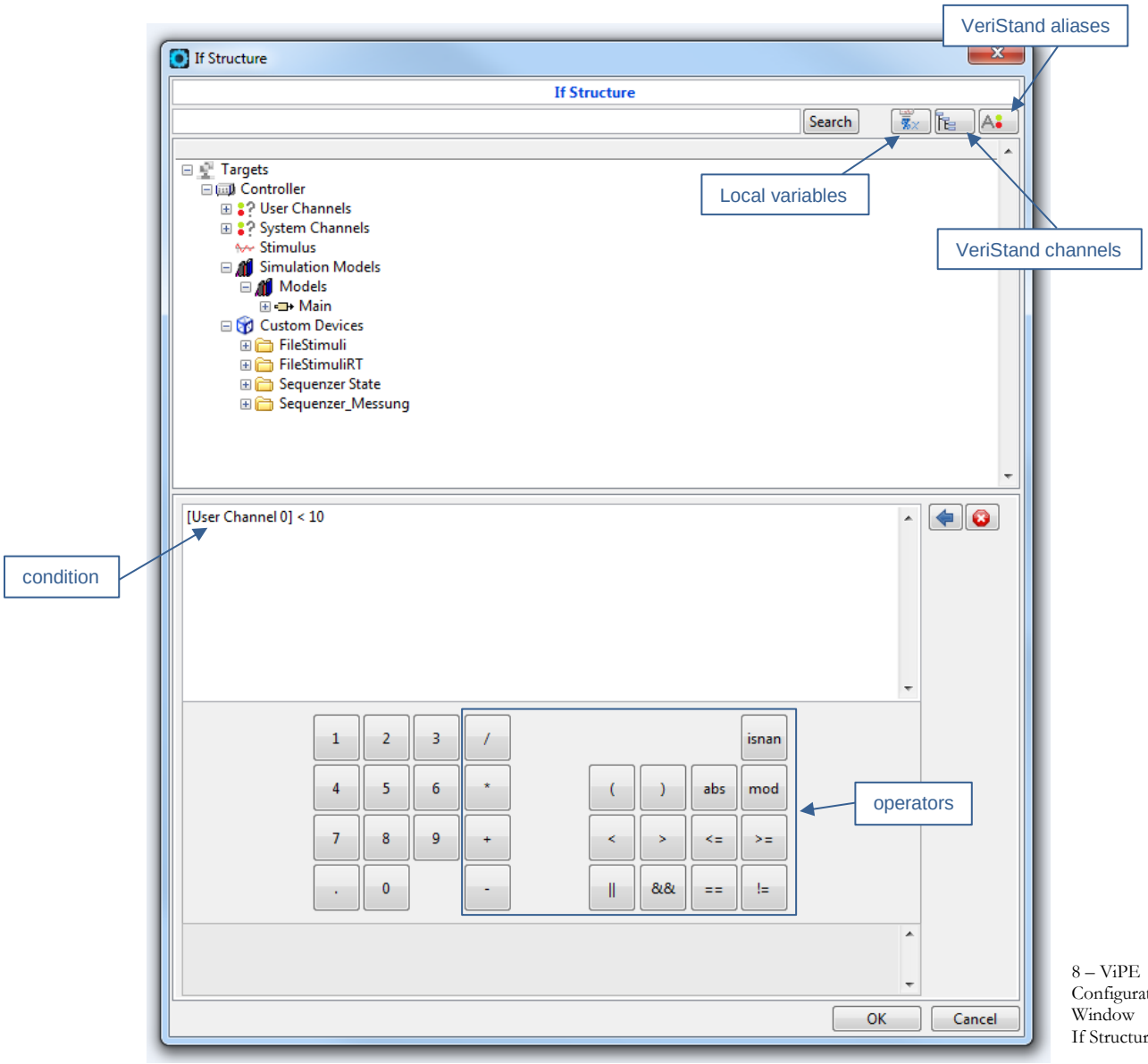
	Or	Logical or operation
	And	Logical and operation
	Equal to	Relational operator
	Not equal to	Relational operator
	Sine	Use: sin(x) with x = angle in degree
	Cosine	Use: cos(x) with x = angle in degree
	Tangent	Use: tan(x) with x = angle in degree
	Root	sqrt(x) returns square root of x
	Exponential function	Exponential funktion with the constant e as base Use: exp(x) with x as exponent
	Exponentiation	Use: pow(x,y) with x as base and y as exponent
	Logarithm	Use: log(x,y) with y as base
	Natural logarithm	Logarithm with base e, Use: ln(x) with x = exp(ln(x))
	IsNotANumber	Returns TRUE if the value is not a number
	Absolute	abs(x) returns absolute value of x
	Arccosine	Use: acos(x) with x between -1 and 1 returns angle in radian

	Arcsine	Use: acos(x) with x between -1 and 1 returns angle in radian
	Arctangent	Use: atan(x) returns angle in radian
	Modulo	Use: mod(x,y) returns rest of x ÷ y
	Random number	Use: rand(x) returns random number between 0 and x
	Sign	Use: sign(x) returns 1 with x > 0, -1 with x < 0, 0 with x = 0

Control

If Structure

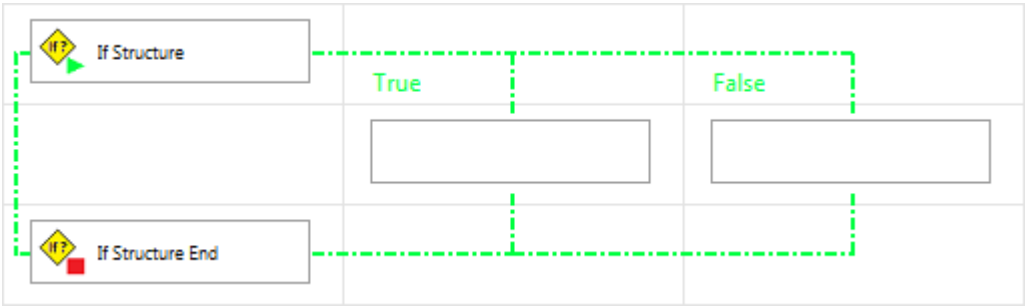
After an *If Structure* is drawn into the block diagram, the configuration window, in which a condition can be defined, opens.



8 – ViPE Configuration Window If Structure

Local variables, VeriStand channels, or VeriStand aliases can be used to define the condition. You can use the search function to find specific variables, channels and aliases. To set or edit the condition, you can use the numeric or operator buttons as well as the keyboard.

When executing the sequence, the condition is checked. At a branch, either the true path or false path is executed depending on whether the condition is met or not.



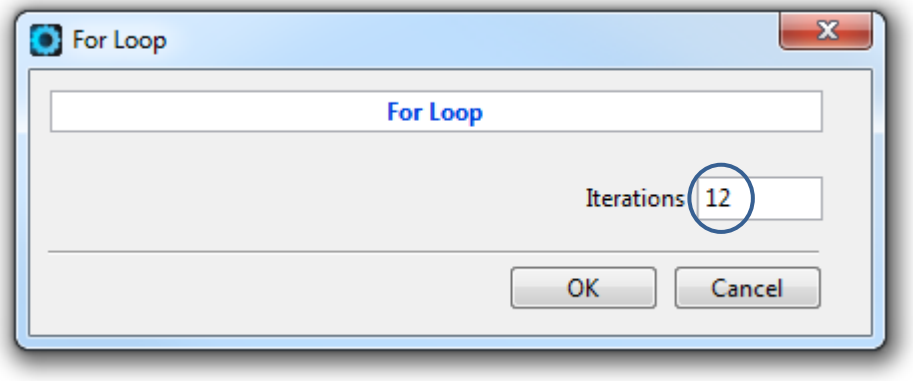
9 – ViPE If Structure

If Structure End

An *If Structure* must always end with an *If Structure End*. If you have accidentally removed the *If Structure End* block, you can readd it by simply dragging the *If Structure End* to the block diagram.

