zenon
by COPA-DATA

zenon Treiber Handbuch

MATHDR32

v.8.20

© 2020 Ing. Punzenberger COPA-DATA GmbH

Alle Rechte vorbehalten.

Die Weitergabe und Vervielfältigung dieses Dokuments ist - gleich in welcher Art und Weise - nur mit schriftlicher Genehmigung der Firma COPA-DATA gestattet. Technische Daten dienen nur der Produktbeschreibung und sind keine zugesicherten Eigenschaften im Rechtssinn. Änderungen - auch in technischer Hinsicht - vorbehalten.

Inhaltsverzeichnis

1 Willkommen bei der COPA-DATA Hilfe 5

2 MATHDR32 5

3 MATHDR32 - Datenblatt 6

4 Treiber-Historie 8

5 Konfiguration 8

5.1 Anlegen eines Treibers 9

6 Variablen anlegen 12

6.1 Variablen im Editor anlegen 13

6.2 Adressierung 16

6.3 Treiberobjekte und Datentypen 17

6.3.1 Treiberobjekte 18

6.3.2 Zuordnung der Datentypen 18

6.4 Variablen anlegen durch Import 19

6.4.1 XML Import 19

6.4.2 DBF Import/Export 20

6.5 Kommunikationsdetails (Treibervariablen) 26

7 Treiberspezifische Funktionen 32

7.1 Formeln 33

7.1.1 Variablen zuordnen 36

7.1.2 Verknüpfungsarten 38

7.1.3 Bitformeln 41

7.1.4 Floatformel 42

7.2 Ereignisgesteuertes Berechnen der Mathematik-Variablen 65

7.2.1 Aktivierung und Deaktivierung der Berechnung 67

7.2.2 Berechnungsarten mit besonderem Verhalten 68

7.2.3 Fehlermeldungen 69

7.3 Import von Mathematik-Variablen 69

8 Funktion Treiberkommandos 70

9 Fehleranalyse 75

9.1 Analysetool 75

9.2 Treiberüberwachung 76
9.3 Logserver 78
9.4 Checkliste 78

1 Willkommen bei der COPA-DATA Hilfe

ZENON VIDEO-TUTORIALS

Praktische Beispiele für die Projektierung mit zenon finden Sie in unserem YouTube-Kanal (https://www.copadata.com/tutorial_menu). Die Tutorials sind nach Themen gruppiert und geben einen ersten Einblick in die Arbeit mit den unterschiedlichen zenon Modulen. Alle Tutorials stehen in englischer Sprache zur Verfügung.

ALLGEMEINE HILFE

Falls Sie in diesem Hilfekapitel Informationen vermissen oder Wünsche für Ergänzungen haben, wenden Sie sich per E-Mail an documentation@copadata.com.

PROJEKTUNTERSTÜTZUNG

Unterstützung bei Fragen zu konkreten eigenen Projekten erhalten Sie vom Customer Service, den Sie per E-Mail an support@copadata.com erreichen.

LIZENZEN UND MODULE

Sollten Sie feststellen, dass Sie weitere Module oder Lizenzen benötigen, sind unsere Mitarbeiter unter sales@copadata.com gerne für Sie da.

2 MATHDR32

Der 8.20 Mathematik Treiber dient zur Definition von Formeln und deren Berechnung unter Einbeziehung der Daten aus anderen Prozesstreibern.

> **Info**
>
> Der Mathematik Treiber verbraucht keine I/Os.

Wird eine Zählervariable des Mathematik Treibers in einem Archiv mit Speicherung bei Wertänderung (spontan) gespeichert, kann das zur Speicherung von sehr großen Datenmengen führen.

Workaround: Weisen Sie die Mathematik-Variable einer Simulationsvariablen zu und speichern Sie die Simulationsvariable im Archiv.
Achtung: Die Simulationsvariable muss vom Datentyp *INT* sein (nicht *REAL*).

Info

Werte des Mathematik Treibers werden beim normalen Beenden der Runtime gespeichert. Um die Speicherung der Werte sicher zu stellen, wenn Grenzwerte oder Rema-Bedingungen, bei denen der Zähler läuft, inaktiv werden:

Führen Sie die Funktion **AML- und CEL-Ringpuffer speichern** oder die Funktion **Remanentdaten speichern** aus. Zusätzlich können Sie die Werte noch zyklisch (z.B jede Stunde) speichern.

3 MATHDR32 - Datenblatt

Allgemein:	
Treiberdateiname	MATHDR32.exe
Treiberbezeichnung	Treiber für math. Berechnungen
Steuerungs-Typen	--
Steuerungs-Hersteller	zenon system driver; Internal driver

Treiber unterstützt:	
Protokoll	unknown
Adressierung: Adress-basiert	Name based
Adressierung: Namens-basiert	--
Kommunikation spontan	--
Kommunikation pollend	X
Online Browsing	--
Offline Browsing	--

Treiber unterstützt:	
Echtzeitfähig	--
Blockwrite	--
Modemfähig	--
RDA numerisch	--
RDA String	--
Hysterese	--
erweiterte API	--
Unterstützung von Statusbit **WR-SUC**	--
alternative IP-Adresse	--

Voraussetzungen:	
Hardware PC	--
Software PC	--
Hardware Steuerung	--
Software Steuerung	--
Benötigt v-dll	--

Plattformen:	
Betriebssysteme	Windows 10; Windows 7; Windows 8; Windows 8.1; Windows Server 2008 R2; Windows Server 2012; Windows Server 2012 R2; Windows Server 2016

4 Treiber-Historie

Datum	Treiberversion	Änderung
07.07.08	400	Treiberdokumentation wurde neu erstellt

TREIBERVERSIONIERUNG

Mit zenon 7.10 wurde die Versionierung der Treiber verändert. Ab dieser Version gibt es eine versionsübergreifende Build-Nummer. Das ist die Zahl an der 4. Stelle der Dateiversion.
Zum Beispiel: **7.10.0.4228** bedeutet: Der Treiber ist für Version **7.10**, Service Pack **0** und hat die Build-Nummer **4228**.

Erweiterungen oder Fehlerbehebungen werden zukünftig in einem Build eingebaut und sind dann ab der nächsthöheren Build-Nummer verfügbar.

Beispiel

Eine Treibererweiterung wurde in Build **4228** implementiert. Der Treiber, den Sie im Einsatz haben, verfügt über die Build-Nummer **8322**. Da die Build-Nummer Ihres Treibers höher ist als die Build-Nummer der Erweiterung, ist die Erweiterung enthalten. Die Versionsnummer des Treiber (die ersten drei Stellen der Dateiversion) spielen dabei keine Rolle. Die Treiber sind versionsunabsabhängig

5 Konfiguration

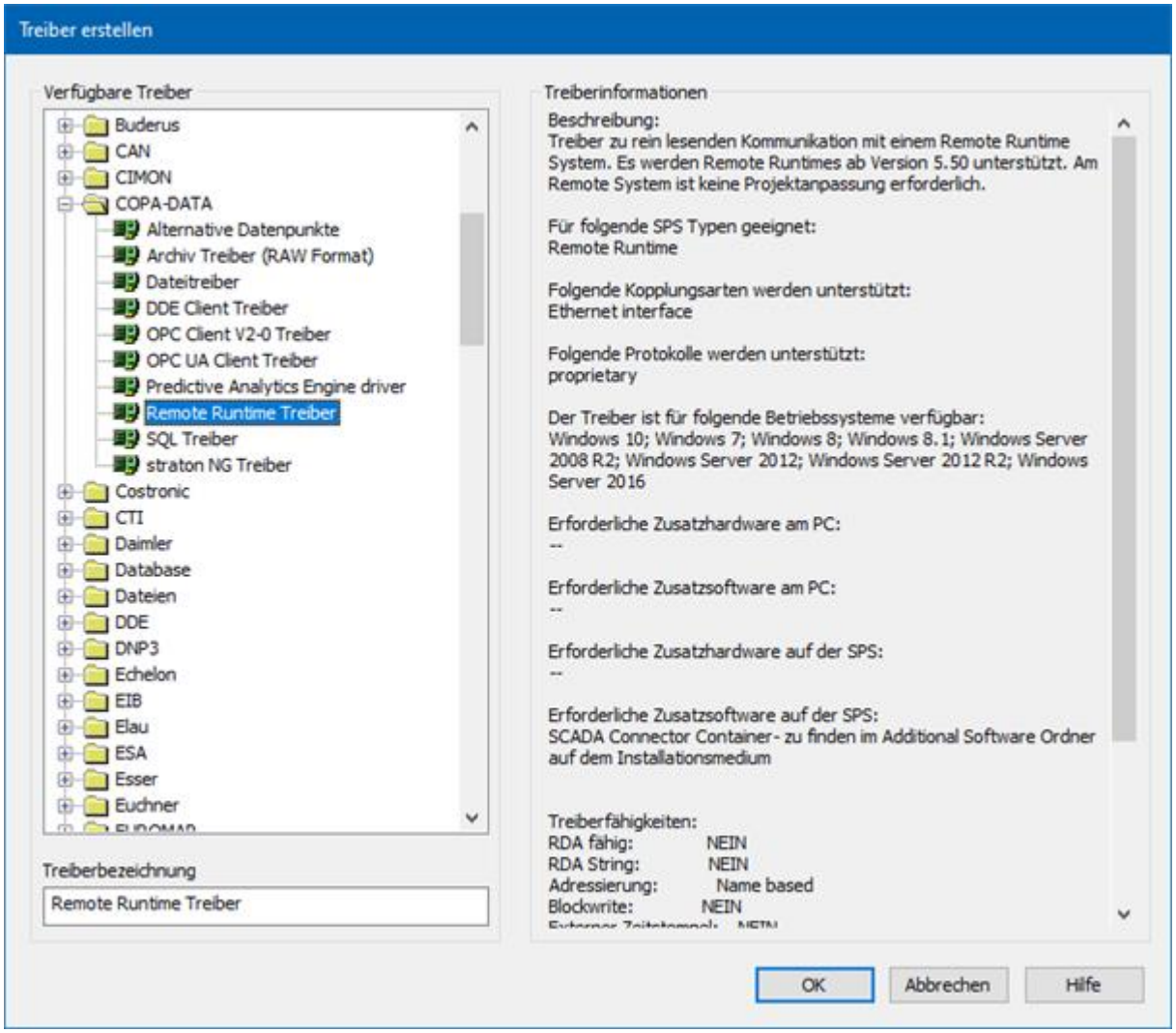
In diesem Kapitel lesen Sie, wie Sie den Treiber im Projekt anlegen und welche Einstellungen beim Treiber möglich sind.

Info

Weitere Einstellungen, die Sie für Variablen in zenon vornehmen können, finden Sie im Kapitel Variablen der Online-Hilfe.

5.1 Anlegen eines Treibers

Im Dialog **Treiber erstellen** wählen Sie aus einer Liste jenen Treiber, den Sie neu anlegen wollen.



Parameter	Beschreibung
Verfügbare Treiber	Liste aller verfügbaren Treiber. Die Darstellung erfolgt in einer Baumstruktur: *[+]* erweitert die Ordnerstruktur und zeigt die darin enthaltenen Treiber. [-] reduziert die Ordnerstruktur Default: *keine Auswahl*
Treiberbezeichnung	Eindeutige **Bezeichnung** des Treibers. Default: *leer* Das Eingabefeld wird nach Auswahl eines Treibers

Parameter	Beschreibung
	aus der Liste der verfügbaren Treiber mit der vordefinierten **Bezeichnung** vorausgefüllt.
Treiberinformationen	Weiterführende Informationen über den gewählten Treiber. Default: *leer* Nach Auswahl eines Treibers werden in diesem Bereich die Informationen zum gewählten Treiber angezeigt.

DIALOG BEENDEN

Option	Beschreibung
OK	Übernimmt alle Einstellungen und öffnet den Treiberkonfigurationsdialog des ausgewählten Treibers.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

> **Info**
>
> Die Inhalte dieses Dialogs sind in der Datei Treiber_[Sprachkürzel].xml gespeichert. Sie finden diese Datei im Ordner *C:\ProgramData\COPA-DATA\zenon[Versionsnummer]*.

TREIBER NEU ANLEGEN

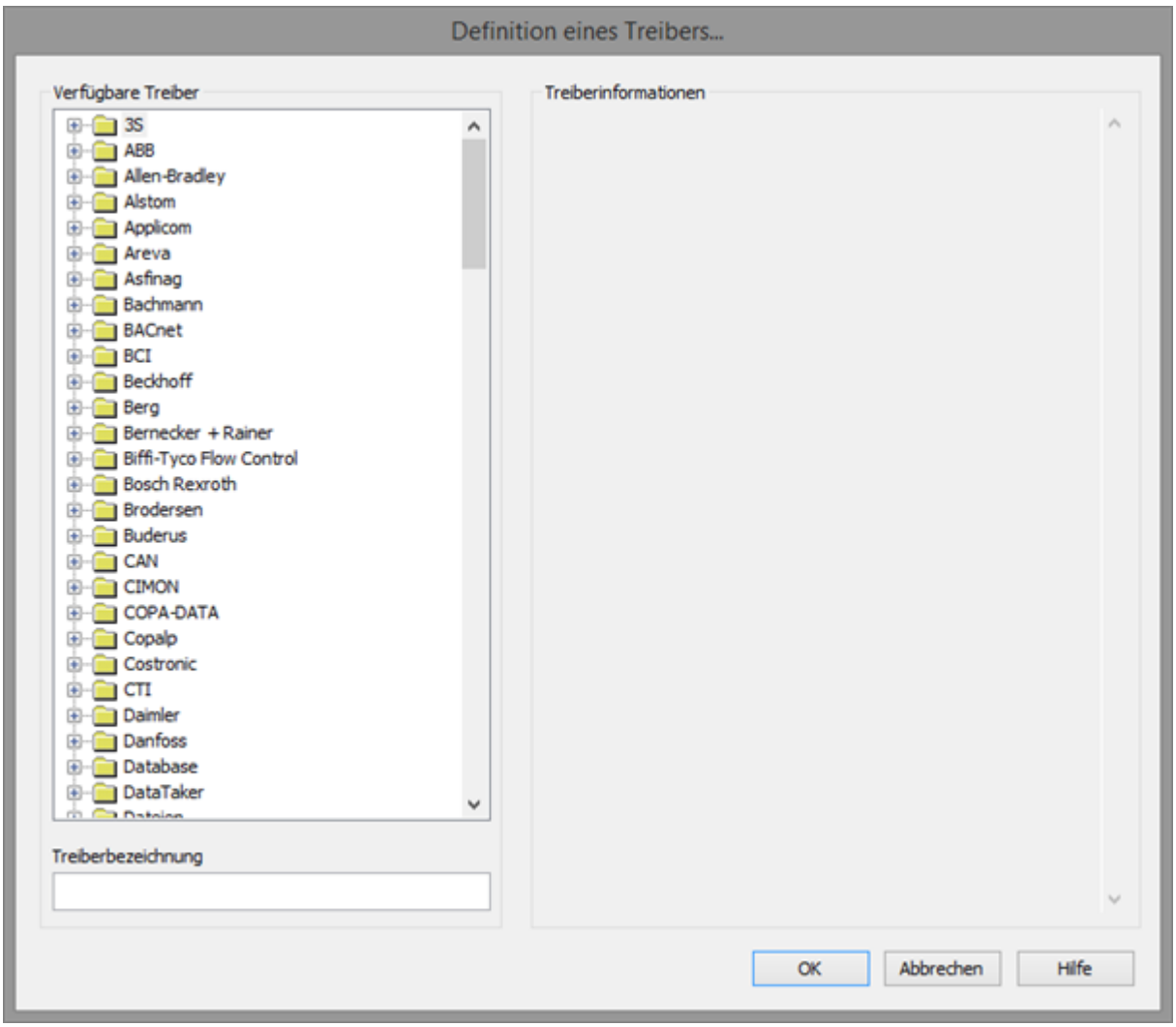
Um einen neuen Treiber anzulegen:

1. Klicken Sie mit der rechten Maustaste im Projektmanager auf **Treiber** und wählen Sie im Kontextmenü **Treiber neu** aus.

   Optional: Wählen Sie die Schaltfläche **Treiber neu** aus der Symbolleiste der Detailansicht der **Variablen**.

   Der Dialog **Treiber erstellen** wird geöffnet.

2. Der Dialog bietet eine Auflistung aller verfügbaren Treiber an.



3. Wählen Sie den gewünschten Treiber und benennen Sie diesen im Eingabefeld **Treiberbezeichnung**.
   Dieses Eingabefeld entspricht der Eigenschaft **Bezeichnung**. Per Default wird der Name des ausgewählten Treibers in diesem Eingabefeld automatisch eingefügt.
   Für die **Treiberbezeichnung** gilt:

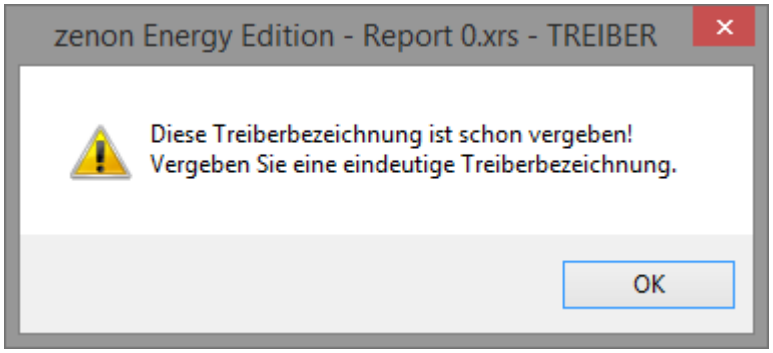
   - Die **Treiberbezeichnung** muss eindeutig sein.
     Wird ein Treiber mehrmals im Projekt verwendet, so muss jeweils eine neue Bezeichnung vergeben werden.
     Dies wird durch Klick auf die Schaltfläche **OK** evaluiert. Ist die Treiber im Projekt bereits vorhanden wird dies mit einem Warndialog angezeigt.
   - Die **Treiberbezeichnung** ist Bestandteil des Dateinamens.
     Daher darf Sie nur Zeichen enthalten, die vom Betriebssystem unterstützt werden. Nicht gültige Zeichen werden durch einen Unterstrich (_) ersetzt.
   - **Achtung:** Die Bezeichnung kann später nicht mehr geändert werden.

4. Bestätigen Sie den Dialog mit Klick auf die Schaltfläche **OK**.
   Der Konfigurationsdialog des ausgewählten Treibers wird geöffnet.

Hinweis: Treibernamen sind nicht sprachumschaltbar. Sie werden später immer in der Sprache angezeigt, in der sie angelegt wurden, unabhängig von der Sprache des Editors. Das gilt auch für Treiberobjekttypen.

DIALOG TREIBERBEZEICHNUNG BEREITS VORHANDEN

Ist ein Treiber bereits im Projekt vorhanden wird dies in einem Dialog angezeigt. Mit Klick auf die Schaltfläche **OK** wird der Warndialog geschlossen. Der Treiber kann korrekt benannt werden.



ZENON PROJEKT

Bei neu angelegten Projekten werden die folgenden Treiber automatisch angelegt:

- **Intern**
- **MathDr32**
- **SysDrv**

> **Info**
>
> In einem zenon Projekt müssen nur die benötigten Treiber vorhanden sein. Treiber können bei Bedarf zu einem späteren Zeitpunkt hinzugefügt werden.

