

# zenon Treiber Handbuch

## HOFBUS

v.7.11





©2014 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.

# Inhalt

<b>1. Willkommen bei der COPA-DATA Hilfe .....</b>	<b>4</b>
<b>2. HOFBUS.....</b>	<b>4</b>
<b>3. HOFBUS - Datenblatt .....</b>	<b>5</b>
<b>4. Treiber-Historie .....</b>	<b>6</b>
<b>5. Konfiguration .....</b>	<b>7</b>
5.1 Anlegen eines Treibers.....	7
5.2 Einstellungen im Treiberdialog .....	9
5.2.1 Allgemein.....	9
5.2.2 Com .....	12
5.2.3 Settings .....	15
<b>6. Variablen anlegen.....</b>	<b>15</b>
6.1 Variablen im Editor anlegen.....	15
6.2 Adressierung .....	19
6.3 Treiberobjekte und Datentypen .....	20
6.3.1 Treiberobjekte .....	20
6.3.2 Zuordnung der Datentypen .....	20
6.4 Variablen anlegen durch Import .....	21
6.4.1 XML Import.....	22
6.4.2 DBF Import/Export .....	22
6.5 Treibervariablen .....	29
<b>7. Treiberspezifische Funktionen.....</b>	<b>34</b>
<b>8. Treiberkommandos .....</b>	<b>35</b>
<b>9. Fehleranalyse .....</b>	<b>37</b>
9.1 Analysetool .....	37
9.2 Checkliste .....	38

# 1. Willkommen bei der COPA-DATA Hilfe

## **ALLGEMEINE HILFE**

Falls Sie in diesem Hilfekapitel Informationen vermissen oder Wünsche für Ergänzungen haben, wenden Sie sich bitte per E-Mail an [documentation@copadata.com](mailto:documentation@copadata.com) (<mailto:documentation@copadata.com>).

## **PROJEKTUNTERSTÜTZUNG**

Unterstützung bei Fragen zu konkreten eigenen Projekten erhalten Sie vom Support-Team, das Sie per E-Mail an [support@copadata.com](mailto:support@copadata.com) (<mailto:support@copadata.com>) erreichen.

## **LIZENZEN UND MODULE**

Sollten Sie feststellen, dass Sie weitere Module oder Lizenzen benötigen, sind unsere Mitarbeiter unter [sales@copadata.com](mailto:sales@copadata.com) (<mailto:sales@copadata.com>) gerne für Sie da.

# 2. HOFBUS

Treiber zur Kommunikation mit PCR-1 Prozesssteuerungen über das HOFBUS Protokoll.

### 3. HOFBUS - Datenblatt

Allgemein:	
Treiberdateiname	HOFBUS.exe
Treiberbezeichnung	HOFBUS
Steuerungs-Typen	PCR-1 Prozesssteuerung
Steuerungs-Hersteller	Roche;

Treiber unterstützt:	
Protokoll	HOFBUS;
Adressierung: Adress-basiert	x
Adressierung: Namens-basiert	-
Kommunikation spontan	-
Kommunikation pollend	x
Online Browsing	-
Offline Browsing	-
Echtzeitfähig	-
Blockwrite	-
Modemfähig	-
Seriellles Logging	x
RDA numerisch	-
RDA String	-

<b>Voraussetzungen:</b>	
Hardware PC	Serielle Schnittstelle
Software PC	-
Hardware Steuerung	-
Software Steuerung	-
Benötigt v-dll	-

<b>Plattformen:</b>	
Betriebssysteme	Windows Vista, 7, 8, 8.1 Server 2008/R2, Server 2012/R2;
CE Plattformen	-;

## 4. Treiber-Historie

<b>Datum</b>	<b>Build Nummer</b>	<b>Änderung</b>
17.06.13	6364	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, Servicepack 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 versionsunabhängig*