For Loop

The content of a *For Loop* is repeated a certain number of iterations. The number of iterations is set to 12 in the example below.



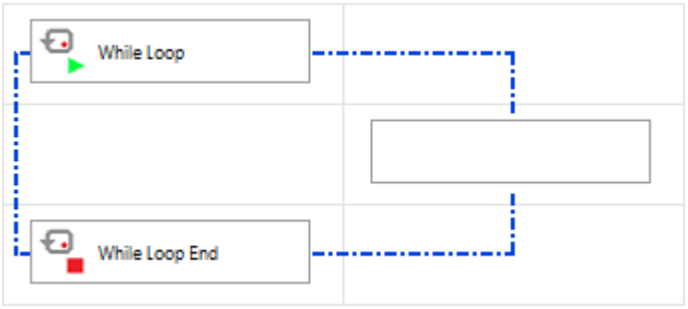
10 – ViPE For Loop

For Loop End

A *For Loop* must always end with a *For Loop End*. If you have accidentally removed the *For Loop End* block, you can readd it by simply dragging the *For Loop End* to the block diagram.

While Loop

The content of a *While Loop* is repeated as long as a defined condition is met. If the condition is never false, *the While* Loop never stops.



11 – ViPE While Loop

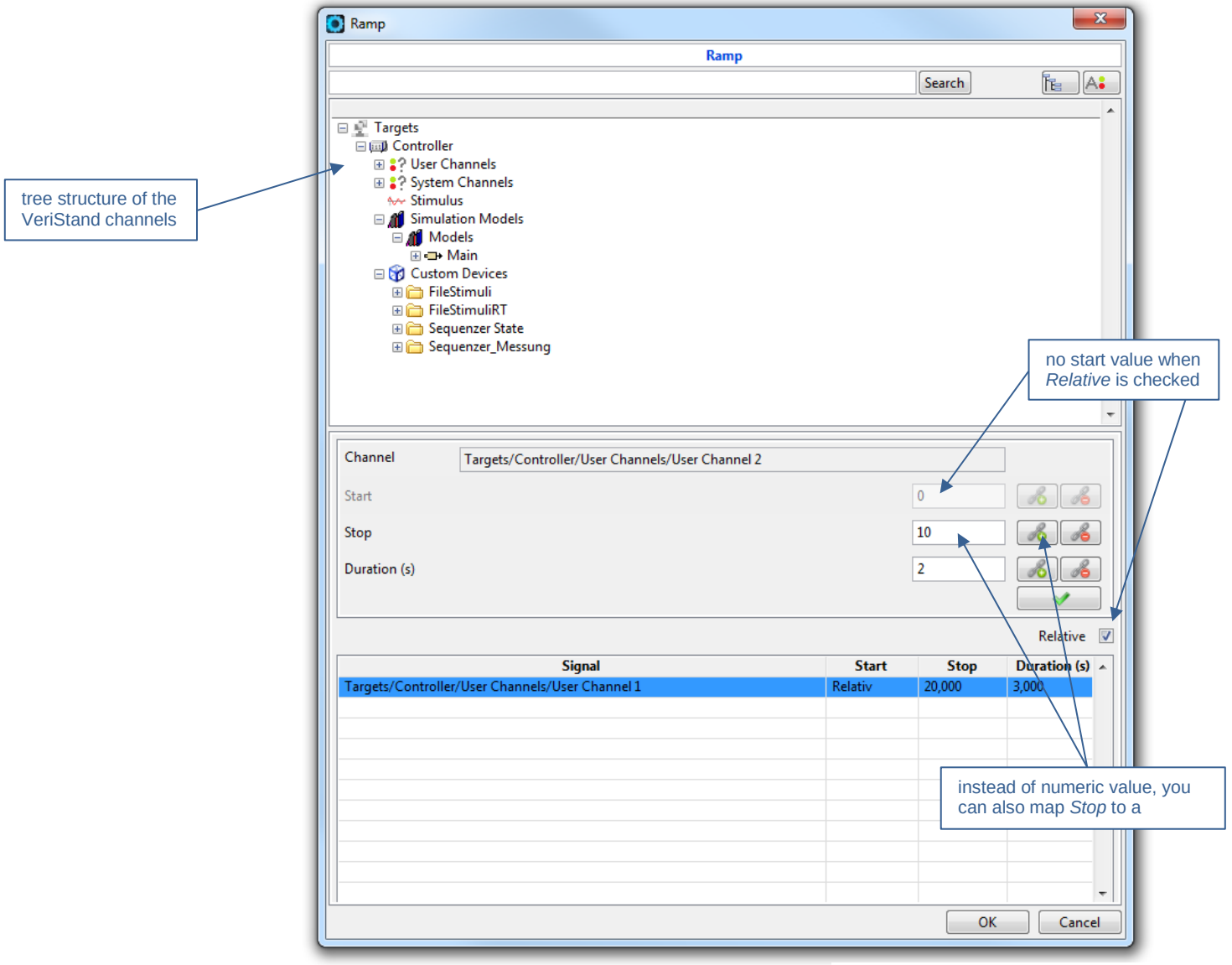
While Loop End

A *While Loop* must always end with a *While Loop End*. If you have accidentally removed the *While Loop End* block, you can readd it by simply dragging the *While Loop End* to the block diagram.

Channel Assignment

Ramp

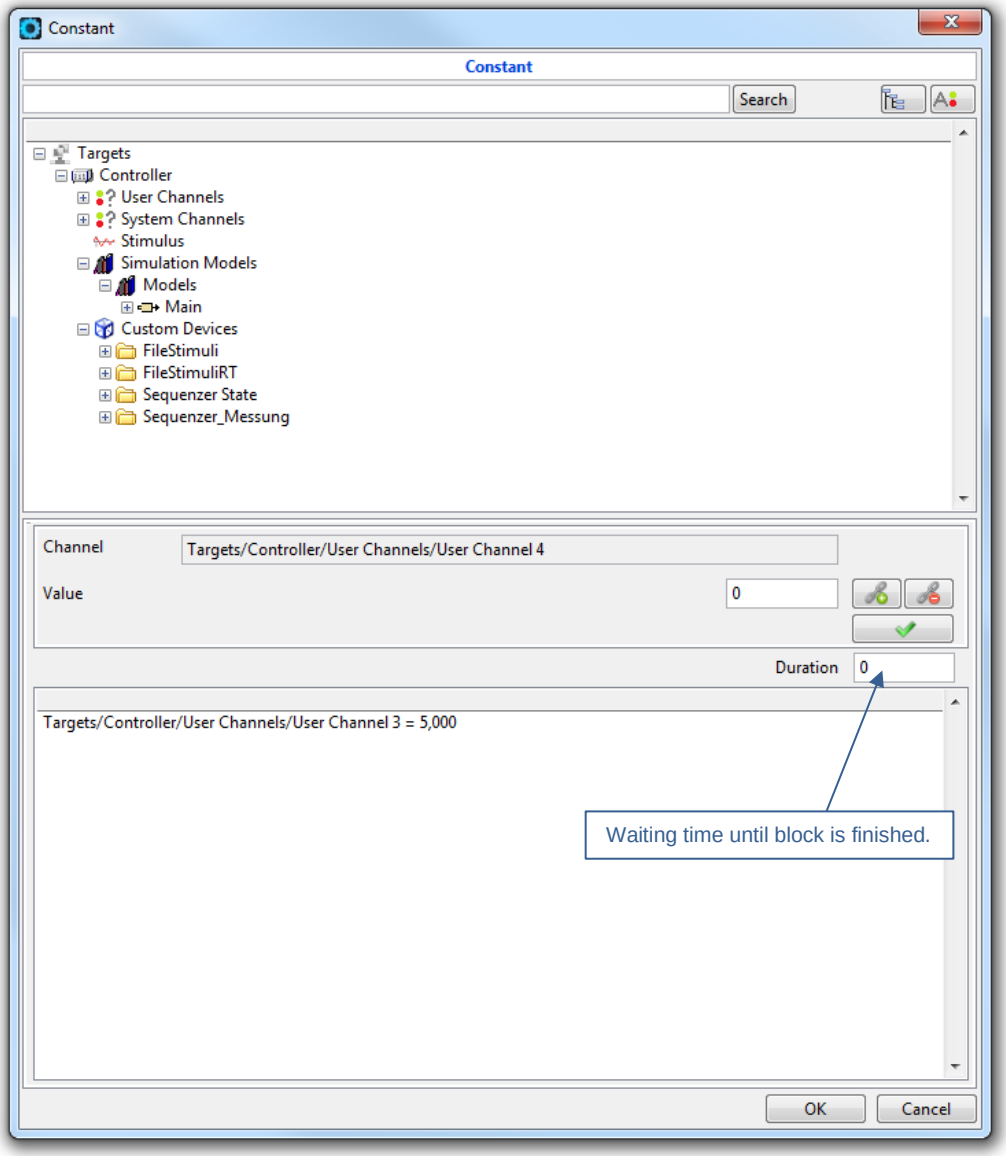
The value of a channel to be selected changes linearly within a time span from a start to a stop value. It is possible to specify the duration as well as start and stop value numerically or to map these to a channel. Channels and aliases can be used as channels. Furthermore, you can set the starting value absolute or relative. A relative start value means that the current value of the channel is the start value.