6 Variablen anlegen

So werden Variablen im zenon Editor angelegt:

6.1 Variablen im Editor anlegen

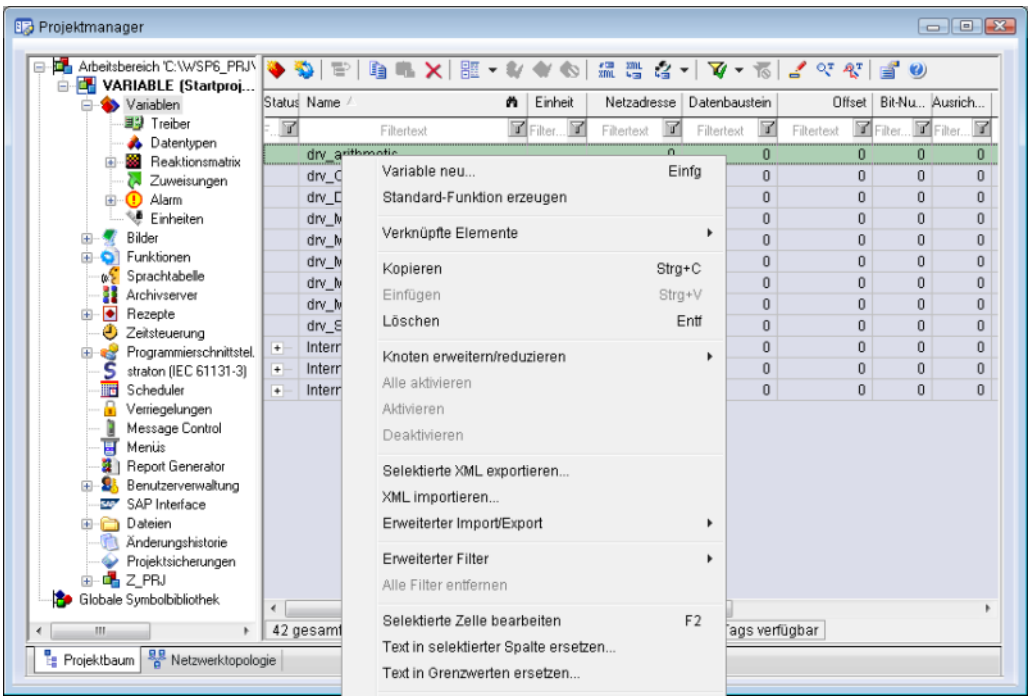
Variablen können angelegt werden:

- als einfache Variable
- in Arrays
- als Struktur-Variablen

DIALOG VARIABLE

Um eine neue Variable zu erstellen, gleich welchen Typs:

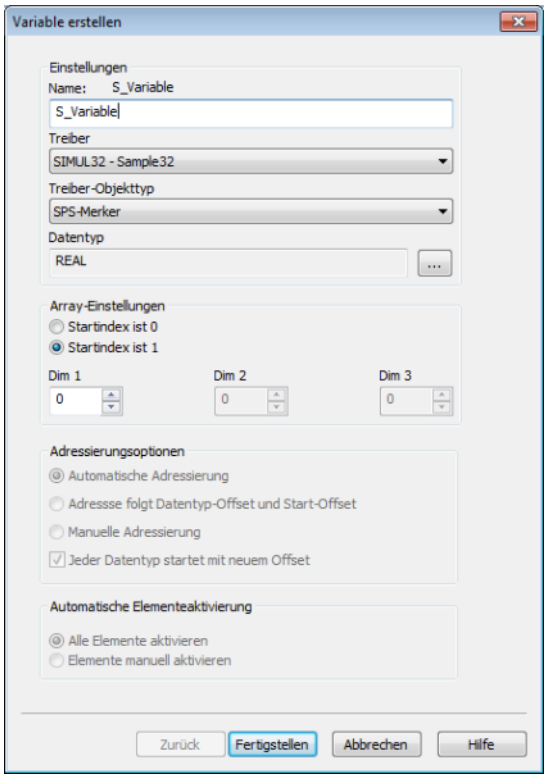
1. Wählen Sie im Knoten **Variablen** im Kontextmenü den Befehl **Variable neu**.



   Der Dialog zur Konfiguration der Variable wird geöffnet.

2. Konfigurieren Sie die Variable.
3. Welche Einstellungen möglich sind, hängt ab vom Typ der Variablen.

DIALOG VARIABLE ERSTELLEN



Eigenschaft	Beschreibung
Name	Eindeutiger Name der Variablen. Ist eine Variable mit gleichem Namen im Projekt bereits vorhanden, kann keine weitere Variable mit diesem Namen angelegt werden. Maximale Länge: 128 Zeichen **Achtung:** Die Zeichen **#** und **@** sind für Variablennamen nicht erlaubt. Bei Verwendung nicht zugelassener Zeichen kann die Variablenerstellung nicht abgeschlossen werden, die Schaltfläche **Fertigstellen** bleibt inaktiv. **Hinweis:** Manche Treiber erlauben die Adressierung auch über die Eigenschaft **Symbolische Adresse**.
Treiber	Wählen Sie aus der Dropdownliste den gewünschten Treiber. **Hinweis:** Sollte im Projekt noch kein Treiber angelegt sein, wird automatisch der Treiber für interne Variable (**Intern.exe**) geladen.
Treiberobjekttyp	Wählen Sie aus der Dropdownliste den passenden Treiberobjekttyp aus.
Datentyp	Wählen Sie den gewünschten Datentyp. Klick auf die Schaltfläche

Eigenschaft	Beschreibung
	... öffnet den Auswahl-Dialog.
Array-Einstellungen	Erweiterte Einstellungen für Array-Variablen. Details dazu lesen Sie im Abschnitt Arrays.
Adressierungsoptionen	Erweiterte Einstellungen für Arrays und Struktur-Variablen. Details dazu lesen Sie im jeweiligen Abschnitt.
Automatische Elementeaktivierung	Erweiterte Einstellungen für Arrays und Struktur-Variablen. Details dazu lesen Sie im jeweiligen Abschnitt.

SYMBOLISCHE ADRESSE

Die Eigenschaft **Symbolische Adresse** kann für die Adressierung alternativ zu **Name** oder **Kennung** der Variablen verwendet werden. Die Auswahl erfolgt im Treiberdialog, die Konfiguration in der Variableneigenschaft. Beim Import von Variablen unterstützter Treiber wird die Eigenschaft automatisch eingetragen.

Maximale Länge: *1024* Zeichen.

Folgende Treiber unterstützen die **Symbolische Adresse**:

- **3S_V3**
- **AzureDrv**
- **BACnetNG**
- **IEC850**
- **KabaDPServer**
- **OPCUA32**
- **Phoenix32**
- **POZYTON**
- **RemoteRT**
- **S7TIA**
- **SEL**
- **SnmpNg32**
- **PA_Drv**
- **EUROMAP63**

ABLEITUNG VOM DATENTYP

Messbereich, **Signalbereich** und **Sollwert Setzen** werden immer:

- vom Datentyp abgeleitet
- beim Ändern des Datentyps automatisch angepasst

Hinweis Signalbereich: Bei einem Wechsel auf einen Datentyp, der den eingestellten **Signalbereich** nicht unterstützt, wird der **Signalbereich** automatisch angepasst. Zum Beispiel wird bei einem Wechsel von **INT** auf **SINT** der **Signalbereich** auf *127* geändert. Die Anpassung erfolgt auch dann, wenn der **Signalbereich** nicht vom Datentyp abgeleitet wurde. In diesem Fall muss der **Messbereich** manuell angepasst werden.

6.2 Adressierung

Gruppe/Eigenschaft	Beschreibung
Allgemein	Gruppe mit allgemeinen Eigenschaften.
Name	Frei vergebbarer Name. **Achtung:** Je zenon Projekt muss der Name eindeutig sein.
Kennung	Frei vergebbare Kennung. Z. B. für **Betriebsmittelkennung**, Kommentar usw.
Adressierung	
Netzadresse	Wird für diesen Treiber nicht verwendet.
Datenbaustein	Wird für diesen Treiber nicht verwendet.
Offset	Wird für diesen Treiber nicht verwendet.
Ausrichtung	Wird für diesen Treiber nicht verwendet.
Bitnummer	Wird für diesen Treiber nicht verwendet.
Stringlänge	Wird für diesen Treiber nicht verwendet.
Treiber Anbindung/Datentyp	Datentyp der Variablen. Wird beim Erstellen der Variablen ausgewählt und kann hier geändert werden. **ACHTUNG:** Wenn der Datentyp nachträglich geändert wird, müssen alle anderen Eigenschaften der Variablen überprüft bzw. angepasst werden.
Treiber Anbindung/Treiber objekttyp	Objekttyp der Variablen. Wird abhängig vom verwendeten Treiber beim Erstellen der Variablen ausgewählt und kann hier geändert werden.
Treiber Anbindung/Priorität	Wird für diesen Treiber nicht verwendet. Der Treiber unterstützt keine zyklisch pollende Kommunikation in Prioritätsklassen.

MATHEMATIK-VARIABLE

Der Mathematiktreiber ist ein Treiber zur Definition von Formeln (Verknüpfungsart Arithmetik, Trigonometrie etc.) und deren Berechnung unter Einbeziehung der Daten aus anderen Prozesstreibern. Von zenon aus betrachtet, ist der Mathematiktreiber ein "normaler" Prozesstreiber. Die Variablendefinition erfolgt im Editor (Mathematikvariable), die berechneten Werte der entsprechenden Mathematikvariable werden im Onlinebetrieb zur Verfügung gestellt. Für die Definition und Verwendung einer Mathematikvariablen muss der Mathematiktreiber geladen werden.

Zu jeder mathematischen Variablen des Mathematiktreibers muss eine Formel definiert werden. Jede Formel hat wenigstens eine Quellvariable. Als Quellvariable kann jede vorher definierte Variable (auch vorher definierte Mathematikvariable) dienen. Die Verknüpfung erfolgt mit den Signalauflösungen (technischen Werten) der Variablen.

Achtung

Sie können maximal 4096 Mathematik-Variablen mit Formeln projektieren. Wenn Sie mehr Formeln verwenden, werden nur die ersten 4096 in der Runtime ausgeführt. Alle weiteren werden unterdrückt.

Sie können maximal 4096 numerische Konstante über alle Mathematikformeln aufsummieren. Alle weiteren Konstanten werden in der Runtime als 0 angenommen.

Falls Sie mehr Variablen/Konstanten benötigen, empfehlen wir die Verwendung von zenon Logic.

Tipp: zenon Logic ist bereits in zenon integriert. Falls zenon Logic in Ihrem Lizenzumfang nicht enthalten ist, wenden Sie sich an Ihren Vertriebspartner.

Info

Es können bis zu 99 Variablen in einer Formel verknüpft werden. X01 bis X99. Die Länge der Formel darf dabei 4096 Zeichen nicht überschreiten.

6.3 Treiberobjekte und Datentypen

Treiberobjekte sind in der Steuerung verfügbare Bereiche wie z. B. Merker, Datenbausteine usw. Hier lesen Sie, welche Treiberobjekte vom Treiber zur Verfügung gestellt werden und welche IEC-Datentypen dem jeweiligen Treiberobjekt zugeordnet werden können.

6.3.1 Treiberobjekte

Folgende Objekttypen stehen in diesem Treiber zur Verfügung:

TREIBER OBJEKTTYPEN UND UNTERSTÜTZTE IEC DATENTYPEN FÜR PROZESSVARIABLEN IN ZENON

Treiberobjekttyp	Kanaltyp	Lesen	Schreiben	Unterstützte Datentypen	Kommentar
Formel	17	X	--	*BOOL, LREAL*	

KANALTYP

Der Begriff "**Kanaltyp**" ist die interne numerische Bezeichnung des Treiberobjekttyps. Diese wird auch für den erweiterten DBF Import/Export der Variablen verwendet.

"**Kanaltyp**" wird für den erweiterten CSV Import/Export der Variablen in der Spalte "**HWObjectType**" verwendet.

6.3.2 Zuordnung der Datentypen

Alle Variablen in zenon werden von IEC-Datentypen abgeleitet. In folgender Tabelle werden zur besseren Übersicht die IEC-Datentypen den Datentypen der Steuerung gegenübergestellt.

ZUORDNUNG DER DATENTYPEN AUS DER STEUERUNG AUF DIE ZENON DATENTYPEN

Steuerung	zenon	Datenart
BOOL	BOOL	8
LREAL	LREAL	5

DATENART

Der Begriff **Datenart** ist die interne numerische Bezeichnung des Datentyps. Diese wird auch für den erweiterten DBF Import/Export der Variablen verwendet.

6.4 Variablen anlegen durch Import

Variablen können auch mittels Variablenimport angelegt werden. Für jeden Treiber stehen XML- und DBF-Import zur Verfügung.

> **Info**
>
> Details zu Import und Export von Variablen finden Sie im Handbuch Import-Export im Abschnitt Variablen.

6.4.1 XML Import

Beim XML- Import von Variablen oder Datentypen werden diese erst einem Treiber zugeordnet und dann analysiert. Vor dem Import entscheidet der Benutzer, ob und wie das jeweilige Element (Variable oder Datentyp) importiert werden soll:

- *Importieren*:
  Das Element wird neu importiert.
- *Überschreiben*:
  Das Element wird importiert und überschreibt ein bereits vorhandenes Element.
- *Nicht importieren*:
  Das Element wird nicht importiert.

Hinweis: Beim Import werden die Aktionen und deren Dauer in einem Fortschrittsbalken angezeigt. In der folgenden Dokumentation wird der Import von Variablen beschrieben. Datentypen werden analog dazu importiert.

VORAUSSETZUNGEN

Beim Import gelten folgende Bedingungen:

- **Abwärtskompatibilität**

  Beim XML Import/Export ist keine Abwärtskompatibilität gegeben. Daten aus älteren zenon Versionen können übernommen werden. Die Übergabe von Daten aus neueren Versionen an ältere wird nicht unterstützt.

- **Konsistenz**

  Die zu importierende XML-Datei muss konsistent sein. Beim Import der Datei erfolgt keine Plausibilitätsprüfung. Weisen die importierten Daten Fehler auf, kann es zu unerwünschten Effekten im Projekt kommen.

Dies muss vor allem auch beachtet werden, wenn in einer XML-Datei nicht alle Eigenschaften vorhanden sind und diese dann durch Default-Werte ersetzt werden. Z. B.: Eine binäre Variable hat einen Grenzwert von *300*.

- **Struktur-Datentypen**

  Struktur-Datentypen müssen über die gleiche Anzahl von Strukturelementen verfügen. Beispiel: Ein Strukturdatentyp im Projekt hat 3 Strukturelemente. Ein gleichnamiger Datentyp in der XML-Datei hat 4 Strukturelemente. Dann wird keine der auf diesem Datentyp basierenden Variablen der Datei in das Projekt importiert.

> **Tipp**
>
> Weitere Informationen zum XML-Import finden Sie im Handbuch **Import - Export**, im Kapitel **XML-Import**.

6.4.2 DBF Import/Export

Daten können nach dBase exportiert und aus dBase importiert werden.

> **Info**
>
> Import und Export über CSV oder dBase unterstützt keine treiberspezifischen Variableneinstellungen wie z. B. Formeln. Nutzen Sie dafür den Export/Import über XML.

IMPORT DBF-DATEI

Um den Import zu starten:

1. Führen Sie einen Rechtsklick auf die Variablenliste aus.
2. Wählen Sie in der Dropdownliste von **Erweiterter Export/Import** ... den Befehl **dBase importieren**.
3. Folgen Sie den Anweisungen des Importassistenten.

Das Format der Datei ist im Kapitel Dateiaufbau beschrieben.

> **Info**
>
> Beachten Sie:
>
> - Treiberobjekttyp und Datentyp müssen in der DBF-Datei an den Zieltreiber angepasst werden, damit Variablen importiert werden.
> - dBase unterstützt beim Import keine Strukturen oder Arrays (komplexe Variablen).

EXPORT DBF-DATEI

Um den Export zu starten:

1. Führen Sie einen Rechtsklick auf die Variablenliste aus.
2. Wählen Sie im Dropdownliste von **Erweiterter Export/Import** ... den Befehl **dBase exportieren...** .
3. Folgen Sie den Anweisungen des Exportassistenten.

> **Achtung**
>
> DBF-Dateien:
>
> - müssen in der Benennung dem 8.3 DOS Format für Dateinamen entsprechen (8 alphanumerische Zeichen für Name, 3 Zeichen Erweiterung, keine Leerzeichen)
> - dürfen im Pfadnamen keinen Punkt (.) enthalten.
> Z. B. ist der Pfad *C:\users\Max.Mustermann\test.dbf* ungültig.
> Gültig wäre: *C:\users\MaxMustermann\test.dbf*
> - müssen nahe am Stammverzeichnis (Root) abgelegt werden, um die eventuelle Beschränkungen für Dateinamenlänge inklusive Pfad zu erfüllen: maximal 255 Zeichen

Das Format der Datei ist im Kapitel Dateiaufbau beschrieben.

> **Info**
>
> dBase unterstützt beim Export keine Strukturen oder Arrays (komplexe Variablen).

DATEIAUFBAU DER DBASE EXPORTDATEI

Für den Variablenimport und -export muss die dBaseIV-Datei folgende Struktur und Inhalte besitzen.

Achtung

dBase unterstützt keine Strukturen oder Arrays (komplexe Variablen).