## 5. Konfiguration

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



### Info

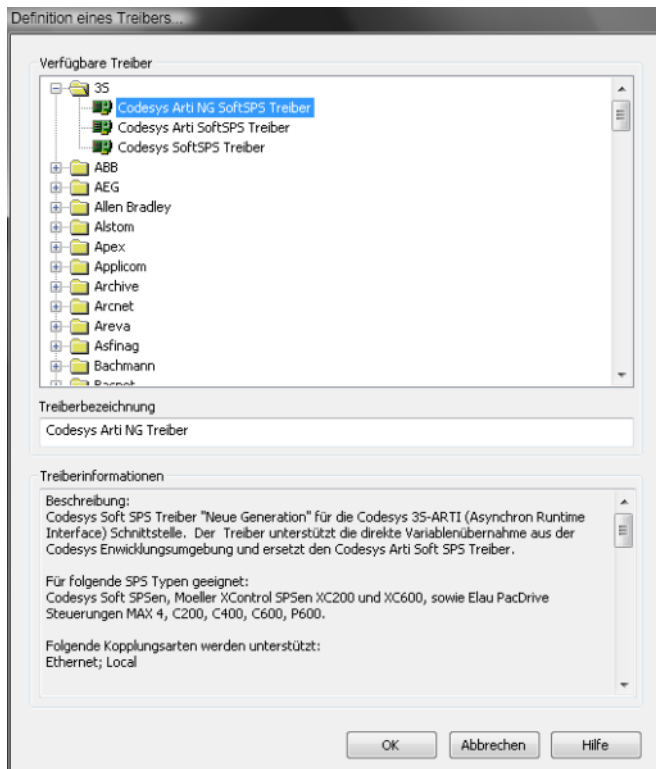
*Weitere Einstellungen, die Sie für Variablen in zenon vornehmen können, finden Sie im Kapitel Variablen (<main.chm::/15247.htm>) der Online-Hilfe.*

### 5.1 Anlegen eines Treibers

Um einen neuen Treiber anzulegen:

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

2. In der folgenden Dialogbox bietet Ihnen das Programm eine Auflistung aller verfügbaren Treiber an.



3. Selektieren Sie den gewünschten Treiber und vergeben Sie eine Bezeichnung für diesen:
  - Die Treiberbezeichnung muss eindeutig sein, d.h. wird ein und derselbe Treiber mehrmals im Projekt verwendet, so muss jeweils eine neue Bezeichnung vergeben werden.
  - 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 **OK**. Im folgenden Dialog werden die einzelnen Konfigurationen der jeweiligen Treiber eingestellt.

Für ein Projekt müssen nur die jeweils notwendigen Treiber eingebunden werden. Späteres Einbinden eines weiteren Treibers ist problemlos möglich.





### Info

*Bei neu angelegten Projekten oder bei bestehenden Projekten, die in eine Version ab 6.21 konvertiert werden, werden die folgenden Treiber automatisch angelegt:*

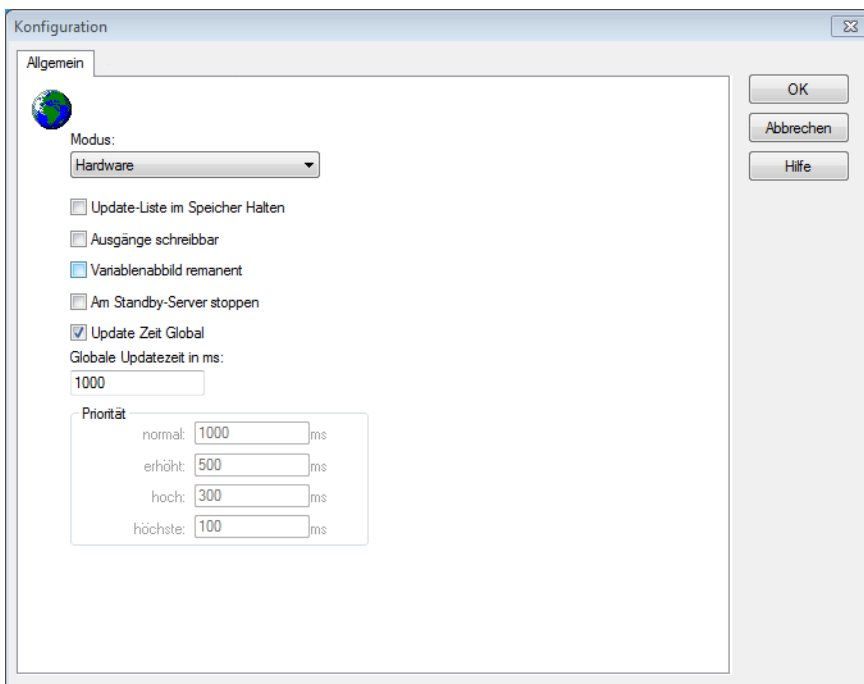
- ▶ Intern
- ▶ MathDr32
- ▶ SysDrv.