12 – ViPE Configuration Window Ramp

Constant

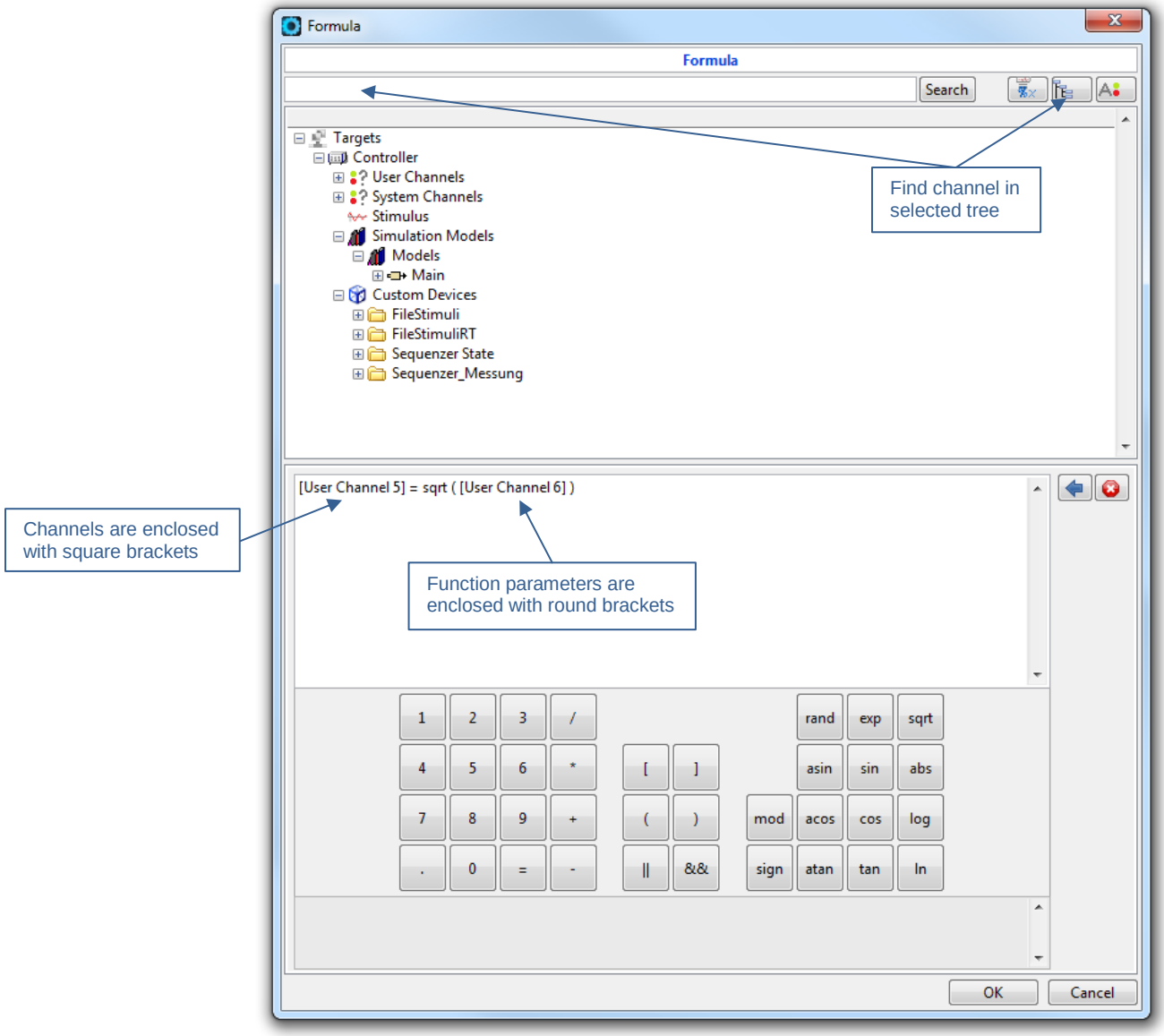
A specific value can be assigned to each channel. These values are taken directly when the constant function is executed. Duration specifies the time how long the system waits until the function is terminated.



13 – ViPE Configuration Window Constant

$f(x)$ Formula

A **Formula** can be applied in many ways. The values of variables, channels or aliases can be manipulated with the help of operators and functions.