DBF-Dateien müssen:

- in der Benennung dem 8.3 DOS Format für Dateinamen entsprechen (8 alphanumerische Zeichen für Name, 3 Zeichen Erweiterung, keine Leerzeichen)
- nahe am Stammverzeichnis (Root) abgelegt werden

STRUKTUR

Bezeichnung	Typ	Feldgröße	Bemerkung
KANALNAME	Char	128	Variablenname. Länge kann über den Eintrag **MAX_LAENGE** in der **project.ini** eingeschränkt werden.
KANAL_R	C	128	Ursprünglicher Name einer Variablen, der durch den Eintrag unter VARIABLENNAME ersetzt werden soll (Feld/Spalte muss manuell angelegt werden). Länge kann über den Eintrag **MAX_LAENGE** in der **project.ini** eingeschränkt werden.
KANAL_D	Log	1	Variable wird bei Eintrag *1* gelöscht (Feld/Spalte muss manuell angelegt werden).
TAGNR	C	128	Kennung. Länge kann über den Eintrag **MAX_LAENGE** in der **project.ini** eingeschränkt werden.
EINHEIT	C	11	Technische Maßeinheit
DATENART	C	3	Datentyp (z. B. Bit, Byte, Wort, ...) entspricht dem Datentyp.
KANALTYP	C	3	Speicherbereich in der SPS (z. B. Merkerbereich, Datenbereich, ...) entspricht Treiberobjekttyp.
HWKANAL	Num	3	Netzadresse
BAUSTEIN	N	3	Datenbaustein-Adresse (nur bei Variablen aus den Datenbereich der SPS)

Bezeichnung	Typ	Feldgröße	Bemerkung
ADRESSE	N	5	Offset
BITADR	N	2	Für Bit-Variablen: Bitadresse Für Byte-Variablen: 0=niederwertig, 8=höherwertig Für String-Variablen: Stringlänge (max. 63 Zeichen)
ARRAYSIZE	N	16	Anzahl der Variablen im Array für Index-Variablen ACHTUNG: Nur die erste Variable steht voll zur Verfügung. Alle folgenden sind nur über VBA oder den Rezeptgruppen Manager zugänglich
LES_SCHR	L	1	Lese-Schreib-Berechtigung 0: Sollwert setzen ist nicht erlaubt 1: Sollwert setzen ist erlaubt
MIT_ZEIT	L	1	Zeitstempelung in zenon (nur wenn vom Treiber unterstützt)
OBJEKT	N	2	Treiberspezifische ID-Nummer des Primitivobjekts setzt sich zusammen aus TREIBER-OBJEKTTYP und DATENTYP
SIGMIN	Float	16	Rohwertsignal minimal (Signalauflösung)
SIGMAX	F	16	Rohwertsignal maximal (Signalauflösung)
ANZMIN	F	16	technischer Wert minimal (Messbereich)
ANZMAX	F	16	technischer Wert maximal (Messbereich)
ANZKOMMA	N	1	Anzahl der Nachkommastellen für die Darstellung der Werte (Messbereich)
UPDATERATE	F	19	Updaterate für Mathematikvariablen (in sec, eine Dezimalstelle möglich) bei allen anderen Variablen nicht verwendet
MEMTIEFE	N	7	Nur aus Kompatibilitätsgründen vorhanden
HDRATE	F	19	HD-Updaterate für hist. Werte (in sec, eine Dezimalstelle möglich)
HDTIEFE	N	7	HD-Eintragtiefe für hist. Werte (Anzahl)
NACHSORT	L	1	HD-Werte als nachsortierte Werte
DRRATE	F	19	Aktualisierung an die Ausgabe (für zenon DDE-Server,

Bezeichnung	Typ	Feldgröße	Bemerkung
			in sec, eine Kommastelle möglich)
HYST_PLUS	F	16	Positive Hysterese; ausgehend vom Messbereich
HYST_MINUS	F	16	Negative Hyterese; ausgehend vom Messbereich
PRIOR	N	16	Priorität der Variable
REAMATRIZE	C	32	Name der zugeordnete Reaktionsmatrix
ERSATZWERT	F	16	Ersatzwert; ausgehend vom Messbereich
SOLLMIN	F	16	Sollwertgrenze Minimum; ausgehend vom Messbereich
SOLLMAX	F	16	Sollwertgrenze Maximum; ausgehend vom Messbereich
VOMSTANDBY	L	1	Variable vom Standby Server anfordern; der Wert der Variable wird im redundanten Netzwerkbetrieb nicht vom Server sondern vom Standby Server angefordert
RESOURCE	C	128	Betriebsmittelkennung. Freier String für Export und Anzeige in Listen. Länge kann über den Eintrag MAX_LAENGE in der **project.ini** eingeschränkt werden.
ADJWVBA	L	1	Nichtlineare Wertanpassung: *0*: Nichtlineare Wertanpassung wird verwendet *1*: Nichtlineare Wertanpassung wird nicht verwendet
ADJZENON	C	128	Verknüpftes VBA-Makro zum Lesen der Variablenwerte für die nichtlineare Wertanpassung.
ADJWVBA	C	128	Verknüpftes VBA-Makro zum Schreiben der Variablenwerte für die nichtlineare Wertanpassung.
ZWREMA	N	16	Verknüpfte Zählwert-Rema.
MAXGRAD	N	16	Maximaler Gradient für die Zählwert-Rema.

Achtung

Beim Import müssen Treiberobjekttyp und Datentyp in der DBF-Datei an den Zieltreiber angepasst werden, damit Variablen importiert werden.

GRENZWERTDEFINITION

Grenzwertdefinition für Grenzwert *1* bis *4*, oder Zustand *1* bis *4*:

Bezeichnung	Typ	Feldgröße	Bemerkung
AKTIV1	L	1	Grenzwert aktiv (pro Grenzwert vorhanden)
GRENZWERT1	F	20	technischer Wert oder ID-Nummer der verknüpften Variable für einen dynamischen Grenzwert (siehe VARIABLEx) (wenn unter VARIABLEx *1* steht und hier *-1*, wird die bestehende Variablenzuordnung nicht überschrieben)
SCHWWERT1	F	16	Schwellwert für den Grenzwert
HYSTERESE1	F	14	wird nicht verwendet
BLINKEN1	L	1	Blinkattribut setzen
BTB1	L	1	Protokollierung in CEL
ALARM1	L	1	Alarm
DRUCKEN1	L	1	Druckerausgabe (bei CEL oder Alarm)
QUITTIER1	L	1	quittierpflichtig
LOESCHE1	L	1	löschpflichtig
VARIABLE1	L	1	dyn. Grenzwertverknüpfung der Grenzwert wird nicht durch einen absoluten Wert (siehe Feld GRENZWERTx) festgelegt.
FUNC1	L	1	Funktionsverknüpfung
ASK_FUNC1	L	1	Ausführung über die Alarmmeldeliste
FUNC_NR1	N	10	ID-Nummer der verknüpften Funktion (steht hier -1, so wird die bestehende Funktion beim Import nicht überschrieben)
A_GRUPPE1	N	10	Alarm/Ereignis-Gruppe
A_KLASSE1	N	10	Alarm/Ereignis-Klasse
MIN_MAX1	C	3	Minimum, Maximum
FARBE1	N	10	Farbe als Windowskodierung

Bezeichnung	Typ	Feldgröße	Bemerkung
GRENZTXT1	C	66	Grenzwerttext
A_DELAY1	N	10	Zeitverzögerung
INVISIBLE1	L	1	Unsichtbar

Bezeichnungen in der Spalte Bemerkung beziehen sich auf die in den Dialogboxen zur Definition von Variablen verwendeten Begriffe. Bei Unklarheiten, siehe Kapitel Variablendefinition.

6.5 Kommunikationsdetails (Treibervariablen)

Das Treiberkit implementiert eine Reihe von Treibervariablen, welche in dem Treiberobjekttyp *Kommunikationsdetails* zusammengefasst sind. Diese sind unterteilt in:

- Information
- Konfiguration
- Statistik und
- Fehlermeldungen

Die Definitionen der im Treiberkit implementierten Variablen sind in der Importdatei **DRVVAR.DBF** verfügbar und können von dort importiert werden.
Pfad zur Datei: *%ProgramData%\COPA-DATA\zenon<Versionsnummer>\PredefinedVariables*

Hinweis: Variablennamen müssen in zenon einzigartig sein. Soll nach einem Import der Variablen vom Treiberobjekttyp *Kommunikationsdetails* aus **DRVVAR.DBF** ein erneuter Import durchgeführt werden, müssen die zuvor importierten Variablen umbenannt werden.

Info

Nicht jeder Treiber unterstützt alle Treibervariablen des Treiberobjekttyps *Kommunikationsdetails*.

Zum Beispiel:

- Variablen für Modem-Informationen werden nur von modemfähigen Treibern unterstützt.
- Treibervariablen für den Polling-Zyklus stehen nur für rein pollende Treiber zur Verfügung.
- Verbindungsbezogene Informationen wie **ErrorMSG** werden nur von Treibern unterstützt, die zu einem Zeitpunkt nur eine Verbindung bearbeiten.

INFORMATION

Name aus Import	Typ	Offset	Erklärung
MainVersion	*UINT*	0	Haupt-Versionsnummer des Treibers.
SubVersion	*UINT*	1	Sub-Versionsnummer des Treibers.
BuildVersion	*UINT*	29	Build-Versionsnummer des Treibers.
RTMajor	*UINT*	49	zenon Hauptversionsnummer
RTMinor	*UINT*	50	zenon Sub-Versionsnummer
RTSp	*UINT*	51	zenon Service Pack-Nummer
RTBuild	*UINT*	52	zenon Buildnummer
LineStateIdle	*BOOL*	24.0	TRUE, wenn die Modemleitung belegt ist.
LineStateOffering	*BOOL*	24.1	TRUE, wenn ein Anruf rein kommt.
LineStateAccepted	*BOOL*	24.2	Der Anruf wird angenommen.
LineStateDialtone	*BOOL*	24.3	Rufton wurde erkannt.
LineStateDialing	*BOOL*	24.4	Wahl aktiv.
LineStateRingBack	*BOOL*	24.5	Während Verbindungsaufbau.
LineStateBusy	*BOOL*	24.6	Zielstation besetzt.
LineStateSpecialInfo	*BOOL*	24.7	Spezielle Statusinformation empfangen.
LineStateConnected	*BOOL*	24.8	Verbindung hergestellt.
LineStateProceeding	*BOOL*	24.9	Wahl ausgeführt.
LineStateOnHold	*BOOL*	24.10	Verbindung in Halten.
LineStateConferenced	*BOOL*	24.11	Verbindung im Konferenzmodus.
LineStateOnHoldPendConf	*BOOL*	24.12	Verbindung in Halten für Konferenz.
LineStateOnHoldPendTransfer	*BOOL*	24.13	Verbindung in Halten für Transfer.
LineStateDisconnected	*BOOL*	24.14	Verbindung beendet.
LineStateUnknow	*BOOL*	24.15	Verbindungszustand nicht bekannt.
ModemStatus	*UDINT*	24	Aktueller Modemstatus.

Name aus Import	Typ	Offset	Erklärung
TreiberStop	*BOOL*	28	Treiber gestoppt Bei *Treiberstop*, hat die Variable den Wert *TRUE* und ein **OFF**-Bit. Nach dem Treiberstart, hat die Variable den Wert *FALSE* und kein **OFF**-Bit.
SimulRTState	*UDINT*	60	Informiert über Status der Runtime bei Treibersimulation.
ConnectionStates	*STRING*	61	Interner Verbindungsstatus des Treibers zur SPS. Verbindungszustände: ▸ *0:* Verbindung OK ▸ *1:* Verbindung gestört ▸ *2:* Verbindung simuliert Formatierung: **<Netzadresse>:<Verbindungszustand>;...;...;** Eine Verbindung ist erst nach dem ersten Anmelden einer Variablen bekannt. Damit eine Verbindung im String enthalten ist, muss einmal eine Variable dieser Verbindung angemeldet worden sein. Der Zustand einer Verbindung wird nur aktualisiert, wenn eine Variable der Verbindung angemeldet ist. Ansonsten wird nicht mit der entsprechenden Steuerung kommuniziert.

KONFIGURATION

Name aus Import	Typ	Offset	Erklärung
ReconnectInRead	*BOOL*	27	Wenn TRUE, dann wird beim Lesen automatisch ein Neuaufbau der Verbindung durchgeführt.
ApplyCom	*BOOL*	36	Änderungen an den Einstellungen der seriellen Schnittstelle zuweisen. Das

Name aus Import	Typ	Offset	Erklärung
			Schreiben auf diese Variable hat unmittelbar den Aufruf der Methode SrvDrvVarApplyCom zur Folge (aktuell ohne weitere Funktion).
ApplyModem	*BOOL*	37	Änderungen an den Modemeinstellungen zuweisen. Das Schreiben auf diese Variable hat unmittelbar den Aufruf der Methode SrvDrvVarApplyModem zur Folge. Diese schließt die aktuelle Verbindung und öffnet eine neue entsprechend den Einstellungen **PhoneNumberSet** und **ModemHwAdrSet**.
PhoneNumberSet	*STRING*	38	Telefonnummer, welche verwendet werden soll.
ModemHwAdrSet	*DINT*	39	Hardwareadresse, welche zu der Telefonnummer gehört.
GlobalUpdate	*UDINT*	3	Updatezeit in Millisekunden (ms).
BGlobalUpdaten	*BOOL*	4	TRUE, wenn die Updatezeit global ist.
TreiberSimul	*BOOL*	5	TRUE, wenn der Treiber in Simulation ist.
TreiberProzab	*BOOL*	6	TRUE, wenn das Prozessabbild gehalten werden soll.
ModemActive	*BOOL*	7	TRUE, wenn das Modem bei diesem Treiber aktiv ist.
Device	*STRING*	8	Name der seriellen Schnittstelle oder Name des Modem.
ComPort	*UINT*	9	Nummer der seriellen Schnittstelle.
Baudrate	*UDINT*	10	Baudrate der seriellen Schnittstelle.
Parity	*SINT*	11	Parität der seriellen Schnittstelle.
ByteSize	*USINT*	14	Bitanzahl pro Zeichen der seriellen Schnittstelle. Wert = *0*, wenn der Treiber keine serielle Kommunikation herstellen kann.
StopBit	*USINT*	13	Anzahl der Stoppbits der seriellen Schnittstelle.

Name aus Import	Typ	Offset	Erklärung
Autoconnect	*BOOL*	16	TRUE, wenn die Modemverbindung automatisch beim Lesen/Schreiben aufgebaut werden soll.
PhoneNumber	*STRING*	17	Aktuelle Telefonnummer.
ModemHwAdr	*DINT*	21	Hardwareadresse zur aktuellen Telefonnummer.
RxIdleTime	*UINT*	18	Wenn länger als diese Zeit in Sekunden (s) erfolgreich kein Datenverkehr stattfindet, wird die Modemverbindung beendet.
WriteTimeout	*UDINT*	19	Maximale Schreibdauer bei einer Modemverbindung in Millisekunden (ms).
RingCountSet	*UDINT*	20	So oft läutet ein hereinkommender Anruf, bevor dieser angenommen wird.
ReCallIdleTime	*UINT*	53	Wartezeit zwischen Anrufen in Sekunden (s).
ConnectTimeout	*UINT*	54	Zeit in Sekunden (s) für Verbindungsaufbau.

STATISTIK

Name aus Import	Typ	Offset	Erklärung
MaxWriteTime	*UDINT*	31	Längste Zeit in Millisekunden (ms), die zum Schreiben benötigt wird.
MinWriteTime	*UDINT*	32	Kürzeste Zeit in Millisekunden (ms), die zum Schreiben benötigt wird.
MaxBlkReadTime	*UDINT*	40	Längste Zeit in Millisekunden (ms), die zum Lesen eines Datenblocks benötigt wird.
MinBlkReadTime	*UDINT*	41	Kürzeste Zeit in Millisekunden (ms), die zum Lesen eines Datenblocks benötigt wird.
WriteErrorCount	*UDINT*	33	Anzahl der Schreibfehler.
ReadSucceedCount	*UDINT*	35	Anzahl der erfolgreichen Leseversuche.
MaxCycleTime	*UDINT*	22	Längste Zeit in Millisekunden (ms), die zum Lesen aller angeforderten Daten benötigt wurde.
MinCycleTime	*UDINT*	23	Kürzeste Zeit in Millisekunden (ms), die zum Lesen

Name aus Import	Typ	Offset	Erklärung
			aller angeforderten Daten benötigt wurde.
WriteCount	*UDINT*	26	Anzahl der Schreibversuche.
ReadErrorCount	*UDINT*	34	Anzahl der fehlerhaften Leseversuche.
MaxUpdateTimeNormal	*UDINT*	56	Zeit seit letzter Aktualisierung der Prioritätsgruppe **Normal** in Millisekunden (ms).
MaxUpdateTimeHigher	*UDINT*	57	Zeit seit letzter Aktualisierung der Prioritätsgruppe **Höher** in Millisekunden (ms).
MaxUpdateTimeHigh	*UDINT*	58	Zeit seit letzter Aktualisierung der Prioritätsgruppe **Hoch** in Millisekunden (ms).
MaxUpdateTimeHighest	*UDINT*	59	Zeit seit letzter Aktualisierung der Prioritätsgruppe **Höchste** in Millisekunden (ms).
PokeFinish	*BOOL*	55	Geht für eine Abfrage auf *1*, wenn alle anstehenden Pokes ausgeführt wurden.

FEHLERMELDUNGEN

Name aus Import	Typ	Offset	Erklärung
ErrorTimeDW	*UDINT*	2	Zeit (in Sekunden seit 1.1.1970), wann der letzte Fehler auftrat.
ErrorTimeS	*STRING*	2	Zeit (in Sekunden seit 1.1.1970), wann der letzte Fehler als String auftrat.
RdErrPrimObj	*UDINT*	42	Nummer des PrimObjektes, als der letzte Lesefehler verursacht wurde.
RdErrStationsName	*STRING*	43	Name der Station, als der letzte Lesefehler verursacht wurde.
RdErrBlockCount	*UINT*	44	Anzahl der zu lesenden Blöcke, als der letzte Lesefehler verursacht wurde.
RdErrHwAdresse	*DINT*	45	Hardwareadresse, als der letzte Lesefehler verursacht wurde.
RdErrDatablockNo	*UDINT*	46	Bausteinnummer, als der letzte Lesefehler verursacht wurde.
RdErrMarkerNo	*UDINT*	47	Merkernummer, als der letzte Lesefehler verursacht wurde.

Name aus Import	Typ	Offset	Erklärung
RdErrSize	*UDINT*	48	Blockgröße, als der letzte Lesefehler verursacht wurde.
DrvError	*USINT*	25	Fehlermeldung als Nummer.
DrvErrorMsg	*STRING*	30	Fehlermeldung als Klartext.
ErrorFile	*STRING*	15	Name der Fehlerprotokolldatei.

7 Treiberspezifische Funktionen

Die Konfiguration des Mathematik Treibers ist fest vorgelegt und kann nicht geändert werden. Die **Update Zeit** des Treibers beträgt *500 ms*.

Achtung

Wird eine Zählervariable - mit Formel: *Count()* - des Mathematriktreibers in einem Archiv mit *Speicherung bei Wertänderung* (spontan) gespeichert, kann das zur Speicherung von sehr großen Datenmengen führen.

Workaround: Weisen Sie die Mathematik-Variable einer Simulationsvariablen zu (**SIMUL32** Treiber, **Treiberobjekttyp** *HD*) und speichern Sie die Simulationsvariable im Archiv.
Achtung: Die Simulationsvariable muss vom Integer Datentyp (e.g. *INT*, *DINT*) sein, nicht *REAL*.

MATHEMATIK TREIBER IN ZENON NETZWERK

Der Treiber führt die Berechnungen am prozessführenden Server. Diese Ergebnisse werden an alle Clients in Netzwerk verteilt, also auch an den Standby Server.

Am Standby Server berechnet der Treiber im hintergrund, basierend auf aktuellen Werten der Variablen wie vom prozessführenden Server im Netzwerk verteilt; nicht auf Werten der Variablen in internen Puffer des Standby Servers. Erst ein Ausfall des prozessführenden Servers hat zu folge dass der Standby Server für die Berechnungen die Werte von seinen Prozesstreibern verwendet.

Werden in Formeln die lokale (nicht Prozess-) Variablen verknüpft, kann die Ergebnis am Standby Server der von prozessführenden Server abweichen, was aber nicht sichtbar ist solange er ein Standby ist. Nach einer Redundanzumschaltung kann dann der Standby Server, der sich zum Prozessführenden umstuft, die Mathematik Variablen auf ein anderer Wert umschalten - aus eigener Berechnung.

Hinweis: per Default haben die Variablen des Intern Treibers die Eigenschaft **Berechnung** = *lokal*, auch manche der SYSDRV Variablen sind lokal.

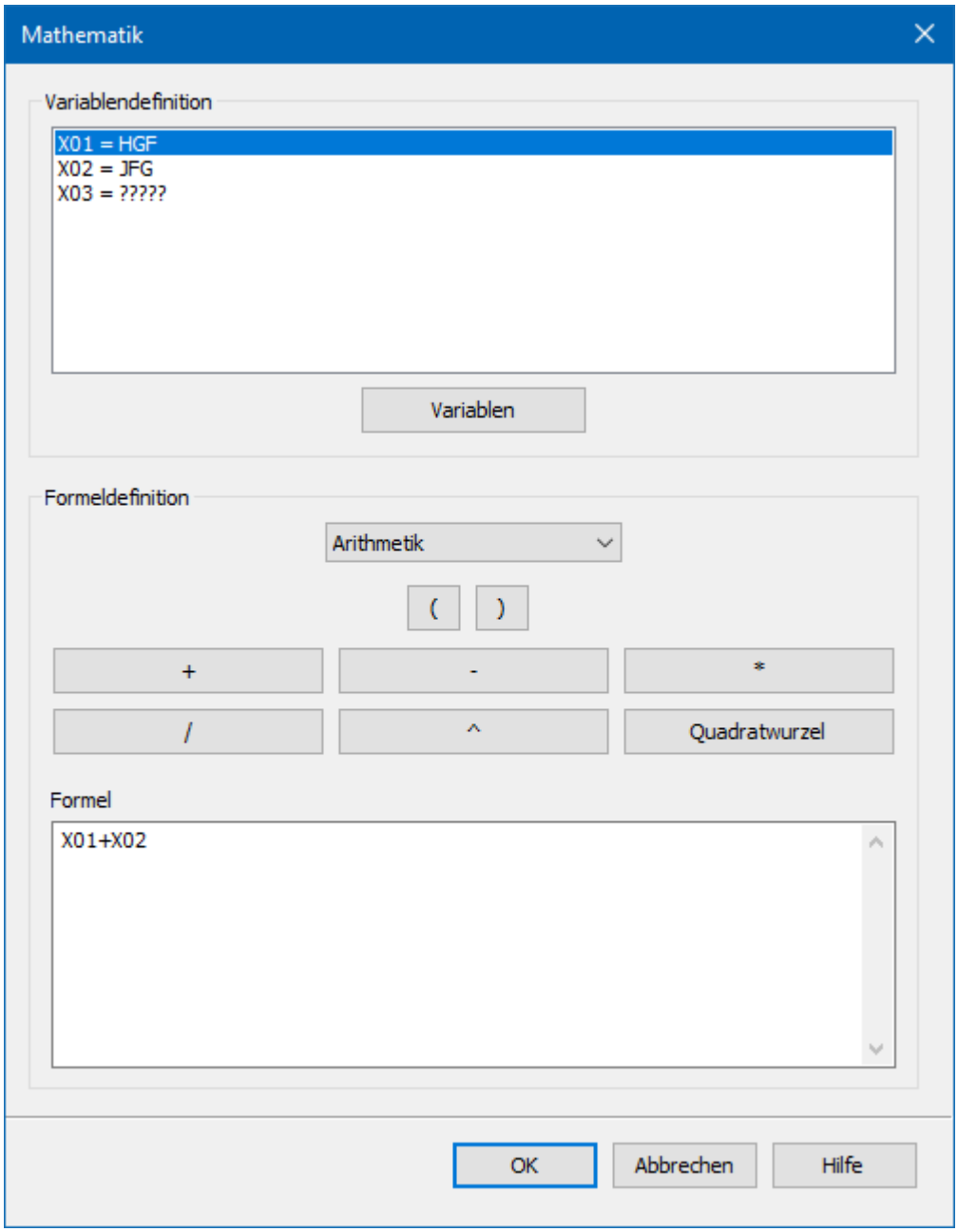
Dieser Treiber unterstützt folgende Funktionen:

7.1 Formeln

Um Formeln einzugeben

- muss eine Variable vom Typ MATHDR32 - Mathematiktreiber angelegt sein
- öffnen Sie in den Eigenschaften der Variablen den Knoten **Wertberechnung**.
- Klicken Sie in das leere Feld neben der Eigenschaft **Formel**.

- der Dialog zur Formeleingabe öffnet sich



Formeln können als Floatformel (auf Seite 42) oder als Bitformeln (auf Seite 41) eingegeben werden:

- Floatformeln:
  - Direkteingabe in Bereich **Formel**

    oder
  - Wahl der Variablen im Fenster **Variablendefinition** und Wahl der Verknüpfungsart durch Klick auf die entsprechenden Schaltflächen für Funktionen im Bereich **Formeldefinition**.

- Bitformeln:
  Direkteingabe in Bereich **Formeln** nach Auswahl von **Boolesche Algebra**. Wurde die Variable als Binärvariable angelegt, wird **Boolesche Algebra** fest vorgegeben.

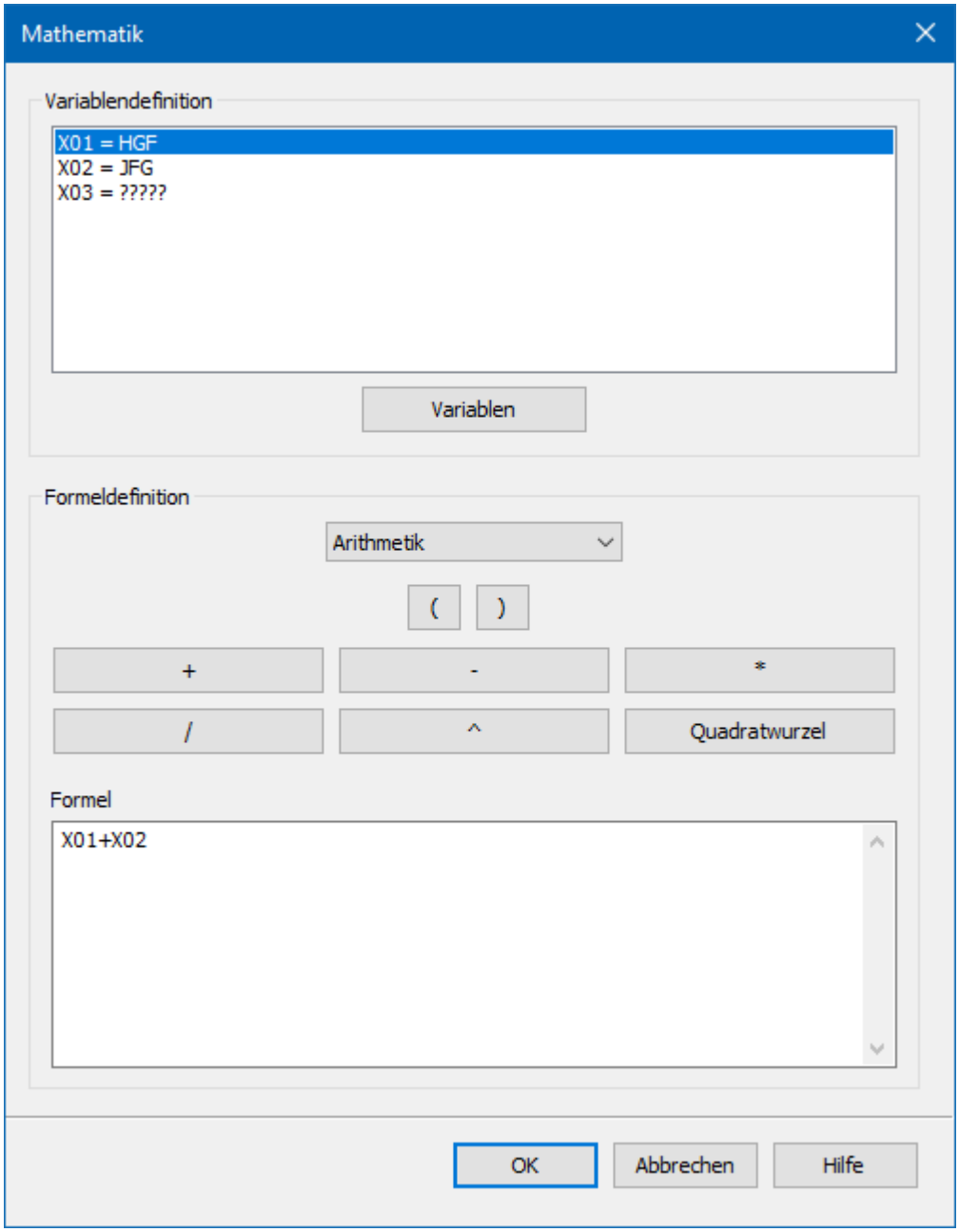
> **Info**
>
> Es können bis zu 99 Variablen in einer Formel verknüpft werden. X01 bis X99.
> Die Länge der Formel darf dabei 4096 Zeichen nicht überschreiten.

Hinweis: Bei der Auswahl einer Funktion werden die nötigen Klammern automatisch erstellt und der Cursor wird zwischen den Klammern platziert.

> **Achtung**
>
> Vermeiden Sie rekursive Aufrufe, wie zum Beispiel eine Mathematik-Variable vom Typ *Statistik*, die sich selbst in **X01** zählt. Rekursive Aufrufe können zur Instabilität des gesamten Systems führen.

7.1.1 Variablen zuordnen

Quellvariablen für die Formeleingabe werden im Feld **Variablendefinition** angezeigt.



Um eine neue Variable hinzuzufügen, eine bestehende Variable zu ändern oder eine Variable zu löschen, öffnen Sie die Variablenauswahl:

1. Klicken Sie auf die Schaltfläche Variablen.

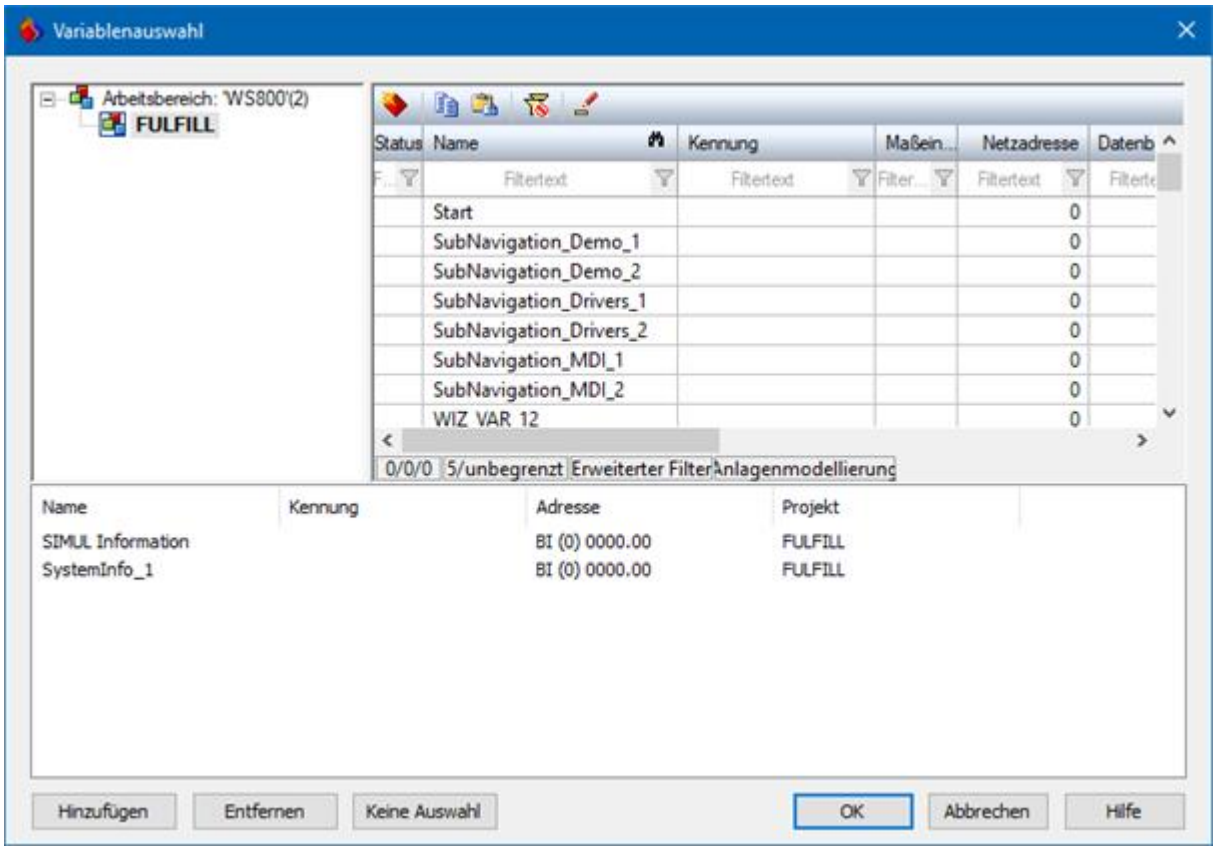
   Oder:

   Klicken Sie im Feld **Variablendefinition** doppelt auf die letzte fortlaufende Nummer mit Fragezeichen als Definitionsmenge.

Der Dialog zu Variablenauswahl wird geöffnet.

2. Fügen Sie neue Variablen hinzu oder löschen sie bestehende aus der Liste.

VARIABLEN HINZUFÜGEN



Element	Beschreibung
Projektbaum	Definition des Projekts, aus dem die Variable gewählt werden soll.
Auswahlfenster	Auswahl der Variablen durch: ▸ Mit Doppelklick wird die ausgewählte Variable wird der Variablenliste hinzugefügt ▸ Mit Drag&Drop kann die ausgewählte Variable in die Variablenliste gezogen werden ▸ Die gewünschten Variablen auswählen. Mit Hilfe der **Steuerung-** bzw. der **Umschalt-Taste** ist eine Mehrfachauswahl möglich. Mit Klick auf **Hinzufügen** werden die ausgewählten Variablen in die Variablenliste übernommen.
Variablenliste	Listet alle ausgewählten Variablen auf.
Hinzufügen	Fügt die aktuell im Auswahlfenster markierte Variable der Variablenliste

Element	Beschreibung
	hinzu.
Entfernen	Entfernt die in der Variablenliste markierte Variable aus der Liste.
Keine Auswahl	Je nach Element: ▸ der Dialog wird abgebrochen ▸ bestimmte Verknüpfungen, wie Chargenvariablen in der Archivierung, können gelöst werden

▸

7.1.2 Verknüpfungsarten

Für die Eingabe von Formeln stehen Ihnen unterschiedliche Funktionen aus folgenden mathematischen Bereichen zur Verfügung:

- **Arithmetik** (kombinierbar mit anderen Bereichen)
- **Trigonometrie** (kombinierbar mit anderen Bereichen)
- **Boolesche Algebra** (kombinierbar mit anderen Bereichen)
- **Statistik** (nicht kombinierbar)
- **Querberechnung** (nicht kombinierbar)
- **Datenreduktion** (nicht kombinierbar)
- **Vergleich** (kombinierbar mit anderen Bereichen)
- **Zählwertverarbeitung** (nur SICAM 230)

Bereich	Ausdruck	Funktion
Arithmetik	Grundfunktionen	
	+	Addition
	-	Subtraktion
	*	Multiplikation
	/	Division
	^	Exponentialberechnung
	SQRT	Quadratwurzel
	()	Klammern

Bereich	Ausdruck	Funktion
Trigonometrie	Winkelfunktionen: Wert wird als Radiant übernommen (0...x*PI)	
	sin	Sinus
	cos	Cosinus
	tan	Tangens
	sinh	Sinus-Hyperbolikus
	cosh	Cosinus-Hyperbolikus
	tanh	Tangens-Hyperbolikus
	()	Klammern
Boolesche Algebra	Logische Verknüpfungen: Variablen werden nur auf den logischen Wert '0' überprüft, ist der Wert ungleich '0', wird er mit '1' definiert. Im Unterschied zur Bitformel kann der technische Bereich der Mathematikvariablen durch einen Streckungsfaktor verändert werden (ungleich '0' oder '1').	
	AND	logisch 'UND'
	OR	logisch 'ODER'
	XOR	logisch 'EXKLUSIV ODER'
	NOT	Negation
	()	Klammern
Statistik	Wert wird erst am Ende des Berechnungszeitraumes (Anzahl der Ereignisse) bereitgestellt.	
	GltndMW	gleitender Mittelwert über Änderungen
	GltndStdabw	gleitende Standardabweichung Änderungen

Bereich	Ausdruck	Funktion
	Zähler	Zählerfunktion
	()	Klammern
Querberechnung		
	QuerMW	Mittelwertberechnung mehrerer Kanäle gleichzeitig
	()	Klammern
Datenreduktion		
	TMittelung	Mittelwert über Zeit oder Ereignisse
	Summe	Summe über Zeit oder Ereignisse
	Max	Maximum über Zeit oder Ereignisse
	Min	Minimum über Zeit oder Ereignisse
	Dif	Differenz über Zeit
	()	Klammern
Vergleich		
	<	kleiner als
	>	größer als
	<=	kleiner gleich
	>=	größer gleich
	<>	kleiner oder größer
	=	gleich
	()	Klammern
Zählwertverarbeitung	Für die Verknüpfungen mit Zeit- oder Ereignisbezug (Statistik, Datenreduktion) sind zusätzliche Eingaben notwendig. Diese sind beim Beenden der Verknüpfung anzugeben.	nur für SICAM 230!

7.1.3 Bitformeln

Bitformeln eignen sich für Formelergebnisse mit LReal Mathematikvariablen, die nur einen logischen Zustand Low oder High besitzen. Im Unterschied zu Floatformeln mit Boolescher Verarbeitung (auf Seite 43) ist der Rohwert bereits fix vordefiniert (0,1).

Bitformeln eingeben:

- Zuordnung des Variablennamens
- optional: Eingaben für den technischen Bereich, Grenzwert und Updateverhalten
- Formelzuweisung

Als Verknüpfung stehen nur die Booleschen Operatoren zur Verfügung (siehe Kapitel Booleesche Algebra (auf Seite 43).)

Im Mathematiktreiber können auch Werte, z. B. Mehrbitmeldungen, bitweise über Boolesche Verknüpfungen definiert werden. Dazu muss das zu verknüpfende Bitmuster der Variablen nach einer Verknüpfungsanweisung (=Steuerzeichen) in einer Bitformel dezimal angegeben werden (z. B. **Bit0** und **Bit1** ->*3*).

Syntax	Beschreibung
(X01)NAND(X02)	NOT AND
(X01)NOR(X02)	NOT OR
(X01&3)	Mit dem Wert der Variablen X01 werden **Bit0**=*1* und **Bit1**=*1* **AND**-verknüpft
(X01|3)	Mit dem Wert der Variablen X01 werden **Bit0**=*1* und **Bit1**=*1* **OR**-verknüpft
(X01#3)	Mit dem Wert der Variablen X01 werden **Bit0**=*1* und **Bit1**=*1* **XOR**-verknüpft

BEISPIELE

Formel	Werte (binär)	Resultat	= numerisch	= Bool
für: X01=9 (1001b)				
X01|5	*1001|101*	**1101b**	*13*	*true*
X01&5	*1001&101*	**0001b**	*1*	*true*
X01#5	*1001#101*	**1100b**	*12*	*true*
für X01=4 (0100b)				
X01&2	*0100&010*	**0000b**	*0*	*false*

7.1.4 Floatformel

Der Mathematikinterpreter in zenon stellt für Floatformeln Funktionen aus folgenden Bereichen zur Verfügung:

- Arithmetik (auf Seite 42)
- Trigonometrie (auf Seite 43)
- Boolesche Algebra (auf Seite 43)
- Statistik (auf Seite 44)
- Querberechnung (auf Seite 50)
- Datenreduktion (auf Seite 51)
- Vergleich (auf Seite 64)
- Zählwertverarbeitung (auf Seite 64)

sowie einige weitere Floatformeln (auf Seite 64), die direkt eingegeben werden müssen.