14 – ViPE Configuration Window Formula

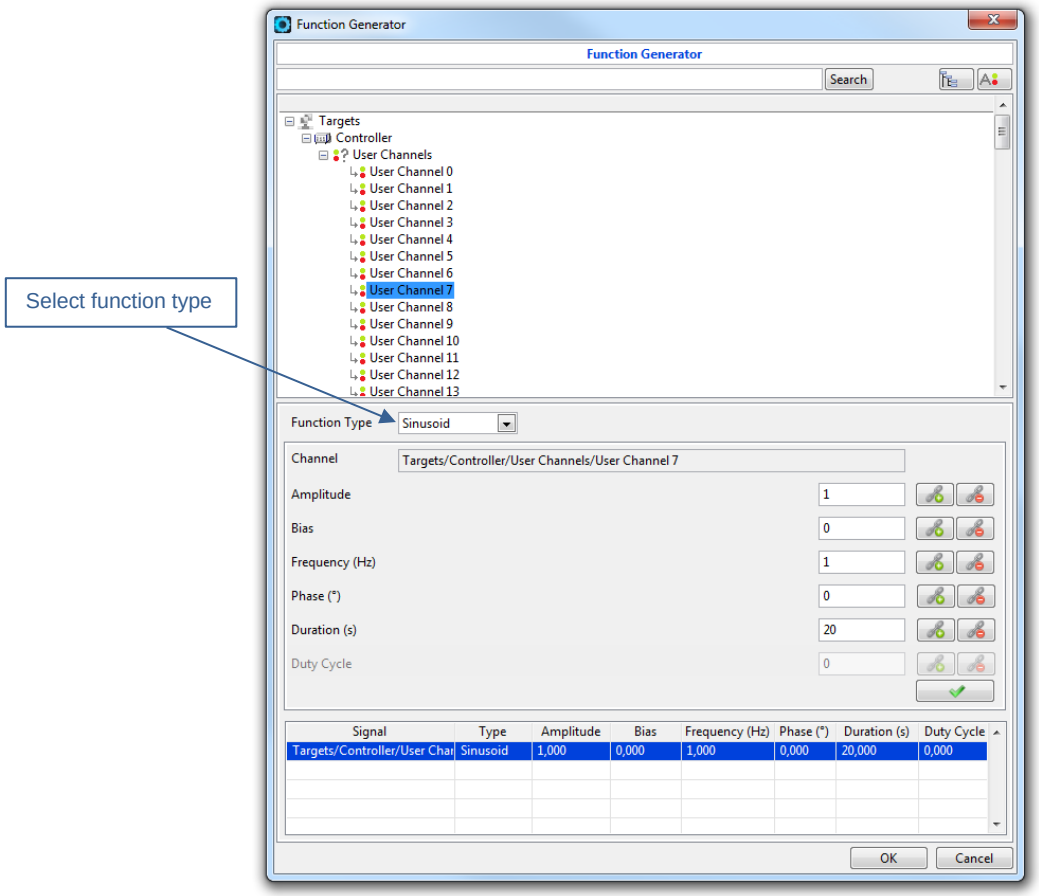
Function Generator

Functions can be assigned to channels. There are four types of functions available:

Sinusoid, **Sawtooth**, **PWM** (Pulse-width modulation), **Triangle**

The following settings can be made for each function:

Amplitude	Maximum deflection in the y-direction (positive and negative range) from the x-axis
Bias	Offset which is added to y. x-axis is set to y = 0 by default
Frequency	Speed of successive repetitions in Hertz
Phase	Start position of the function in angle (degree)
Duration	Duration of the function in seconds
Duty Cycle	Ratio of *Amplitude - Bias* to *Amplitude + Bias* as a percentage (only relevant for PWM)

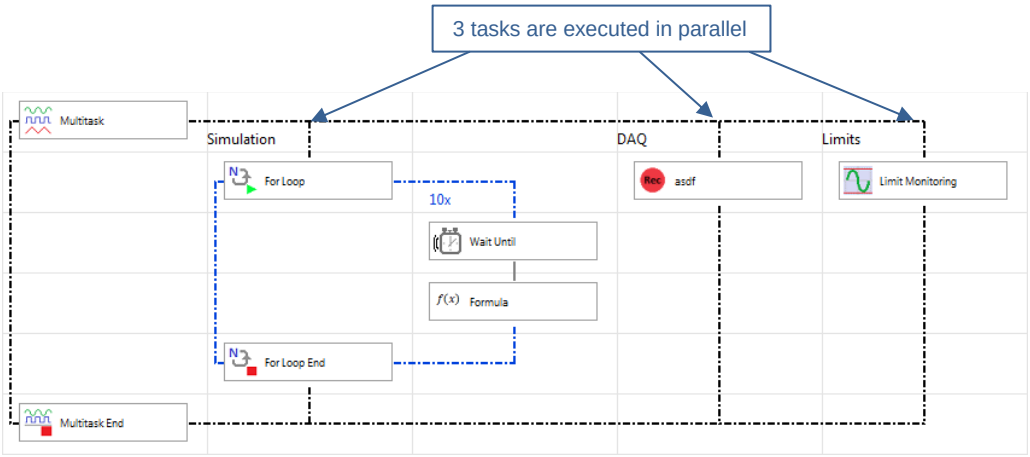


15 – ViPE Configuration Window Function Generator

Mulitasking

Multitask

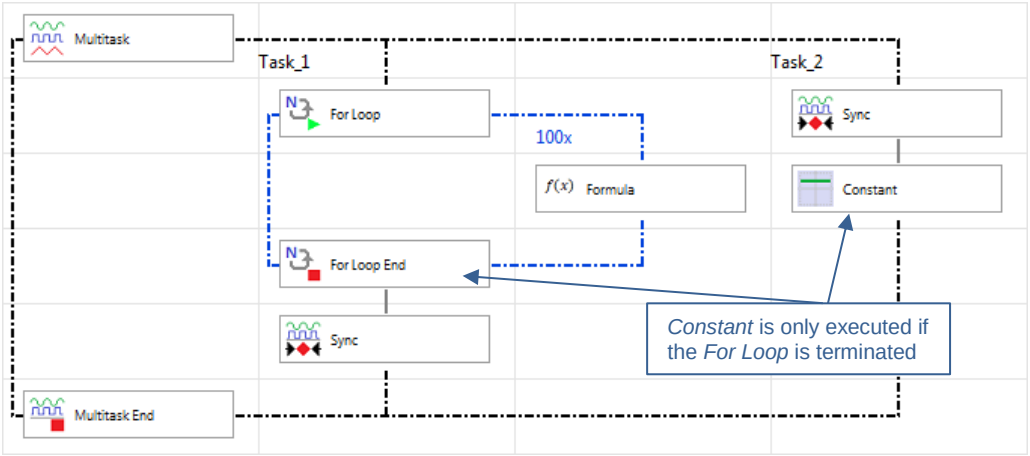
Up to 8 instruction blocks can be executed in parallel in so-called tasks.



16 – ViPE Multitask

Sync

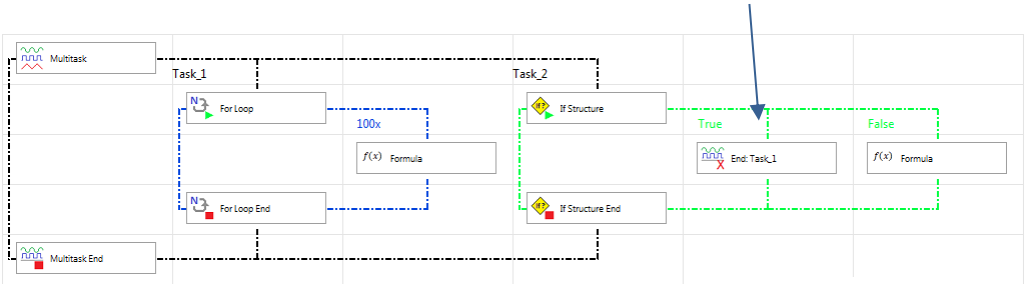
Within *Multitask* blocks, the sync function can be used to specify that all tasks must first have reached the *Sync* block before subsequent function blocks can be executed. In the example below, *Constant* in Task2 is not executed until the *For Loop* block in Task1 is terminated.



17 – ViPE Sync

Subtask End

Within a multitask block, a task can be terminated from another task.



18 – ViPE Subtask End

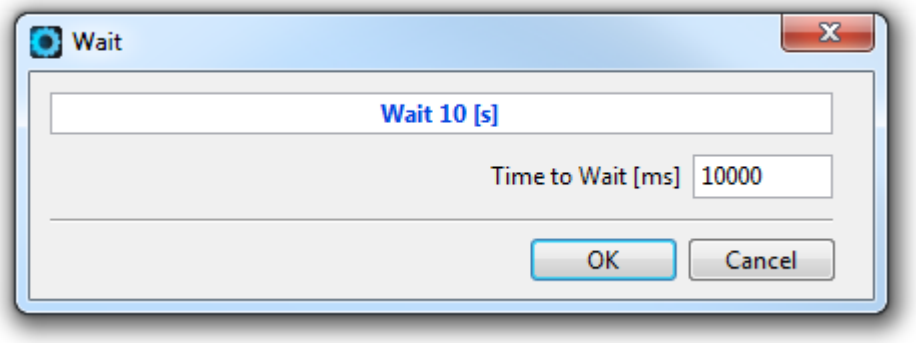
Multitask End

A *Multitask* must always end with a *Multitask End*. If you have accidentally removed the *Multitask End* block, you can readd it by simply dragging the *Multitask End* to the block diagram.

Timing

Wait

Waiting for a certain amount of time. The lower block waits for 10 seconds until the next function block is executed.



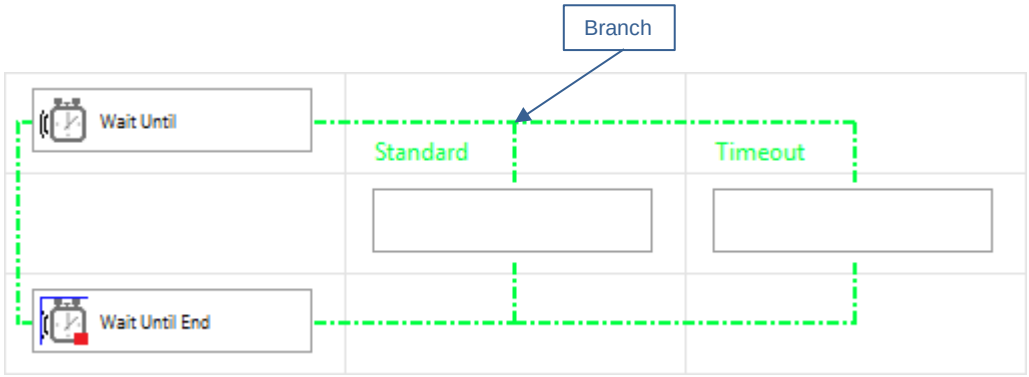
19 – ViPE Wait

Wait Until

Wait until a defined condition is met. It is possible to define a timeout, which terminates the function after a certain time, if the condition is not met.

Wait Until End

If a timeout is set in the *Wait Until* block to, a branch in Standard and Timeout results. This branch must always be completed by *Wait Until End*. If the condition occurs within the defined period, the block is executed within Standard. If this condition does not occur within the defined time period, the timeout occurs, and then the block is executed within the Timeout.

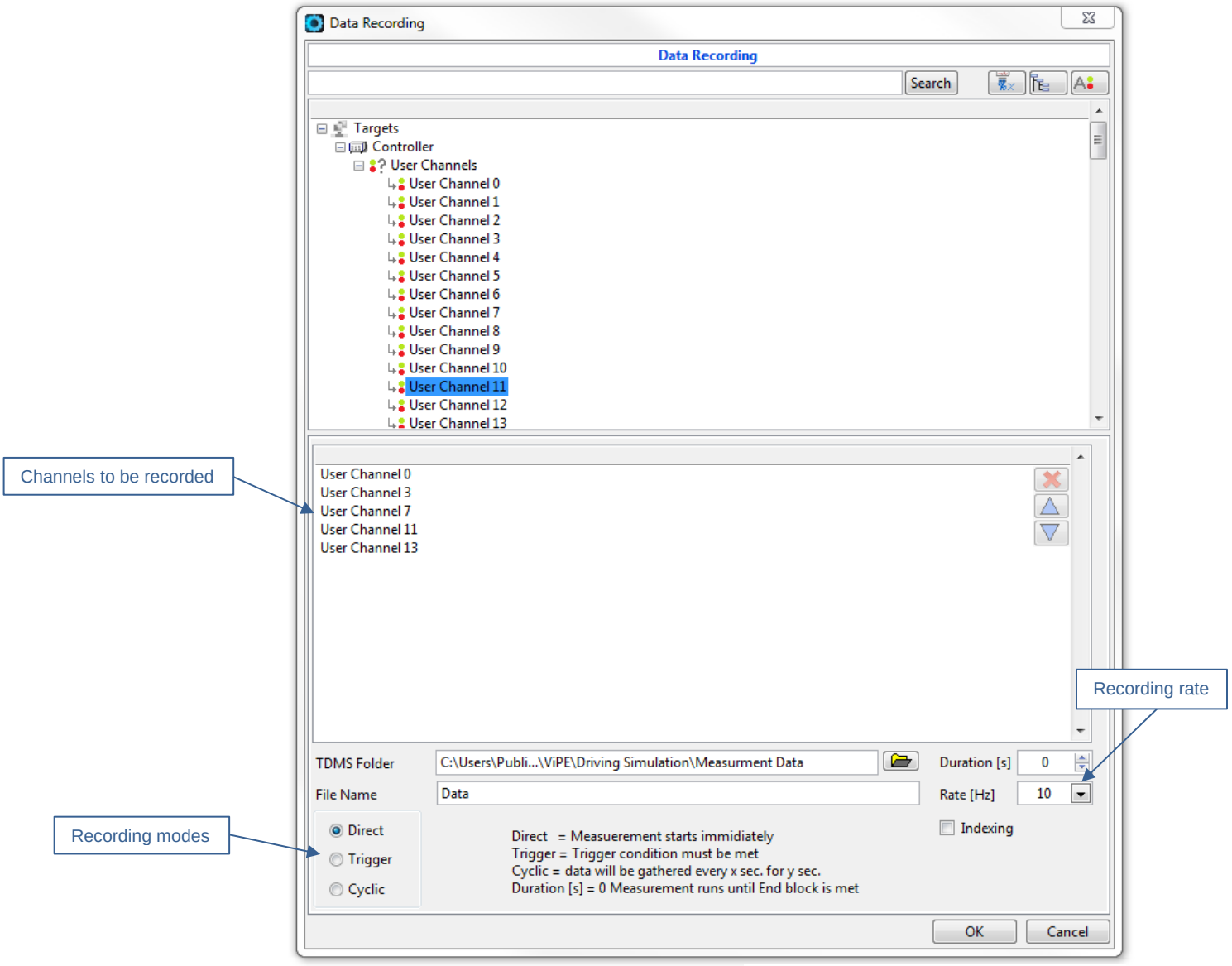


20 – ViPE Wait Until End

Data Recording

Data Recording

Values of variables, channels and aliases can be recorded and saved in a TDMS file. TDMS is a data format of National Instruments, which can be opened directly with the TDMS viewer in **CISWORKS ViPE**. The channels to be recorded are added by double-clicking.



21 – ViPE Configuration Window Data Recording

Configuration options

Duration Duration [s] = 0 means, that recording runs until End block is met or the sequence is terminated

Rate recording frequency in hertz

Indexing If *Data Recording* is repeatedly called within a *For Loop* or a *While Loop* and the checkmark is set, a new file is created with a consecutive number appended to the recording file name.

Recording modes

Direct Recording starts immediately with the set frequency

Trigger Recording starts only when the trigger condition is met. A trigger condition can be defined for a channel or for an alias. The following operators are available: > (larger), > = (greater than), <(smaller), <= (less than), == (equal), - || - (open interval), |--| (completed interval)

Cyclic Data is recorded for the period of the value specified in the measurement period. The data recording is then interrupted for the period of the value specified in the cycle time. This process is repeated for the duration of the data recording.

Data Recording Pause

The data recording is interrupted.

Data Recording Resume

An interrupted data recording is resumed.

Data Recording End

The data recording is ended.

Data Recording Pause and *Data Recording End* are not real-time. This means that after the execution of these blocks further data can be recorded for several milliseconds.

Limit Monitoring

Limit Monitoring

A limit condition can be defined. If the condition is met, a selected alarm sequence is started.

Note that alarm sequences need to be saved in the designated folder (…\Sequences\Alarm Sequences). Only then can these be selected.