7.1.4.1 Arithmetik

Wählen Sie im Bereich Formel Definition in der Dropdownliste **Arithmetik** aus. Folgende Grundfunktionen stehen Ihnen zur Verfügung:

Ausdruck	Funktion
+	Addition
-	Subtraktion
*	Multiplikation
/	Division
^	Exponentialberechnung
SQRT	Quadratwurzel
()	Klammern

Arithmetik-Funktionen können kombiniert werden mit:

- Trigonometrie (auf Seite 43)
- Boolesche Algebra (auf Seite 43)
- Vergleich (auf Seite 64)

7.1.4.2 Trigonometrie

Wählen Sie im Bereich **Formel Definition** in der Dropdownliste **Trigonometrie** aus. Folgende Winkelfunktionen stehen Ihnen zur Verfügung, der Wert wird als Radiant übernommen (0...x*PI):

Ausdruck	Funktion
sin	Sinus
cos	Cosinus
tan	Tangens
sinh	Sinus-Hyperbolikus
cosh	Cosinus-Hyperbolikus
tanh	Tangens-Hyperbolikus
()	Klammern

Trigonometrie-Funktionen können kombiniert werden mit:

- Arithmetik (auf Seite 42)
- Boolesche Algebra (auf Seite 43)
- Vergleich (auf Seite 64)

7.1.4.3 Boolesche Algebra

Wählen Sie im Bereich **Formel Definition** in der Dropdownliste **Boolesche Algebra** aus. Folgende Ausdrücke stehen Ihnen zur Verfügung:

Ausdruck	Funktion
AND	logisch 'UND'
OR	logisch 'ODER'
XOR	logisch 'EXKLUSIV ODER'
NOT()	Negation **Achtung:** Der Ausdruck nach *NOT* muss in Klammern stehen.

Ausdruck	Funktion
()	Klammern

Die Variablen werden bei logischen Verknüpfungen nur auf den logischen Wert '0' überprüft. Ist der Wert ungleich '0', wird er mit '1' definiert. Im Unterschied zu Bitformeln (auf Seite 41) kann der technische Bereich der Mathematikvariablen durch einen Streckungsfaktor verändert werden (ungleich '0' oder '1').

Boolesche Algebra kann kombiniert werden mit:

- Arithmetik (auf Seite 42)
- Trigonometrie (auf Seite 43)
- Vergleich (auf Seite 64)

7.1.4.4 Statistik

Wählen Sie im Bereich **Formel Definition** in der Dropdownliste **Statistik** aus. Folgende Funktionen stehen Ihnen zur Verfügung:

Ausdruck	Funktion
GltndMW	gleitender Mittelwert über Änderungen (auf Seite 45)
GltndStdabw	gleitende Standardabweichung (auf Seite 46)
Zähler	Zählerfunktion (auf Seite 47)
()	Klammern

Der Wert wird erst am Ende des Berechnungszeitraumes (Anzahl der Ereignisse) bereitgestellt.

Statistik-Funktionen können mit keinen weiteren Funktionen kombiniert werden.

Info

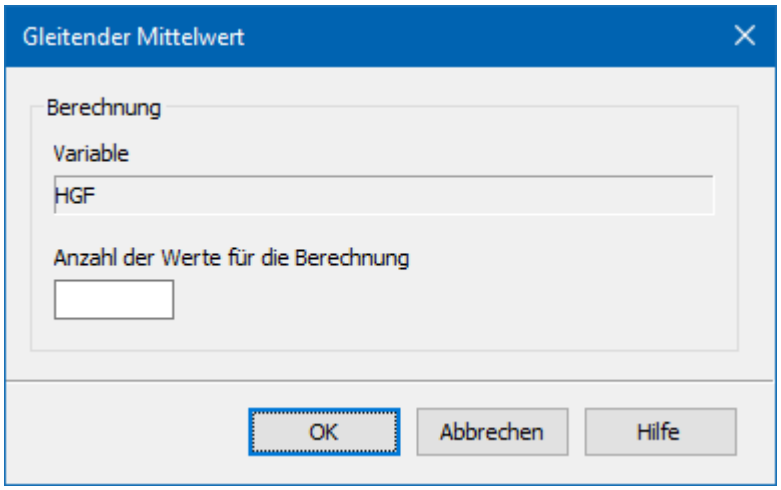
Werte des Mathematik Treibers werden beim normalen Beenden der Runtime gespeichert. Um die Speicherung der Werte sicher zu stellen, wenn Grenzwerte oder Rema-Bedingungen, bei denen der Zähler läuft, inaktiv werden:

Führen Sie die Funktion **AML- und CEL-Ringpuffer speichern** oder die Funktion **Remanentdaten speichern** aus. Zusätzlich können Sie die Werte noch zyklisch (z.B jede Stunde) speichern.

7.1.4.4.1 Gleitender Mittelwert

Es wird der Mittelwert über eine definierte Zahl von Werten berechnet.

Beim Bestätigen der Funktion wird ein Dialog zur Eingabe der Anzahl der Werte geöffnet:



Option	Beschreibung
Berechnung	Konfiguration der Berechnung für die angegebene Variable.
Anzahl der Werte für die Berechnung	Anzahl der Werte, für die die Berechnung durchgeführt wird. Nach Erreichung der Anzahl wird die Berechnung durchgeführt und mit jedem weiteren Wert um eines verschoben.
OK	Übernimmt Eingabe und schließt Dialog sowie Formeleingabe.
Abbrechen	Verwirft Eingabe, schließt Dialog und kehrt zur Formeleingabe zurück.
Hilfe	Öffnet Onlinehilfe.

Beispiel

Es werden *10* Werte als Vorgabe eingestellt.

Ablauf in der Runtime:

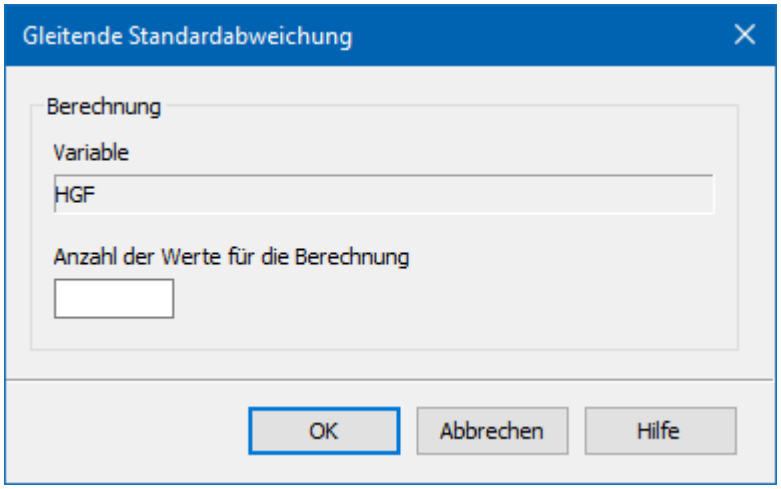
- Beginnend mit dem Ausgangswert wird *10* Werte lang gewartet und dann daraus der Mittelwert berechnet.
- Nach den ersten 10 Werten wird der der erste Wert aus der Berechnung entfernt und der neue 11. Wert in die Berechnung übernommen usw.

Der Bereich für die Berechnung gleitet also mit der wachsenden Zahl der Werte mit.

7.1.4.4.2 Gleitende Standardabweichung

Es wird die Standardabweichung über eine definierte Anzahl von Werten. Die erste Ausgabe des Wertes erfolgt, wenn alle Änderungen vorhanden sind.

Beim Bestätigen der Funktion wird ein Dialog zur Eingabe der Anzahl der Werte geöffnet:



Option	Beschreibung
Berechnung	Konfiguration der Berechnung für die angegebene Variable.
Anzahl der Werte für die Berechnung	Anzahl der Werte, für die die Berechnung durchgeführt wird. Nach Erreichung der Anzahl wird die Berechnung durchgeführt und mit jedem weiteren Wert um eines verschoben.

Option	Beschreibung
OK	Übernimmt Eingabe und schließt Dialog sowie Formeleingabe.
Abbrechen	Verwirft Eingabe, schließt Dialog und kehrt zur Formeleingabe zurück.
Hilfe	Öffnet Onlinehilfe.

Beispiel

Es werden *10* Werte als Vorgabe eingestellt.

Ablauf in der Runtime:

- Beginnend mit dem Ausgangswert, wird *10* Werte lang gewartet und dann daraus die Standardabweichung berechnet.
- Nach den ersten 10 Werten wird der der erste Wert aus der Berechnung entfernt und der neue 11. Wert wird in die Berechnung übernommen usw.

Der Bereich für die Berechnung gleitet also mit der wachsenden Zahl der Werte mit.

7.1.4.4.3Zähler

Die Zählerfunktion bietet die Möglichkeit, eine Zählervariable zu erhöhen, wenn der Rema-Zustand der Quellvariablen oder der Grenzwert der Quellvariablen erreicht wurde. Damit können z. B. die Betriebsstunden einer Anlage gezählt werden und mit der Zählervariablen das Signal für die nötige Wartungsarbeit gegeben werden.

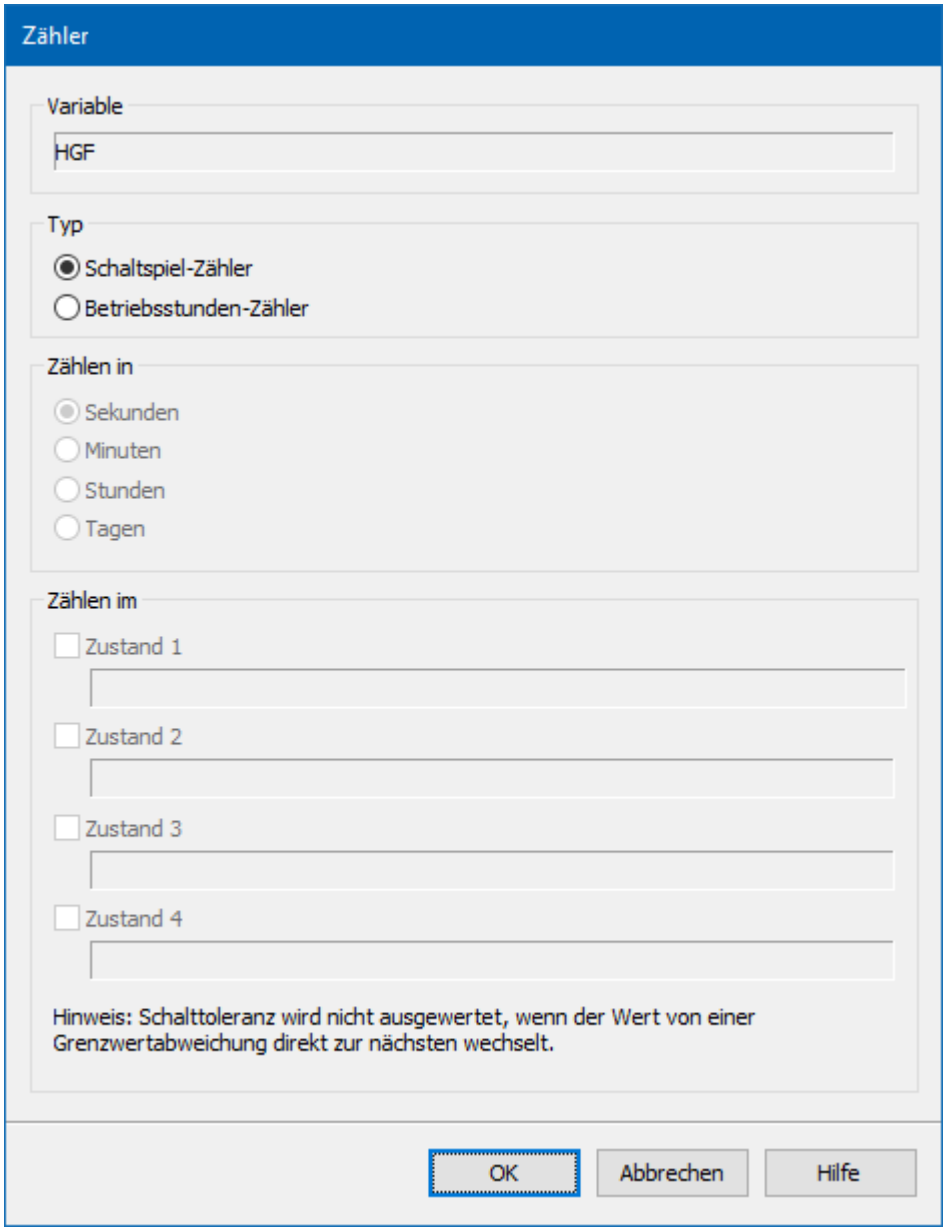
Achtung

Wird eine Zählervariable - mit Formel: *Count()* - des Mathematriktreibers in einem Archiv mit *Speicherung bei Wertänderung* (spontan) gespeichert, kann das zur Speicherung von sehr großen Datenmengen führen.

Workaround: Weisen Sie die Mathematik-Variable einer Simulationsvariablen zu (**SIMUL32** Treiber, **Treiberobjekttyp** *HD*) und speichern Sie die Simulationsvariable im Archiv.
Achtung: Die Simulationsvariable muss vom Integer Datentyp (e.g. *INT*, *DINT*) sein, nicht *REAL*.

ZÄHLER KONFIGURIEREN

Mit Klick auf OK im Statistik-Dialog wird der Dialog zur Konfiguration von Zählerart und den zu zählenden Zuständen bzw. Grenzwerten geöffnet:



VARIABLE

Anzeige der zu konfigurierenden Variable

TYP

Zählertyp. Wahl zwischen Schaltspiel-Zähler und Betriebsstunden-Zähler:

Option	Beschreibung
Schaltspiel-Zähler	Zählt das Erreichen des Zustandes bzw. Grenzwertes nach dem Ablauf einer eventuell projektierten Verzögerungszeit. **Hinweis:** Schaltspiele werden nicht ausgewertet, wenn der Wert von einer Grenzwertverletzung direkt zur nächsten wechselt. So können zum Beispiel Grenzwertverletzungen von Binärvariablen nicht mit dem **Schaltspielzähler** ausgewertet werden.
Betriebsstunden-Zähler	Zählt die Zeit solange die Variable der gewählte Zustand bzw. Grenzwert hat. Die Zeit eventueller Kommunikationsstörung wird dabei berücksichtigt. **Beispiel:** angenommen der Zähler soll die Betriebsstunden des Wertes 1 zählen. Wenn der letzte Wert *1* ist und: ▸ die Kommunikation auf *gestört* geht (Variable hat statusbit *INVALID*) oder ▸ die Runtime beendet wird wird der Zähler weiter zählen. Sobald die Kommunikation wieder hergestellt und der Wert auf *0* ist, stoppt der Zähler.

ZÄHLEN IN

Aktivierung der Zustände, in denen gezählt werden soll.

Option	Beschreibung
Zählen in	Wurde die Quellvariable mit einer Reaktionsmatrix verknüpft ist, entsprechen die Zustände 1 - 4 den Zählernummern der Rema. Wurde die Quellvariable nicht mit einer Reaktionsmatrix verknüpft, entsprechen die möglichen Zustände den Grenzwerten.

DIALOG BEENDEN

Option	Beschreibung
OK	Übernimmt Einstellungen und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

ZÄHLWEISE

Der Zähler zählt nur neue Verletzungen der Summe aller projektierten Grenzwerte, aber nicht Verletzungen einzelner Werte. Das bedeutet:

- Der Zähler-Wert wird nicht erhöht, wenn:
  - bereits ein Grenzwert verletzt und
  - der Variablenwert sich ändert, so dass erneut ein Grenzwert verletzt wird
- Der Zähler-Wert wird nicht erhöht, wenn:
  - der Variablenwert sich innerhalb eines Grenzwerts ändert und
  - die Eigenschaft **Jede Wertänderung als neue Grenzwertverletzung behandeln** aktiviert ist
- Der Zähler-Wert wird nur erhöht, wenn
  - keiner der 4 Zustände zuvor verletzt wurde
  - zumindest einer der 4 Zustände nach dem Update des Variablenwerts verletzt wurde

7.1.4.5 Querberechnung

Die Querberechnung erlaubt die Bildung des Mittelwertes über mehrere Variablen.

Hinweis: Querberechnungen können mit keinen weiteren Funktionen kombiniert werden.

Wählen Sie im Bereich **Formel Definition** in der Dropdownliste **Querberechnung** aus. Folgende Funktionen stehen Ihnen zur Verfügung:

Ausdruck	Funktion
CrossMean	Mittelwertberechnung mehrerer Kanäle gleichzeitig.
()	Klammern

Achtung: Bei Berechnungen mit der Funktion **CrossMean()** werden unabhängig von den gewählten Parametern immer alle in der Mathematik-Variable gewählten Quellvariablen verwendet.

Beispiel: Die Liste enthält 5 Variablen, 3 werden ausgewählt. Das Ergebnis der Berechnung mit **CrossMean()** bezieht sich auf alle 5 vorhandenen Variablen.

7.1.4.6 Datenreduktion

Wählen Sie im Bereich **Formel Definition** in der Dropdownliste **Datenreduktion** aus. Folgende Funktionen stehen Ihnen zur Verfügung:

Ausdruck	Funktion
TMittelung	Mittelwert über Zeit oder Ereignisse (auf Seite 52)
Summe	Summe über Zeit oder Ereignisse (auf Seite 55)
Max	Maximum über Zeit oder Ereignisse (auf Seite 58)
Min	Minimum über Zeit oder Ereignisse (auf Seite 60)
Dif	Differenz über Zeit (auf Seite 62)
()	Klammern

Datenreduktion kann mit keinen weiteren Funktionen kombiniert werden.

HINWEISE ZUR BERECHNUNG

SUMME UND ZEITMITTELWERT

Die Werte werden am Ende des Intervalls berechnet und angezeigt. Das Ende wird über die Optionen **zeitlich begrenzt** oder **ereignisgesteuer**t bestimmt. Wird die Option **laufend** gewählt, dann erfolgt die Anzeige auch entsprechend dem Abtastverhalten der Quellvariable. Das Abtastverhalten entspricht dem neuen Wert oder der Abtastrate. Die Abtastrate ist mit 0,5 Sekunden oder ereignisgesteuert definiert. Der Mittelwert wird aus der Summe durch die entsprechende Division (z.B. Anzahl der Wertänderungen) gebildet.

Optionen:

- **Initialisierung am Anfang des Intervalls**:

  Der Wert vom Ende des Intervalls bleibt angezeigt.

  - Option **mit bestehendem Wert** aktiv: Der letzte Messwert wird zur neuen Summe. Nach dem ersten eintreffenden Wert wird der Durchschnitt aus diesen beiden Werten gebildet - und so weiter.
  - Option **mit neuem Wert** aktiv: Die Summe beginnt mit 0. Zur Bildung des Durchschnitts werden nur die neu eintreffenden Werte herangezogen. Treffen einen Intervall lang keine neuen Werte ein, wird die Mittelwert-Variable ungültig (**I_BIT**).

- **Berechnen als:**
  - **Integral über die Zeit**: Ist eine zeitgewichtete Summe. Es wird jeweils der vorhergehende Wert aufsummiert, multipliziert mit der Dauer der Abtastperiode in Sekunden, entsprechend dem Abtastverhalten. Der Durchschnitt wird mit Division durch die Intervalldauer gebildet.

Verhalten beim Sollwertsetzen:

- Wird ein Sollwert abgesetzt, wirkt das wie ein Neustart des Intervalls. Der Sollwert muss dafür nicht *0* sein.
- Für **mit bestehendem Wert** wird der Sollwert als aktuelle Summe übernommen.
- Für **mit neuem Wert** wird die Summe auf *0* gesetzt. Der Durchschnitt wird berechnet. Das abgebrochene Intervall wird ausgewertet. Ist **Ausgabe am Ende des Intervalls** gewählt, bleibt dieser Wert in der Anzeige. Für **Ausgabe** laufend erscheint sofort die aktuelle Summe oder der Durchschnitt.