There are four different modes for executing this alarm sequence:

Continuous

Whenever the limit condition is met, the Alarm Sequence is called. After the Alarm Sequence has been processed, the current sequence continues. No Global Alarm Sequences are called before and after the Alarm Sequence.

Pre 1 –> Pre 2 –> **Alarm Sequenz** –> Post 1 –> Post 2

Single

The Alarm Sequence is only called upon the first time the limit condition is met. After the Alarm Sequence has been processed, the current sequence continues, even if the limit condition is met another time. No Global Alarm sequences are called before and after the Alarm Sequence.

Pre 1 –> Pre 2 –> **Alarm Sequenz** –> Post 1 –> Post 2

Stop Sequence

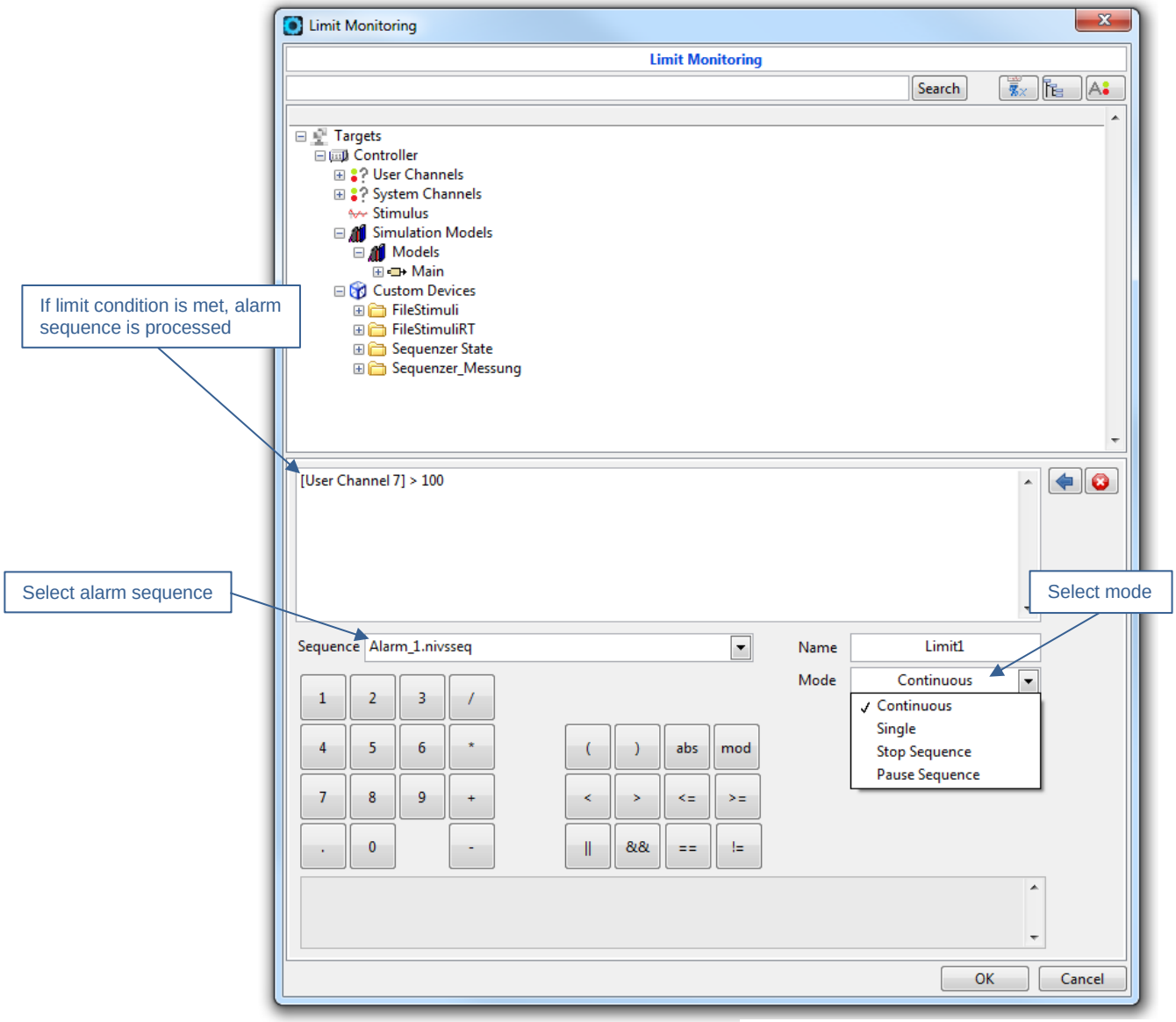
After the Alarm Sequence has been processed, the sequence is aborted. The two Global Alarm Sequences Pre Alarm 1 & 2 are called before this Alarm Sequence.

Pre 1 –> Pre 2 –> Alarm Sequenz –> Post 1 –> Post 2

Pause Sequence

After the alarm sequence has been processed, the sequence is paused. The sequence must be continued manually by the user. Before and after this Alarm Sequence all Global Alarm Sequences are called.

Pre 1 –> Pre 2 –> Alarm Sequenz –> Post 1 –> Post 2



22 – ViPE Configuration Window Limit Monitoring

Limit Monitoring End

Terminates an active limit monitoring. If there are more than one limit monitorings active, you can select the one to terminate.

Sub-sequences

Subsequence

An existing sequence can be integrated into the current sequence as a subsequence. Not only CISWORKS ViPE sequences (*.rtcfg) can be called but also VeriStand sequences (*.nivsseq).

Stop Sequence

A **Stop Sequence** can be included at any point in the current sequence. This stop sequence is only executed if the current running sequence is aborted manually by the user.

If a subsequence is included in a sequence with a stop sequence in which another stop sequence is defined, the stop sequence from the calling sequence is executed in the event of an abort.