MAXIMUM UND MINIMUM

Die Anzeige hängt ab von den gewählten Optionen für die Ausgabe und für die Initialisierung.
Optionen:

- **Ausgabe des Ergebnis**:
  Das Extremum wird bei jeder Wertänderung berechnet und je nach gewählter Option ausgegeben:
  - Option **am Ende des Intervalls**: Das Extremum wird am Ende ausgegeben.
  - Option **laufend**: Das Extremum wird laufend aktualisiert angezeigt.
- **Initialisierung zu Beginn des Intervalls**:
  Referenz für die nächste Anzeige:
  - Option **mit bestehendem Wert**: Der letzte Messwert wird zum aktuellen Extremum.
  - Option **mit neuem Wert**: Das Extremum wird gelöscht. Treffen ein Intervall lang keine neuen Werte ein, wird die Extremum-Variable ungültig (**I_BIT**).

Verhalten beim Sollwertsetzen:

- Der Sollwert wird zum neuen Extremum, unabhängig von der gewählten Option für **Initialisierung zu Beginn des Intervalls**.
- Mit Option **laufend** wird der Wert sofort angezeigt.

7.1.4.6.1 Zeitmittelwert

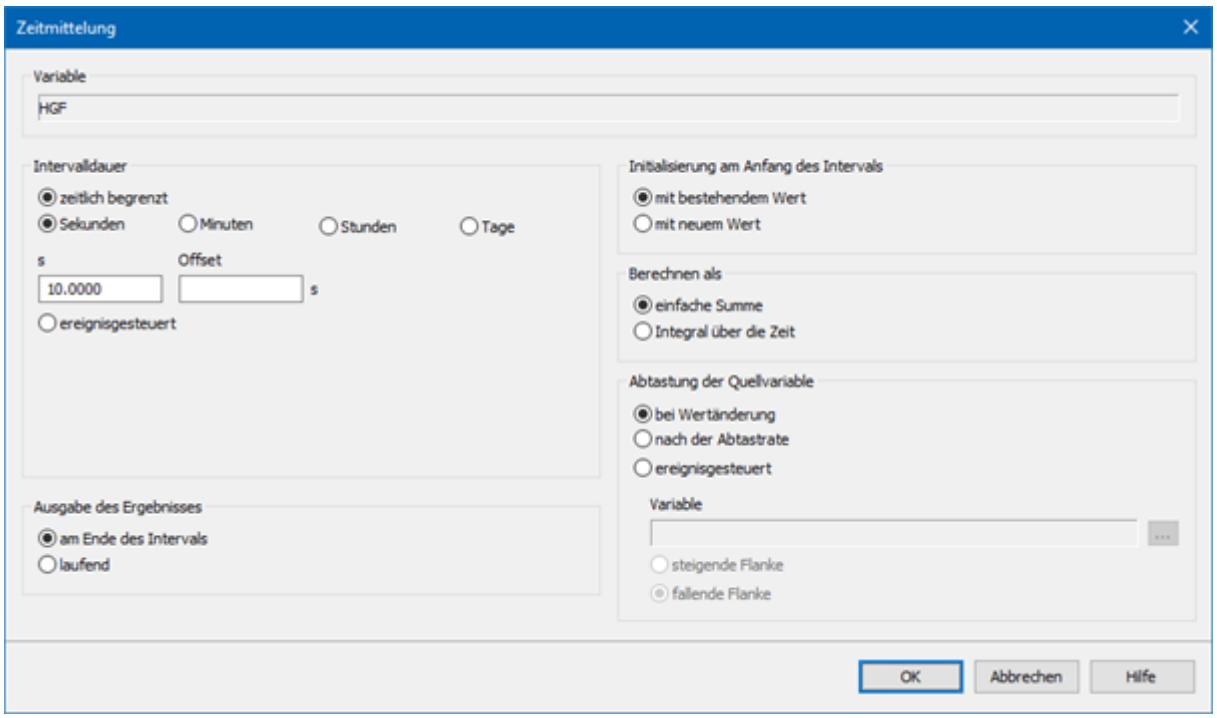
Die Datenreduktion **Zeitmittelung** erlaubt die Bildung eines Mittelwertes einer Variablen. Mögliche Mittelungen sind:

- Zeitintervall:
  Mittelwertbildung über Zeitintervall (Sekunden, Minuten, Stunden, Tage)
- Ereignisintervall:
  Mittelwert wird gebildet, wenn eine Änderung einer Ereignisvariablen auftritt

Weitere Parameter der Projektierung sind:

- Ausgabe: Ergebnis wird am Ende des Intervalls oder ständig bereitgestellt
- Initialisierung: am Anfang des Intervalls wird der Wert initialisiert (bestehender oder neuer Wert)
- Berechnungsart: Festlegung der Berechnungsart (Summe, Integral über die Zeit)
- Abtastart: Art der Abtastung der Quellvariablen (spontan, nach Updateparametrierung, ereignisgesteuert)

Wird die Schaltfläche **OK** betätigt, öffnet sich ein Dialog, in dem die Parameter der Mittelwertbildung eingegeben werden.



VARIABLE

Anzeige der Ereignisvariablen.

INTERVALLDAUER

Option	Beschreibung
zeitlich begrenzt	Mittelwertbildung über die Zeit.
ereignisgesteuert	Mittelwert wird bei Änderung einer Ereignisvariablen gebildet.
s	Eingabe des Zahlenwertes für entsprechende Zeit (echtzeitsynchron, d.h. Eingabe 15 bedeutet 15, 30, 45, 0 etc.).
Sekunden, Minuten, Stunden, Tage	Auswahl des Zeitintervalls.
Offset	Verzögerung für Start und Ende der Mittelwertbildung.

Ausgabe des Ergebnisses

Parameter	Beschreibung
am Ende des Intervalls	Wert wird erst am Ende des Intervalls bereitgestellt, in der Zwischenzeit steht der vorherige Wert zur Verfügung.
laufend	Wert wird bei jeder Abtastung der Quellvariablen aktualisiert.

INITIALISIERUNG AM ANFANG DES INTERVALLS

Option	Beschreibung
mit bestehendem Wert	Wert basiert auf dem vorherigen Wert.
mit neuem Wert	Wert muss neu berechnet werden.

Berechnen als

Parameter	Beschreibung
einfache Summe	Addition der Werte.
Integral über die Zeit	Integration der Werte über das Zeitintervall.

ABTASTUNG DER QUELLVARIABLE

Konfiguration der Quellvariable durch folgende Optionen:

Option	Beschreibung
bei Wertänderung	Spontane Änderungen der Quellvariablen werden sofort

Option	Beschreibung
	für die Berechnung berücksichtigt.
nach der Abtastrate	Quellwerte werden nur in der definierten Updaterate für die Berechnung herangezogen (Schaltfläche **Update, ist fix eingestellt auf 0,5 sec.**).
ereignisgesteuert	Quellwert wird für die Berechnung herangezogen, wenn bei der projektierten binären Ereignisvariable ein Flankenwechsel erfolgt.
Variable	Auswahl der binären Ereignisvariablen. Klick auf die Schaltfläche **...** öffnet den Auswahldialog. Festlegen des entsprechenden Flankenwechsels für die Übernahme des Quellewertes über Radiobuttons: ▸ **steigende Flanke** ▸ **fallende Flanke** Nur verfügbar, wenn die Option **ereignisgesteuert** aktiv ist.

7.1.4.6.2 Summenbildung

Die Datenreduktion **Summenbildung** erlaubt es, die Summe einer Variablen zu bilden.

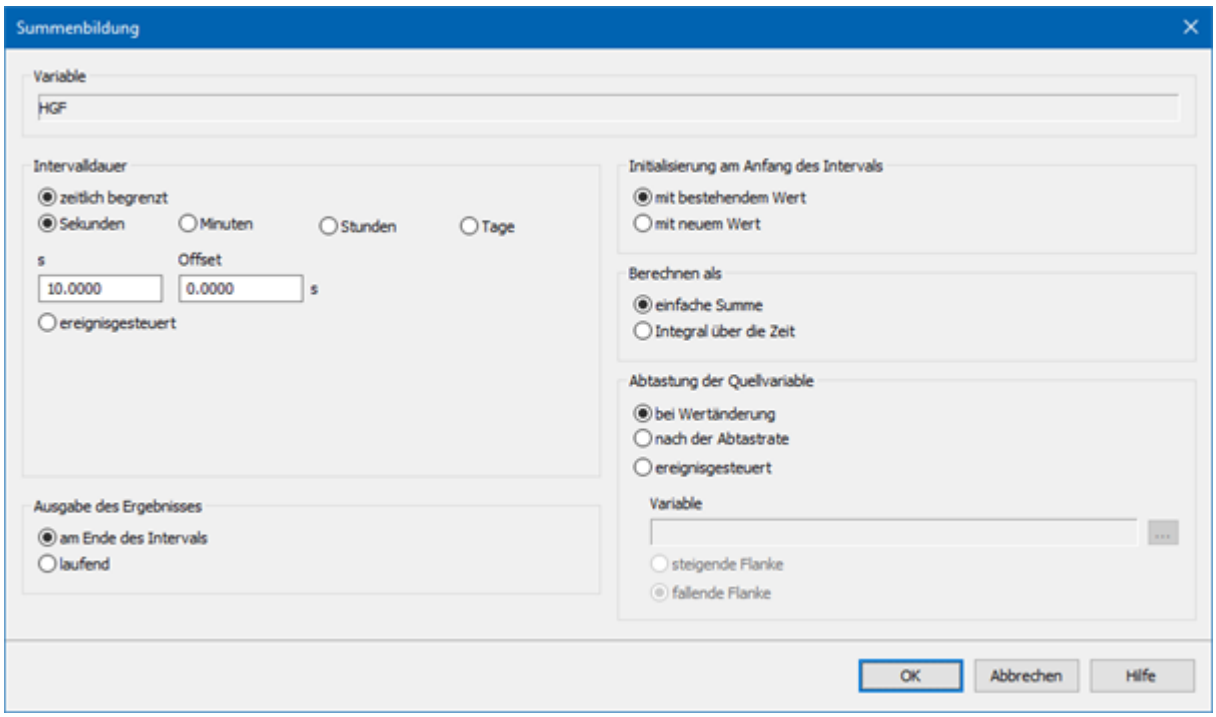
Mögliche Summenbildungen sind:

- Zeitintervall:
  Summenbildung über Zeitintervall (Sekunden, Minuten, Stunden, Tage)
- Ereignisintervall:
  Summe wird gebildet, wenn eine Änderung einer Ereignisvariablen auftritt

Weitere Parameter der Projektierung sind:

- Ausgabe: Ergebnis wird am Ende des Intervalls oder ständig bereitgestellt
- Initialisierung: am Anfang des Intervalls wird der Wert initialisiert (bestehender oder neuer Wert)
- Berechnungsart: Festlegung der Berechnungsart (Summe, Integral über die Zeit)
- Abtastart: Art der Abtastung der Quellvariablen (spontan, nach Updateparametrierung, ereignisgesteuert)

Wird die Schaltfläche **OK** betätigt, öffnet sich ein Dialog, in dem die Parameter der Summenbildung eingegeben werden:



Variable
Anzeige der Ereignisvariablen

INTERVALLDAUER

Option	Beschreibung
zeitlich begrenzt	Summenbildung über die Zeit.
ereignisgesteuert	Summe wird bei Änderung einer Ereignisvariablen gebildet.
s	Eingabe des Zahlenwertes für entsprechende Zeit (echtzeitsynchron, d.h. Eingabe 15 bedeutet 15, 30, 45, 0 etc.).
Sekunden, Minuten, Stunden, Tage	Auswahl des Zeitintervalls.
Offset	Verzögerung für Start und Ende der Summenbildung.

AUSGABE DES ERGEBNISSES

Option	Beschreibung
am Ende des Intervalls	Wert wird erst am Ende des Intervalls bereitgestellt, in der Zwischenzeit steht der vorherige Wert zur Verfügung.
laufend	Wert wird bei jeder Abtastung der Quellvariablen aktualisiert.

INITIALISIERUNG AM ANFANG DES INTERVALLS

Option	Beschreibung
mit bestehendem Wert	Wert basiert auf dem vorherigen Wert.
mit neuem Wert	Wert muss neu berechnet werden.

BERECHNEN ALS

Option	Beschreibung
einfache Summe	Addition der Werte.
Integral über die Zeit	Integration der Werte über das Zeitintervall.

ABTASTUNG DER QUELLVARIABLE

Konfiguration der Quellvariable durch folgende Optionen:

Option	Beschreibung
bei Wertänderung	Spontane Änderungen der Quellvariablen werden sofort für die Berechnung berücksichtigt.
nach der Abtastrate	Quellwerte werden nur in der definierten Updaterate für die Berechnung herangezogen (Schaltfläche **Update, ist fix eingestellt auf 0,5 sec.**).
ereignisgesteuert	Quellwert wird für die Berechnung herangezogen, wenn bei der projektierten binären Ereignisvariable ein Flankenwechsel erfolgt.
Variable	Auswahl der binären Ereignisvariablen. Klick auf die Schaltfläche **...** öffnet den Auswahldialog. Festlegen des entsprechenden Flankenwechsels für die Übernahme des Quellwertes über

Option	Beschreibung
	Radiobuttons: ▸ **steigende Flanke** ▸ **fallende Flanke** Nur verfügbar, wenn die Option **ereignisgesteuert** aktiv ist.

7.1.4.6.3 Maximumbildung

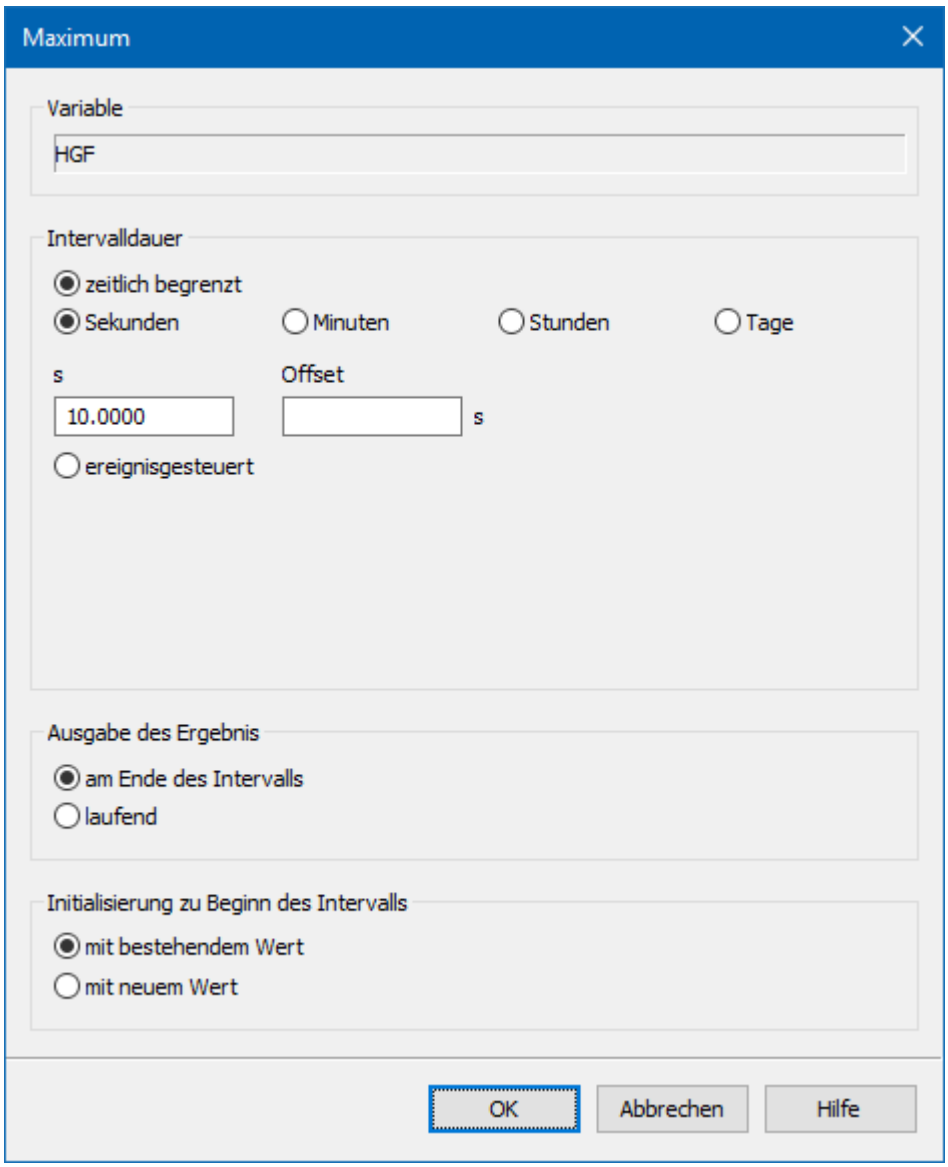
Die Datenreduktion Maximum bietet die Möglichkeit, das Maximum einer Variablen innerhalb des Intervalls zu ermitteln. Mögliche Optionen sind:

- Zeitintervall:
  Maximum innerhalb eines Zeitintervalls (Sekunden, Minuten, Stunden, Tage)
- Ereignisintervall:
  Maximum wird gebildet, wenn eine Änderung einer Ereignisvariablen auftritt

Weitere Parameter der Projektierung sind:

- Ausgabe: Ergebnis wird am Ende des Intervalls oder ständig bereitgestellt
- Initialisierung: am Anfang des Intervalls wird der Wert initialisiert (bestehender oder neuer Wert)

Wird die Schaltfläche OK betätigt, öffnet sich ein Dialog, in dem die Parameter der Maximumermittlung eingegeben werden:



Variable: Anzeige der Ereignisvariablen

INTERVALLDAUER

Option	Beschreibung
zeitlich begrenzt	Maximum über die Zeit.
Sekunden, Minuten, Stunden, Tage	Auswahl des Zeitintervalls.
s	Eingabe des Zahlenwertes für entsprechende Zeit

Option	Beschreibung
	(echtzeitsynchron, d.h. Eingabe 15 bedeutet 15, 30, 45, 0 etc.).
Offset	Verzögerung für Start und Ende der Maximumbildung.
ereignisgesteuert	Maximum wird bei Änderung einer Ereignisvariablen gebildet.

AUSGABE DES ERGEBNIS

Option	Beschreibung
am Ende des Intervalls	Wert wird erst am Ende des Intervalls bereitgestellt, in der Zwischenzeit steht der vorherige Wert zur Verfügung.
laufend	Wert wird bei jeder Abtastung der Quellvariablen aktualisiert.

INITIALISIERUNG ZU BEGINN DES INTERVALLS

Option	Beschreibung
mit bestehendem Wert	Wert basiert auf dem vorherigen Wert.
mit neuem Wert	Wert muss neu berechnet werden.

Info

Der Synchronzeitpunkt wird abhängig vom Zeitintervall ermittelt.

Beispiel: Wird als Zeitintervall 15 Minuten vorgegeben, so wird die Berechnung alle 15 Minuten, ausgehend von der vollen Stunde, durchgeführt.

7.1.4.6.4Minimumbildung

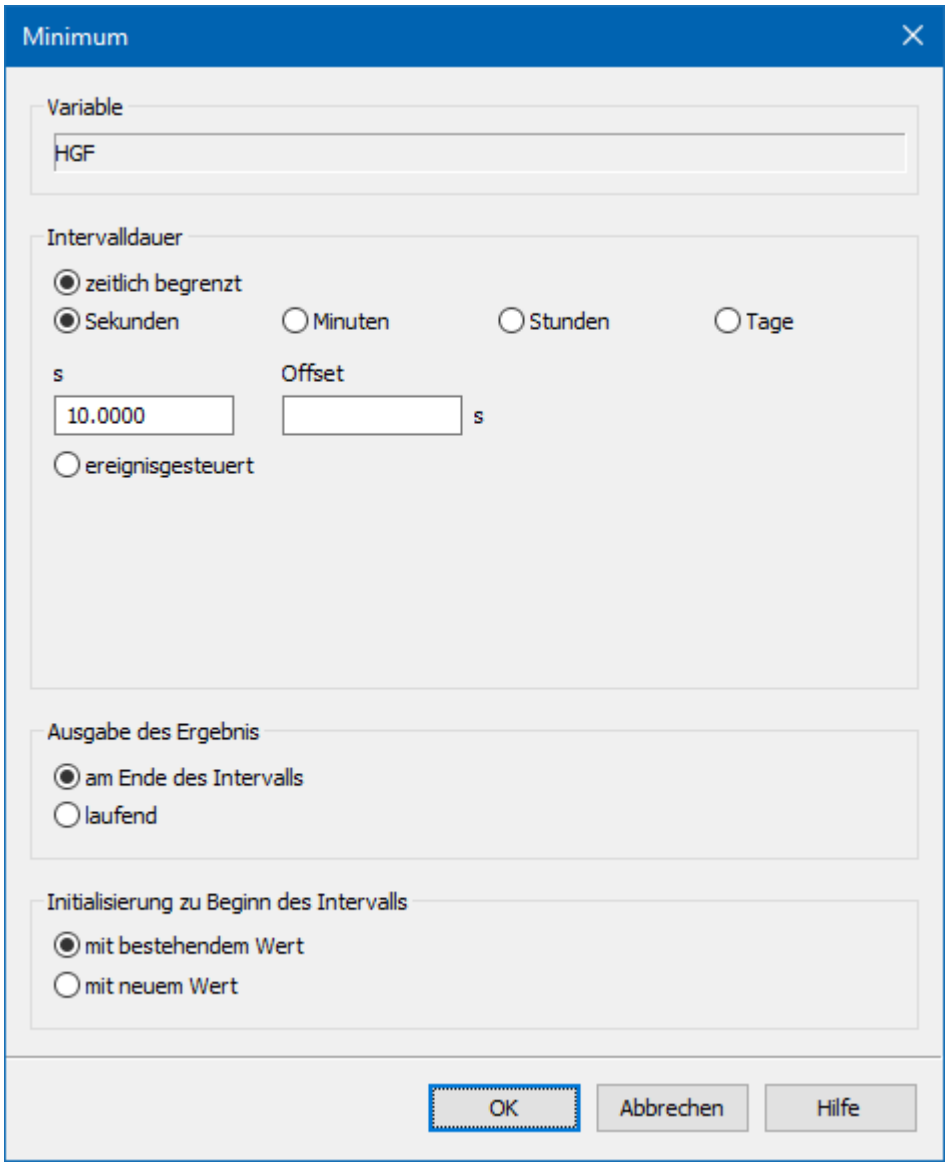
Die Datenreduktion Minimum bietet die Möglichkeit, das Minimum einer Variablen innerhalb des Intervalls zu ermitteln. Mögliche Optionen sind:

- Zeitintervall: Minimum innerhalb des Zeitintervall (Sekunden, Minuten, Stunden, Tage)
- Ereignisintervall: Minimum wird gebildet, wenn eine Änderung einer Ereignisvariablen auftritt

Weitere Parameter der Projektierung sind:

- Ausgabe: Ergebnis wird am Ende des Intervalls oder ständig bereitgestellt
- Initialisierung: am Anfang des Intervalls wird der Wert initialisiert (bestehender oder neuer Wert)

Wird die Schaltfläche OK betätigt, öffnet sich ein Dialog, in dem die Parameter der Minimumermittlung eingegeben werden:



Variable: Anzeige der Ereignisvariablen

INTERVALLDAUER

Option	Beschreibung
zeitlich begrenzt	Minimum über die Zeit.

Option	Beschreibung
Sekunden, Minuten, Stunden, Tage	Auswahl des Zeitintervalls.
s	Eingabe des Zahlenwertes für entsprechende Zeit (echtzeitsynchron, d.h. Eingabe 15 bedeutet 15, 30, 45, 0 etc.).
Offset	Verzögerung für Start und Ende der Minimumbildung.
ereignisgesteuert	Minimum wird bei Änderung einer Ereignisvariablen gebildet.

AUSGABE DES ERGEBNIS

Option	Beschreibung
am Ende des Intervalls	Wert wird erst am Ende des Intervalls bereitgestellt, in der Zwischenzeit steht der vorherige Wert zur Verfügung.
laufend	Wert wird bei jeder Abtastung der Quellvariablen aktualisiert.

INITIALISIERUNG ZU BEGINN DES INTERVALLS

Option	Beschreibung
mit bestehendem Wert	Wert basiert auf dem vorherigen Wert.
mit neuem Wert	Wert muss neu berechnet werden.

Info

Der Synchronzeitpunkt wird abhängig vom Zeitintervall ermittelt.

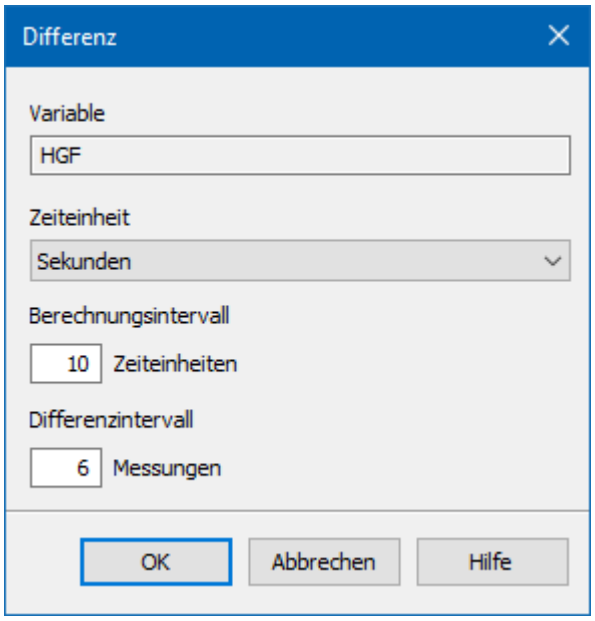
Beispiel: Wird als Zeitintervall 15 Minuten vorgegeben, so wird die Berechnung alle 15 Minuten, ausgehend von der vollen Stunde, durchgeführt.

7.1.4.6.5 Differenz über Zeit

Die Funktion kann z. B. für das dynamische Beobachten des Verbrauchs von Flüssigkeiten verwendet werden oder zur Lecküberwachung von Behältern (auf der Basis von Niveaumesswerten).

Die in die Funktion eingetragene Variable wird beobachtet und ihre Werte werden intern mit dem einstellbaren Messabstand (in Zeiteinheiten) gespeichert. Beim Erreichen eines Differenzabstands (Anzahl der Messungen) stellt die Funktion das erste Resultat (R(n) - R(0)) dar, das ab diesem Zeitpunkt nach jeder Messung aktualisiert wird. Das Resultat der Funktion liegt im Bereich der Realen Zahlen.

Wird die Schaltfläche **OK** betätigt, öffnet sich ein Dialog, in dem die Parameter der Differenzermittlungeingegeben werden:



Option	Beschreibung
Variable	Anzeige der Ereignisvariablen.
Zeiteinheit	Dropdownliste für Wahl der Zeitheinheit (Sekunden, Minuten, Stunden, Tage).
Berechnungsintervall	
Zeiteinheiten	Anzahl der Einheiten für das Berechnungsintervall.
Differenzintervall	
Messungen	Anzahl der Messungen für das Differenzintervall.

Info

Die Zeitberechung der Differenz über Zeit erfolgt nicht absolut, sondern relativ - also bei Runtimestart.

7.1.4.7 Vergleich

Wählen Sie im Bereich Formel Definition in der Dropdownliste **Vergleich** aus. Folgende Operatoren stehen Ihnen zur Verfügung:

Ausdruck	Funktion
<	kleiner als
>	größer als
<=	kleiner gleich
>=	größer gleich
<>	kleiner oder größer
=	gleich
()	Klammern

Vergleichsoperatoren können kombiniert werden mit:

- Arithmetik (auf Seite 42)
- Trigonometrie (auf Seite 43)
- Boolesche Algebra (auf Seite 43)

7.1.4.8 Zählwertverarbeitung

Für die Verknüpfungen mit Zeit- oder Ereignisbezug (Statistik, Datenreduktion) sind zusätzliche Eingaben notwendig. Diese sind beim Beenden der Verknüpfung anzugeben.

Achtung

Diese Funktion steht nur für SICAM 230 zur Verfügung und benötigt eine entsprechende Lizenz. Weitere Informationen finden Sie in der Hilfe zu SICAM 230.

7.1.4.9 Zusätzliche Floatformeln

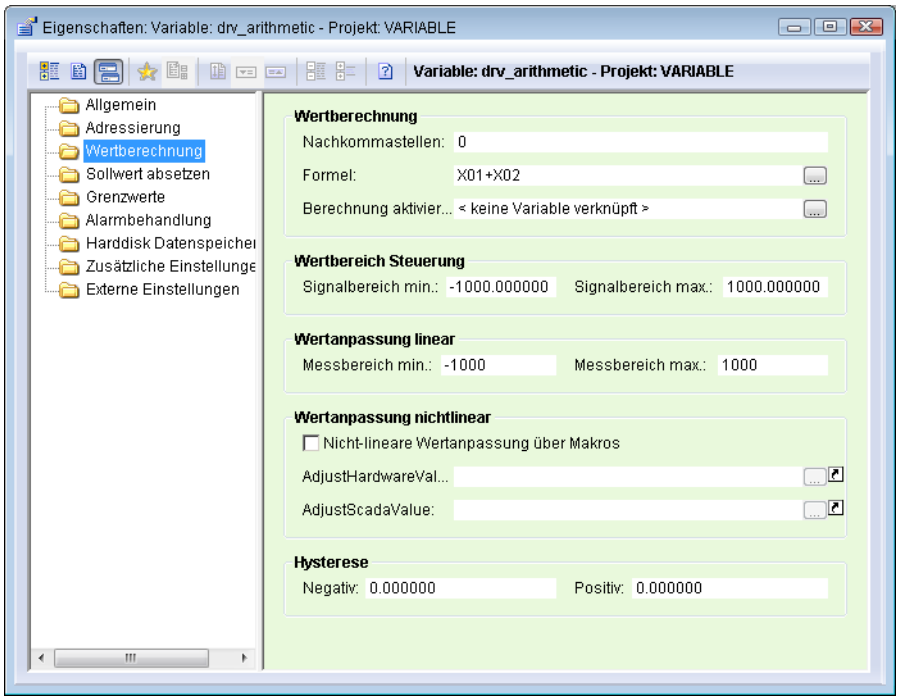
Es stehen Ihnen noch weitere Floatformeln zur Verfügung, die nicht über Schaltflächen erreicht werden können. Diese müssen direkt in den Bereich **Formeln** eingetippt werden:

Syntax	Beschreibung	Notation
ABS(X01)	Absolutwert	|X01|
EXP(X01)	Exponent	$e^{(X01)}$
LN(X01)	Logarithmus naturalis	ln(X01)
LOG(X01)	Logarithmus	log(X01)
PI*(X01)	Konstante Pi	Pi*(X01)
SQR(X01)	Quadratberechnung	$(X01)^2$
SQRT(X01)	Quadratwurzelberechnung	$(X01)^{1/2}$

7.2 Ereignisgesteuertes Berechnen der Mathematik-Variablen

Einer Mathematik Variablen kann eine Variable zugeordnet werden, deren Wert steuert, ob die Berechnung ausgeführt wird. Um eine Variable zur Aktivierung der Berechnung zuzuweisen:

1. navigieren Sie in den Eigenschaften zum Knoten **Wertberechnung**
2. klicken Sie bei der Eigenschaft **Berechnung aktivieren** in das Feld mit dem Variablennamen oder auf die Schaltfläche **...**
3. der Dialog zum Zuweisen einer Variablen öffnet sich
4. wählen Sie die gewünschte Variable



FUNKTIONSWEISE

- Mathematik Variablen ohne Aktivierungsvariable melden alle Quellvariablen, Intervallvariablen und Abtastvariablen beim Start der Runtime an.
- Beginn der Berechnung:
  Ist eine Mathematik Variable mit einer Aktivierungsvariablen verknüpft, erfolgt der Beginn der Berechnung, sobald die Aktivierungsvariable einen Wert *<> 0* hat.
  - Wechselt die Aktivierungsvariable auf *aktiviert*, werden alle zur Berechnung dienenden Quellvariablen, Intervallvariablen und Abtastvariablen angefordert.
  - Wechselt die Aktivierungsvariable auf *deaktiviert*, werden diese wieder abgemeldet.
- Die Aktivierung hat Priorität gegenüber der Berechnung.

  Das zeigt sich, wenn die Aktivierungsvariable als Quellvariable für die Mathematik Variable dient. Wechselt die Aktivierungsvariable auf deaktiviert, bleibt das Ergebnis auf dem gleichen Wert wie zu dem Zeitpunkt, als die Berechnung aktiviert war.
- Datentyp *String* ist möglich, sofern sich der Wert in eine Zahl konvertieren lässt.

  Schlägt die Konvertierung fehl (leer, keine Zahl) entspricht das dem Zustand *deaktiviert*.
- Status der Mathematik Variablen:
  - *deaktiviert: S_AUS* (Bit 20, abgeschaltet).
    Als Wert wird der aktuelle Wert der Mathematik Variablen verwendet. Ist noch kein Wert vorhanden, wird der bei der Mathematik Variablen projektierte Ersatzwert verwendet.

Mathematik Variablen behalten den Status *S_AUS* bis zum Vorliegen eines Berechnungsergebnisses, z. B. Berechnungsart **Zähler -> Schaltspielzähler mit Zustand**.

- bei Berechnungsart *Min* und *Max*, mit Intervallvariable, aber ohne Aktivierungsvariable, bis zum Erhalt der Flanke der Intervallvariablen: *S_IBIT* (Bit 18, gestört)
- bei Berechnungsart *Min* und *Max*, mit Intervallvariable und Aktivierungsvariable bis zum Erhalt der Flanke der Intervallvariablen: *S_AUS*

- Bei Mathematik Variablen, die eine Intervallvariable oder Abtastvariable haben, erfolgt die Flankenerkennung nur, wenn die Berechnung aktiviert ist.

  Ist die Intervallvariable oder Abtastvariable bereits aktiv, wenn die Aktivierungsvariable von deaktiviert auf aktiviert wechselt, entspricht das einer Flanke bei der Intervallvariablen bzw. Abtastvariablen.
- Wird auf eine Mathematik Variable geschrieben, deren Berechnung deaktiviert ist:
  - Wird der geschriebenen Wert nicht versendet und ist damit nicht sichtbar.
  - Der geschriebene Wert dient jedoch als Ausgangswert (aktueller Wert) der Berechnung, sobald die Berechnung aktiviert wird. **Aber:**
  - Zum Standby Server wird der Schreibbefehl unabhängig von der Aktivierungsvariablen gesendet.

7.2.1 Aktivierung und Deaktivierung der Berechnung

AKTIVIERUNG

Die Berechnung wird aktiviert, wenn bei einer Wertänderung für die Aktivierungsvariable folgende Bedingungen erfüllt sind:

- Die Aktivierungsvariable hat einen Wert
- Der Wert wechselt von *0* auf <> *0* und einer der folgenden Status ist aktiv:
  - *S_GA* Generalabfrage Bit 16
  - *S_SPONTAN* Spontan Bit 17
  - *S_REVISION* Revision Bit 9
  - *S_EW_KENNUNG* Ersatzwert Bit 27

Damit ist es auch möglich, durch umschalten auf Ersatzwert eine Aktivierung der Berechnung auszulösen.

DEAKTIVIERUNG

Die Berechnung wird deaktiviert, denn bei der Wertänderung der Aktivierungsvariablen folgende Bedingungen erfüllt sind:

- Die Variable hat noch keinen gültigen Wert.
- Der Wert ist *0* oder keiner der Status *S_GA*, *S_SPONTAN*, *S_REVISION* oder *S_EW_KENNUN*G ist aktiv.

7.2.2 Berechnungsarten mit besonderem Verhalten

ZÄHLER: ZÄHLEN MIT ZUSTAND

Wenn die Berechnung bei **Zählen mit Zustand** deaktiviert ist, wird weder das Kommen noch das Gehen des Zustandes erkannt.

Das heißt: Steht die Aktivierungsvariable auf *deaktiviert*, gilt der Zustand als *nicht verletzt*.

Zum Beispiel:

1. Zustand kommt
2. Aktivierungsvariable wechselt auf *aktiviert*
3. **Verletzung wird erkannt -> Zähler bekommt Wert und zählt**
4. Aktivierungsvariable auf *deaktiviert*
5. Zustand geht
6. Zustand kommt
7. **Aktivierungsvariable auf aktiviert -> es wird nicht gezählt**
8. Zustand geht
9. **Zustand kommt -> es wird gezählt**
10. Aktivierungsvariable auf *deaktiviert*
11. Zustand geht
12. Aktivierungsvariable auf *aktiviert*
13. **Zustand kommt -> es wird gezählt**

7.2.3 Fehlermeldungen

Fehlermeldung	Ursache	Lösung
Source variable <*VariablenProjekt Name*>/<*Variablen-ID*> for mathematics missing. Mathematics variable <*Projektname*>/<*MaV Name*> disabled!	Die Quellvariable des geannten Projekts fehlt. Die Mathematik Variable wird nicht berechnet.	Projektierung oder Startprojekt ändern.
Advise for source variable <*VariablenProjekt Name*>/<*Variable Name*> failed. Mathematics variable <*Projektname*>/<*MaV Name*> disabled	Datenpunktanforderung für die Variable des genannten Projekts ist fehlgeschlagen. Die Mathematik Variable wird nicht berechnet.	Systemressourcen erhöhen, Runtime neu starten.

7.3 Import von Mathematik-Variablen

Beim Importieren (XML Import) von Mathematikvariablen muss zuvor sichergestellt sein, dass alle Variablen die in den Formeln verwendet werden auch vorhanden sind. Werden Variablen verschiedener Treiber importiert, ist es daher empfehlenswert, den Import ein zweites Mal durchzuführen. Damit ist sicher gestellt, dass auch gerade importierte Variablen in den Mathematiktreiber-Formeln verbunden sind.

Achtung

Netzwerk: Auf einem als Datenserver definierten Standbyserver werden Mathematikvariablen nicht angezeigt, wenn dieser zum Server hochgestuft ist.

8 Funktion Treiberkommandos

Die zenon Funktion **Treiberkommandos** dient dazu, Treiber über zenon zu beeinflussen.
Mit einem Treiberkommando können Sie einen Treiber:

- starten
- stoppen
- in einen bestimmten Treibermodus versetzen
- zu bestimmten Aktionen veranlassen

Hinweis: Dieses Kapitel beschreibt Standardfunktionalitäten, die für die meisten zenon Treiber gültig sind.
Nicht alle hier beschriebenen Funktionalitäten stehen für jeden Treiber zur Verfügung. Zum Beispiel enthält ein Treiber, der laut Datenblatt keine Modemverbindung unterstützt, auch keine Modem-Funktionalitäten.