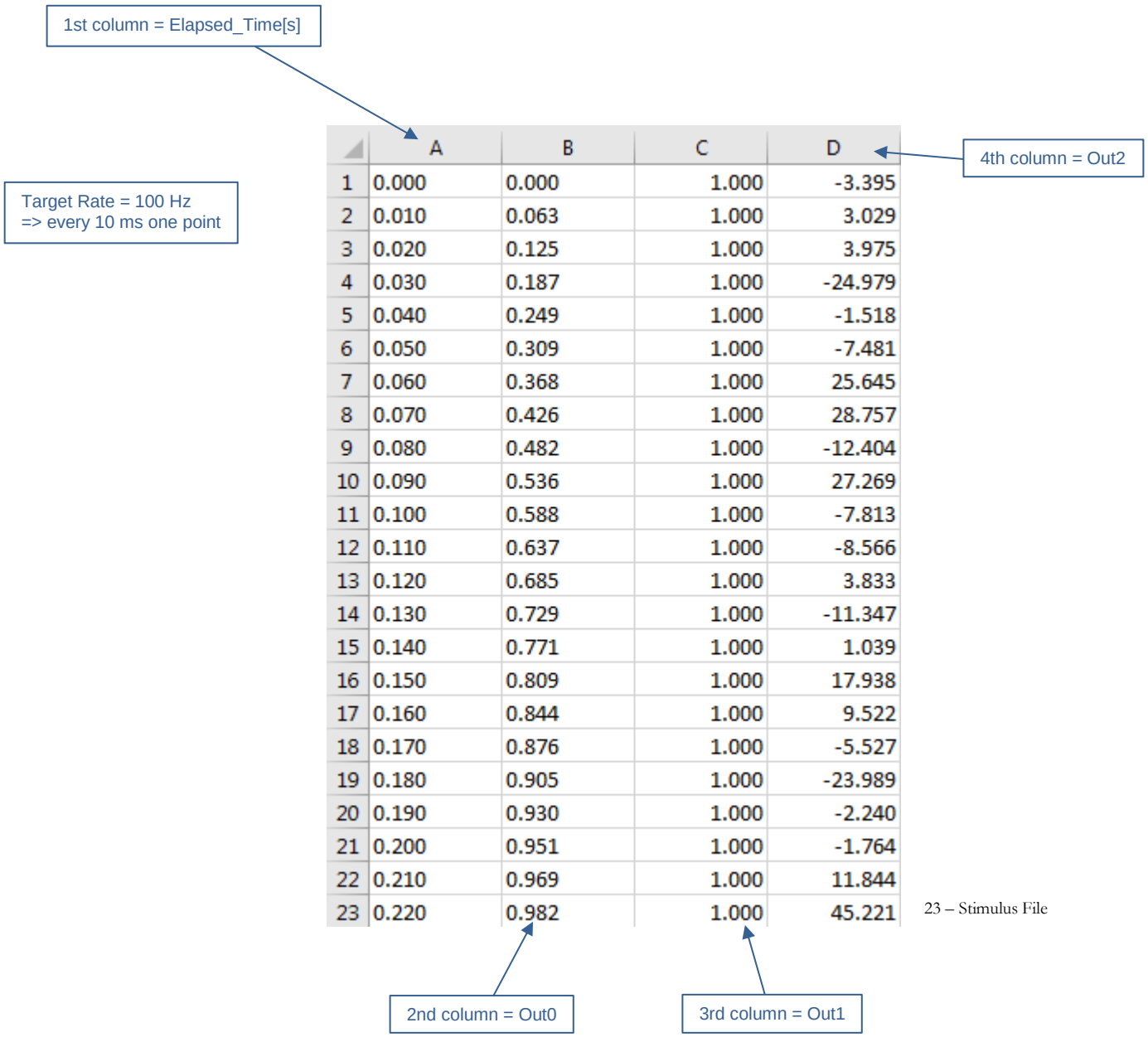
Stimulus File

Stimulus File

As Stimulus Files, you can load files (*.txt, *.csv, *.dsv) that were previously created in the Stimulus Editor. The values of the FileStimuli can be accessed in the VeriStand channels under Custom Devices/FileStimuliRT/Output. The access to the values of the individual columns is made via the following VeriStand channels (all channels under Custom Devices/FileStimuliRT/Output):

Stimulus File	1st column	2nd column	3rd column	4th column	…
VeriStand channel	Elapsed_Time[s]	Out0	Out1	Out2	…

The following example shows a **Stimulus File** (*.csv) with three curves:
Column B = **Sine** (Frequency = 1, Amplitude = 1, Offset = 0)
Column C = **Rectangle** (Frequency = 1, Amplitude = 1, Offset = 0)
Column D = **Noise** (Frequency = 1, Amplitude = 1, Offset = 0)



	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

23 – Stimulus File

The processing of a *Stimulus File* takes place after a waiting time of 2 seconds in the background of a sequence. The values are read for the duration specified in the stimulus file. However, the start and stop position can also be set manually. If the execution of the sequence is terminated, the reading of the stimulus file also ends. The reading of the values can be manipulated with the following function blocks.

Stimulus File Pause

Interrupt the reading in of the stimulus file.

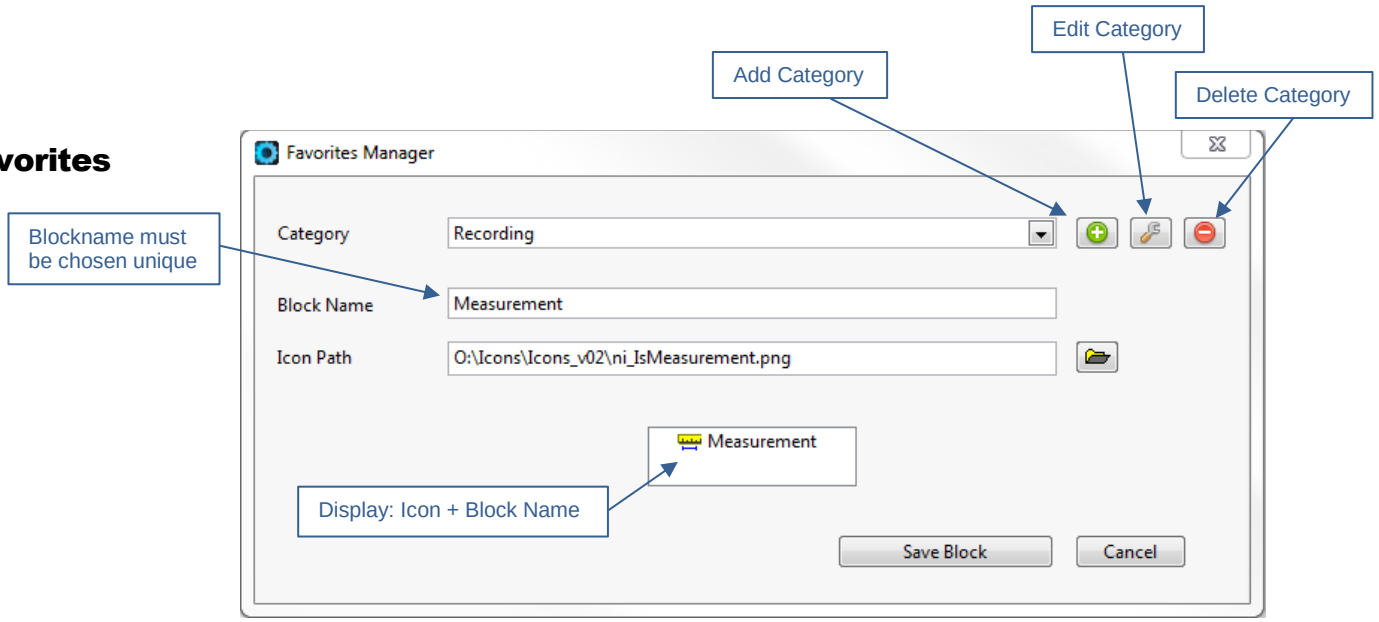
Stimulus File Resume

Resume the reading in of the interrupted stimulus file.

Stimulus File End

Terminate the reading in of the stimulus file.

Favorites



24 - ViPE Favorites Manager

If the configuration of a block is used more than once, it can be stored as a favorite. To do this, right-click on the desired Function Block in the Block Diagram to select the "Add to Favorites" sub-point. The favorite can be added to a category and assigned with an icon. A unique block name must be chosen. If a favorite is created, it is added to the palette.

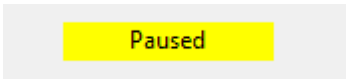
System status

The system status provides information about the current status of the sequence.

Running: The sequence is currently running and cannot be edited during this time. You can follow the current step of execution in the block diagram.

Activate Autoscroll to always keep the current step of exection in focus.

Idle: The sequence is not active and can be edited.



Paused: Execution of the sequence is paused. Press pause again to resume execution of the sequence.

Remote Control

In order to be able to remotely control the **CISWORKS ViPE** via TCP protocol, the **CISWORKS ViPE** must be provided with the appropriate parameters at startup. To do this, the **ViPE** must be started using the Windows console as follows:

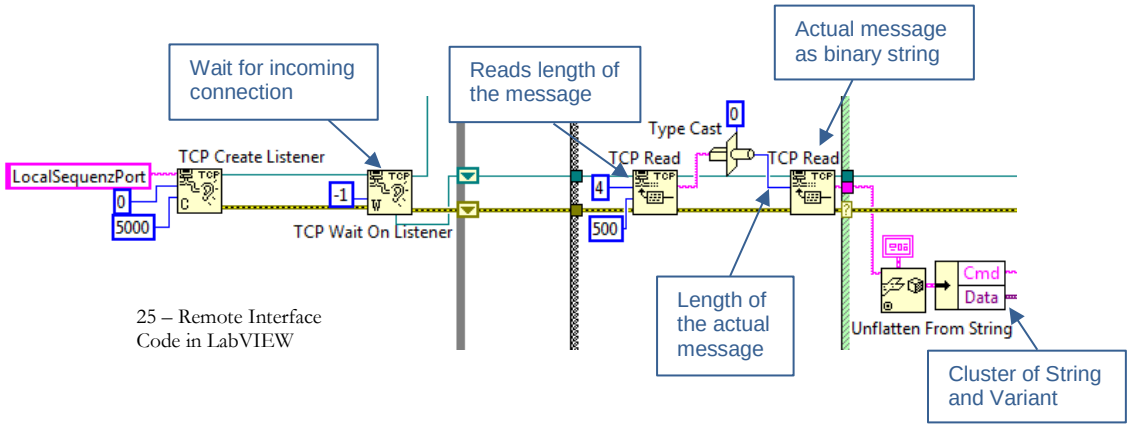
```
<Path to CISWORKS ViPE.exe> Sequenzer/t“LabView-Remote“ /d“<Path
VeriStand System Definition File (.nivssdf)>“ /p“<Path to ViPE
project (.vipeprj)“ /m“Menu“
```

Example

```
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"
```

A TCP listener waits in the background for incoming connections. A TCP connection can be established by means of the IP address of the computer running the **CISWORKS ViPE** and the service name "LocalSequencePort".

Basic Functionality

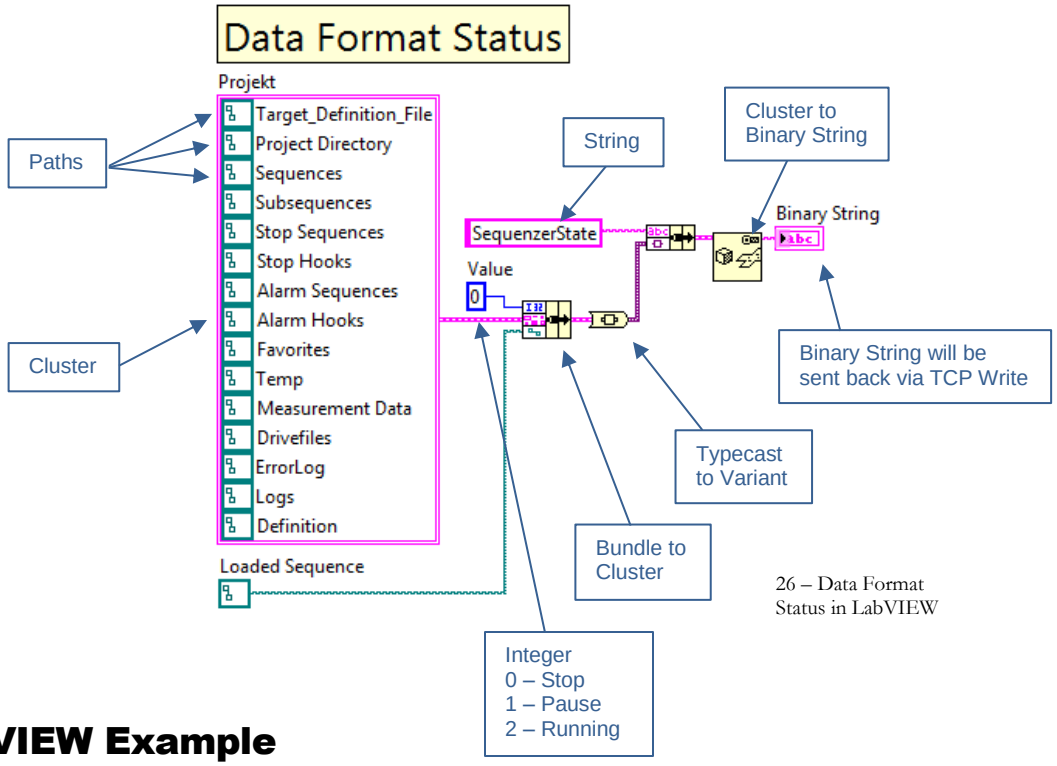


25 – Remote Interface Code in LabVIEW

If a TCP connection is established, two messages must be sent to the remote interface to execute a command. The first message must contain the length of the second message as a binary string. The second message contains the command and any parameters. The remote interface also expects the second message as a binary string. This binary string is converted to a cluster in the remote interface. The cluster consists of a string and a variant data type. The string represents the command and the variant represents the corresponding parameter. Parameters can be integers and paths.

Commands

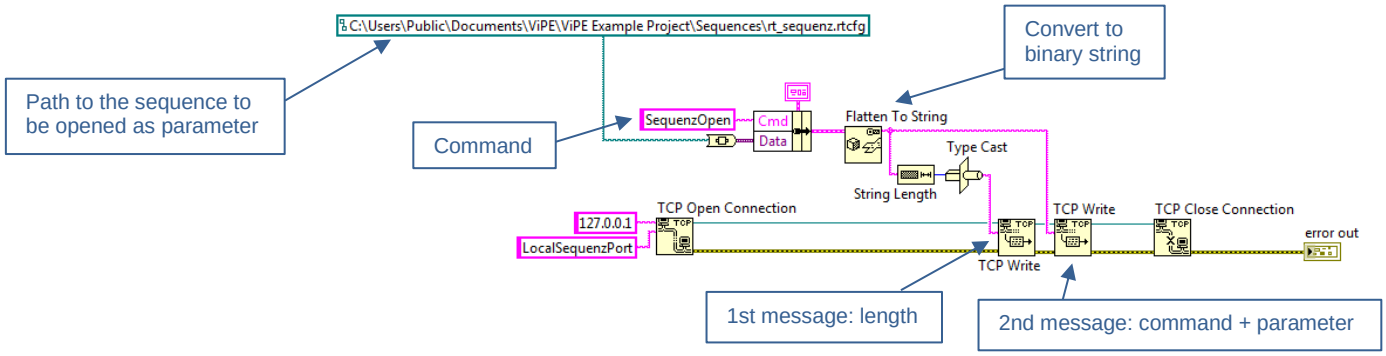
Exit	Terminates **ViPE**, no parameter
SequenzerWindow	Maximizes **ViPE** window if Hidden or vice versa, no parameter
SequenzOpen	Opens a sequence, path of the sequence as parameter
ProjektOpen	Opens a project, path oft he project as parameter
ProjektNeu	Creates a new project, no parameter
SequenzStart	Starts current sequence, no parameter
SequenzStop	Stops current running sequence, no parameter
SequenzPause	Pauses current running sequence, no parameter
Setze_Messpfad	Sets the measuring path, path oft he measurement file as parameter
Status	Writes current status via TCP Write to the TCP connection, does not expect any parameters. A binary string is obtained which can be converted into a cluster consisting of a string and a variant. The variant can be converted to a cluster. This cluster consists of an integer, another cluster consisting of many paths, and a further path. The image below shows the data format of the status in LabVIEW.



26 – Data Format Status in LabVIEW

LabVIEW Example

The following example shows the sending of the remote command: "SequenceOpen" in LabVIEW.



27 – Command „SequenzOpen“ in LabVIEW

In the example, a TCP connection to the remote interface is established using the LabVIEW VI **TCP Open Connection**. The "SequenceOpen" command is processed together with the path converted into a variant data type as a cluster. This cluster is converted to a binary string. The length of this string is sent to the remote interface via TCP Write in the first message. The actual content is sent in the second message via TCP Write too.