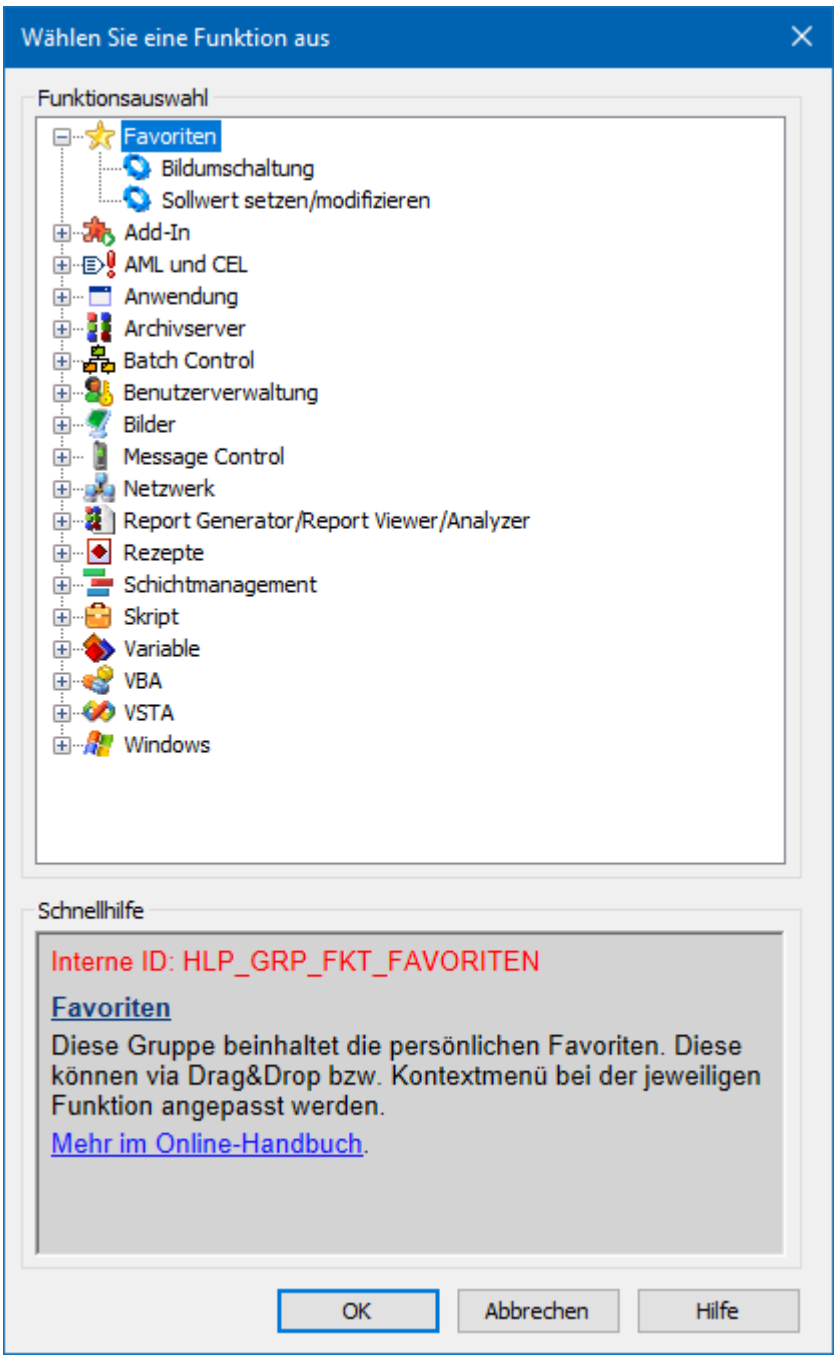
> **Achtung**
>
> Die zenon Funktion **Treiberkommandos** ist nicht ident mit den Treiberkommandos, die bei Energy-Treibern in der Runtime ausgeführt werden können!

PROJEKTIERUNG DER FUNKTION

Die Projektierung erfolgt über die Funktion **Treiberkommandos**.
Um die Funktion zu projektieren:

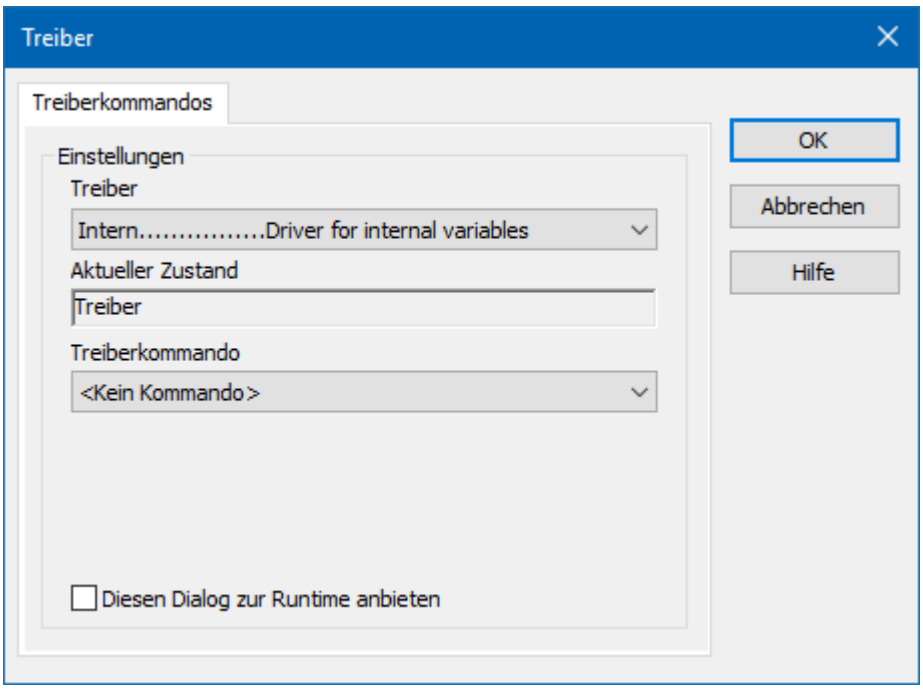
1. Legen Sie im zenon Editor eine neue Funktion an.

Der Dialog zur Auswahl einer Funktion wird geöffnet.



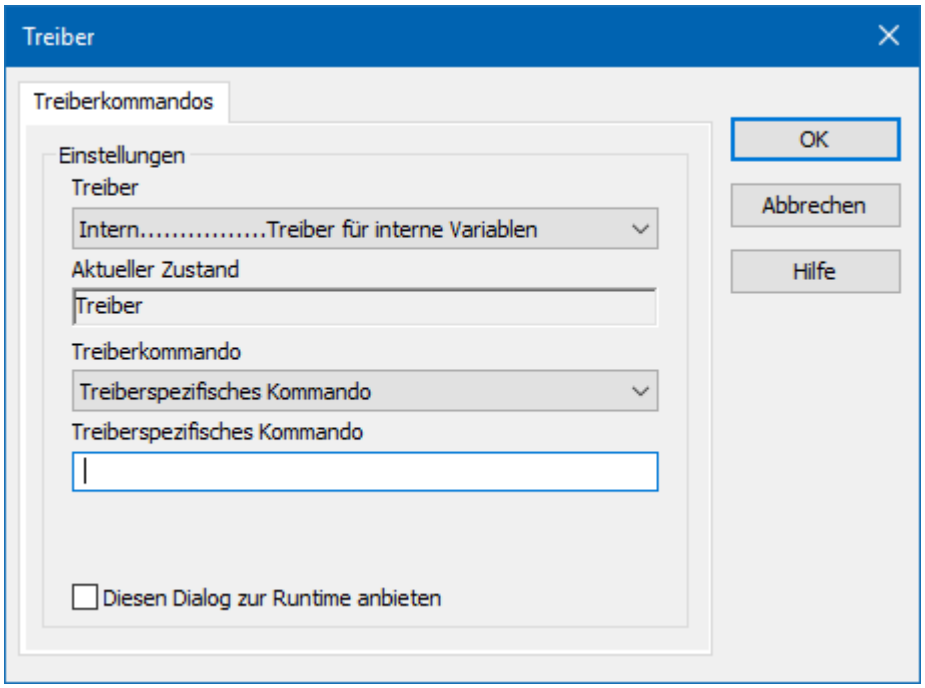
2. Navigieren Sie zum Knoten **Variable.**
3. Wählen Sie den Eintrag **Treiberkommandos**.

Der Dialog zur Konfiguration wird geöffnet.



4. Wählen Sie den gewünschten Treiber und das benötigte Kommando aus.
5. Schließen Sie den Dialog mit Klick auf **OK** und stellen Sie sicher, dass die Funktion in der Runtime ausgeführt wird.
   Beachten Sie die Hinweise im Abschnitt **Funktion Treiberkommandos im Netzwerk**.

DIALOG TREIBERKOMMANDOS



Option	Beschreibung
Treiber	Auswahl des Treibers aus der Dropdownliste. Diese enthält alle im Projekt geladenen Treibern.
Aktueller Zustand	Fixer Eintrag, der vom System gesetzt wird. In aktuellen Versionen ohne Funktion.
Treiberkommando	Auswahl des gewünschten Treiberkommandos aus Dropdownliste. Details zu den konfigurierbaren Treiberkommandos siehe Abschnitt **Verfügbare Treiberkommandos**.
Treiberspezifisches Kommando	Eingabe eines für den gewählten Treiber spezifischen Kommandos. **Hinweis:** Nur verfügbar, wenn für die Option **Treiberkommando** der Eintrag *Treiberspezifisches Kommando* gewählt wurde.
Diesen Dialog zur Runtime anbieten	Konfiguration, ob die Konfiguration in der Runtime geändert werden kann: ▸ *Aktiv*: Dieser Dialog wird in der Runtime vor dem Ausführen der Funktion geöffnet. Die Konfiguration kann damit in der Runtime vor der Ausführung noch geändert werden. ▸ *Inaktiv*: Die Editor-Konfiguration wird in der Runtime beim Ausführen der Funktion angewendet. Default: *inaktiv*

DIALOG BEENDEN

Option	Beschreibung
OK	Übernimmt Einstellungen und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

VERFÜGBARE TREIBERKOMMANDOS

Diese Treiberkommandos stehen - abhängig vom gewählten Treiber - zur Verfügung:

Treiberkommando	Beschreibung
Kein Kommando	Es wird kein Kommando gesendet. Damit kann auch ein bereits bestehendes Kommando aus einer projektierten Funktion entfernt werden.
Treiber starten (Onlinemodus)	Treiber wird neu initialisiert und gestartet. **Hinweis:** Wenn der Treiber bereits gestartet wurde, muss er gestoppt werden. Erst dann kann der Treiber wieder neu initialisiert und gestartet werden.
Treiber stoppen (Offlinemodus)	Treiber wird angehalten, es werden keine neuen Daten angenommen. **Hinweis:** Ist der Treiber im Offline-Modus, erhalten alle Variablen, die für diesem Treiber angelegt wurden, den Status *Abgeschaltet* (*OFF*; Bit *20*).
Treiber in Simulationsmodus	Treiber wird in den Simulationsmodus gesetzt. Die Werte aller Variablen des Treibers werden vom Treiber simuliert. Es werden keine Werte von der angeschlossenen Hardware (z. B. SPS, Bussystem, ...) angezeigt.
Treiber in Hardwaremodus	Treiber wird in den Hardwaremodus gesetzt. Für die Variablen des Treibers werden die Werte von der angeschlossenen Hardware (z. B. SPS, Bussystem, ...) angezeigt.
Treiberspezifisches Kommando	Eingabe eines treiberspezifischen Kommandos. Öffnet Eingabefeld für die Eingabe eines Kommandos.
Treiber Sollwertsetzen aktivieren	Sollwert setzen auf Treiber ist möglich.
Treiber Sollwertsetzen deaktivieren	Sollwert setzen auf Treiber wird verhindert.
Verbindung mit Modem aufbauen	Verbindung aufbauen (für Modem-Treiber). Öffnet Eingabefelder für Hardware-Adresse und Eingabe der zu wählenden Nummer.
Verbindung mit Modem trennen	Verbindung beenden (für Modem-Treiber).
Treiber in Simulationsmodus zählend	Treiber wird in den zählenden Simulationsmodus gesetzt. Alle Werte werden mit *0* initialisiert und in der eingestellten Updatezeit jeweils um *1* bis zum Maximalwert inkrementiert und beginnen dann wieder

Treiberkommando	Beschreibung
	bei *0*.
Treiber in Simulationsmodus statisch	Es wird keine Kommunikation zur Steuerung aufgebaut. Alle Werte werden mit *0* initialisiert.
Treiber in Simulationsmodus programmiert	Die Werte werden von einem frei programmierbaren Simulationsprojekt berechnet. Das Simulationsprojekt wird mit der zenon Logic Workbench erstellt und läuft in der zenon Logic Runtime ab.

FUNKTION TREIBERKOMMANDOS IM NETZWERK

Wenn sich der Rechner, auf dem die Funktion **Treiberkommandos** ausgeführt wird, im zenon Netzwerk befindet, werden zusätzlich weitere Aktionen ausgeführt:

- Ein spezielles Netzwerkkommando wird vom Rechner zum Server des Projekts gesendet. Dieser führt dann die gewünschte Aktion auf seinem Treiber aus.
- Zusätzlich sendet der Server das gleiche Treiberkommando zum Standby des Projekts. Der Standby führt die Aktion auch auf seinem Treiber aus.

Dadurch ist gewährleistet, dass Server und Standby synchronisiert sind. Dies funktioniert nur, wenn Server und Standby jeweils eine funktionierende und unabhängige Verbindung zur Hardware haben.

9 Fehleranalyse

Sollte es zu Kommunikationsproblemen kommen, bietet dieses Kapitel Hilfe, um den Fehler zu finden.

9.1 Analysetool

Alle zenon Module wie z. B. Editor, Runtime, Treiber, usw. schreiben Meldungen in eine gemeinsame LOG-Datei. Um sie korrekt und übersichtlich anzuzeigen, benutzen Sie das Programm Diagnosis Viewer, das mit zenon mitinstalliert wird. Sie finden es unter **Start/Alle Programme/zenon/Tools 8.20 -> Diagviewer.**

zenon Treiber protokollieren alle Fehler in LOG-Dateien. LOG-Dateien sind Textdateien mit einer speziellen Struktur. Der Standardordner für die LOG-Dateien ist der Ordner **LOG** unterhalb des Ordners **ProgramData**, zum Beispiel:

%ProgramData%\COPA-DATA\LOG.

Achtung: Mit den Standardeinstellungen zeichnet ein Treiber nur Fehlerinformationen auf. Mit dem Diagnosis Viewer kann bei den meisten Treibern die Diagnose-Ebene auf „Debug" und „Deep Debug"

erweitert werden. Damit protokolliert der Treiber auch alle anderen wesentlichen Aufgaben und Ereignisse.

Im Diagnosis Viewer kann man auch:

- neu erstellte Einträge in Echtzeit mitverfolgen
- die Aufzeichnungseinstellungen anpassen
- den Ordner, in dem die LOG-Dateien gespeichert werden, ändern

Hinweise:

1. Der Diagnosis Viewer zeigt alle Einträge in UTC (Koordinierter Weltzeit) an und nicht in der lokalen Zeit.
2. Der Diagnosis Viewer zeigt in seiner Standardeinstellung nicht alle Spalten einer LOG-Datei an. Um mehr Spalten anzuzeigen, aktivieren Sie die Eigenschaft **Add all columns with entry** im Kontextmenü der Spaltentitel.
3. Bei Verwendung von reinem **Error-Logging** befindet sich eine Problembeschreibung in der Spalte **Error text**. In anderen Diagnose-Ebenen befindet sich diese Beschreibung in der Spalte **General text**.
4. Viele Treiber zeichnen bei Kommunikationsprobleme auch Fehlernummern auf, die die SPS ihnen zuweist. Diese werden in **Error text** und/oder **Error code** und/oder **Driver error parameter(1** und **2)** angezeigt. Hinweise zur Bedeutung der Fehlercodes erhalten Sie in der Treiberdokumentation und der Protokoll/SPS-Beschreibung.
5. Stellen Sie am Ende Ihrer Tests den Diagnose-Level von **Debug** oder **Deep Debug** wieder zurück. Bei **Debug** und **Deep Debug** fallen beim Protokollieren sehr viele Daten an, die auf der Festplatte gespeichert werden und die Leistung Ihres Systems beeinflussen können. Diese werden auch nach dem Schließen des Diagnosis Viewers weiter aufgezeichnet.

> **Achtung**
>
> Unter Windows CE werden aus Ressourcegründen Fehler standardmäßig nicht protokolliert.

Weitere Informationen zum Diagnosis Viewer finden Sie im Handbuch Diagnosis Viewer.

9.2 Treiberüberwachung

Die Runtime überwacht die Verfügbarkeit des Treibers via Watchdog. Ist ein Treiber nicht mehr verfügbar, wird für alle angemeldeten Variablen des Treibers zusätzlich das Statusbit *INVALID* gesetzt.

Mögliche Ursachen für Auslösen des Watchdogs:

- Der Treiberprozess läuft nicht mehr.

  Überprüfen Sie im Task Manager ob die Treiber-Exe noch läuft.

- Betriebssystem ist mit höher priorisierten Prozessen ausgelastet.

  Überprüfen Sie die Konfiguration Ihres Systems auf zu wenig Arbeitsspeicher und zu geringe CPU-Leistung. In diesem Fall erfolgt das Rücksetzen des *INVALID* Statusbits durch den Treiber nur bei Wertänderung auf der Gegenstelle. Statische Werte behalten das *INVALID* Statusbit bis zum nächsten Start der Runtime oder des Treibers.

WATCHDOG KONFIGURATION

Für die Überwachung der Kommunikation zur Runtime wird die Verbindung zum Treiber in einem fix vorgegebenen Zeitraum von 60 Sekunden überprüft. Dieser Vorgang wird mehrmals wiederholt. Konnte nach 5 Versuchen (= innerhalb von 5 Minuten) keine valide Verbindung zum Treiber erkannt werden, wird das *INVALID*-Bit bei der angemeldeten (*advised*) Variablen gesetzt. Zusätzlich wird das *INVALID*-Bit auch gesetzt, wenn neue Variablen angemeldeten werden. Das *INVALID*-Bit wird nicht mehr zurückgesetzt.

Dafür werden entsprechende LOG-Einträge erstellt.

LOG-EINTRAG

Bei Auslösen des Watchdogs wird eine Fehlermeldung im LOG protokolliert:

Parameter	Beschreibung
Communication with driver:<drvExe>/<drvDesc>(id:<drvId>) timed out. No communication for <time> ms.	Keine Kommunikation mit Treiber innerhalb der angegeben Zeit. - *<time>*: Zeitangabe in Millisekunden - *<drvDesc>*: Treibername - *<drvExe>*: Treiber EXE-Name - *<drvId>*: Treiber-ID im zenon Projekt
Communication with %s timed out. Invalid-Bit will be set.	Die Kommunikation zum Treiber *%s* konnte bei 5 Versuchen nicht innerhalb von 60 Sekunden aufgebaut werden. Bei der Variable wird das *INVALID*-Bit gesetzt.
Communication with %s timed out. Timeout happened %d times	Die Kommunikation zum Treiber *%s* konnte *%d* Mal nicht innerhalb von 60 Sekunden aufgebaut werden.

9.3 Logserver

Sämtliche Meldungen und Logs des Treibers werden an den LogServer gesendet. Die Meldungen können mit dem Diagnosis Viewer angezeigt werden. (Ältere Versionen von zenon benutzen eine „Error – Textdatei".

Per Default werden für das Modul Treiber (DRV) die Meldungen vom Typ Error mitgeloggt. Sollen erweiterte Logs erstellt werden, ist dies in den Client Einstellungen des Diagnosis Viewers zur Laufzeit entsprechend zu konfigurieren: **Settings – Client configuration, List of parameters, ' Einstellen des Message Levels** ', mit **Accept** übernehmen.

9.4 Checkliste

- Analyse mit Hilfe des Diagnosis Viewers (auf Seite 75):
  -> Welche Meldungen werden angezeigt?