▶

## 5.2 Einstellungen im Treiberdialog


Folgende Einstellungen können Sie beim Treiber vornehmen:

### 5.2.1 Allgemein



Konfiguration

Allgemein

 Modus:  
Hardware

☐ Update-Liste im Speicher Halten  
☐ Ausgänge schreibbar  
☒ Variablenabbild remanent  
☐ Am Standby-Server stoppen  
☒ Update Zeit Global  
Globale Updatezeit in ms:  
1000

Priorität:

normal:	1000	ms
erhöht:	500	ms
hoch:	300	ms
höchste:	100	ms

OK  
Abbrechen  
Hilfe

Parameter	Beschreibung
Modus	<p>Ermöglicht ein Umschalten zwischen Hardware und Simulationsmodus</p> <ul style="list-style-type: none"> <li>▶ Hardware:  Die Verbindung zur Steuerung wird hergestellt.</li> <li>▶ Simulation - statisch  Es wird keine Kommunikation zur Steuerung aufgebaut, die Werte werden vom Treiber simuliert. In diesem Modus bleiben die Werte konstant bzw. die Variablen behalten die über zenon Logic gesetzten Werte. Jede Variable hat seinen eigenen Speicherbereich. Zum Beispiel zwei Variablen vom Typ Merker mit Offset 79, können zur Runtime unterschiedliche Werte haben und beeinflussen sich gegenseitig nicht. Ausnahme: Der Simulatortreiber.</li> <li>▶ Simulation - zählend  Es wird keine Kommunikation zur Steuerung aufgebaut, die Werte werden vom Treiber simuliert. In diesem Modus zählt der Treiber die Werte innerhalb ihres Wertebereichs automatisch hoch.</li> <li>▶ Simulation - programmiert  Es wird keine Kommunikation zur Steuerung aufgebaut, die Werte werden von einem frei programmierbaren Simulationsprojekt berechnet. Das Simulationsprojekt wird mit der zenon Logic Workbench erstellt und läuft in einer in den Treiber integrierten zenon Logic Runtime ab. Details siehe Kapitel Treibersimulation. (main.chm::/25206.htm)</li> </ul>
Update-Liste im Speicher Halten	<p>Einmal angeforderte Variablen werden weiterhin von der Steuerung angefordert, auch wenn diese aktuell nicht mehr benötigt werden. Dies hat den Vorteil, dass z. B. mehrmalige Bildumschaltungen nach dem erstmaligen Aufschalten beschleunigt werden, da die Variablen nicht neu angefordert werden müssen. Der Nachteil ist eine erhöhte Belastung der Kommunikation zur Steuerung.</p>
Ausgänge schreibbar	<p>Aktiv: Ausgänge können beschrieben werden.</p> <p>Inaktiv: Das Beschreiben der Ausgänge wird unterbunden.</p>

	<p><b>Hinweis:</b> Steht nicht für jeden Treiber zur Verfügung.</p>
Variablenabbild remanent	<p>Diese Option speichert und restauriert den aktuellen Wert, den Zeitstempel und die Status eines Datenpunkts.</p> <p>Grundvoraussetzung: Die Variable muss einen gültigen Wert und Zeitstempel besitzen.</p> <p>Das Variablenabbild wird im Modus Hardware gespeichert wenn:</p> <ul style="list-style-type: none"> <li>▶ einer der Status S_MERKER_1(0) bis S_MERKER8(7), REVISION(9), AUS(20) oder ERSATZWERT(27) aktiv ist</li> </ul> <p>Das Variablenabbild wird immer gespeichert wenn:</p> <ul style="list-style-type: none"> <li>▶ die Variable vom Objekttyp <code>Treibervariable</code> ist</li> <li>▶ der Treiber im Simulationsmodus läuft. (nicht programmierte Simulation)</li> </ul> <p>Folgende Status werden beim Start der Runtime nicht restauriert:</p> <ul style="list-style-type: none"> <li>▶ SELECT(8)</li> <li>▶ WR-ACK(40)</li> <li>▶ WR-SUC(41)</li> </ul> <p>Der Modus <b>Simulation</b> - <b>programmiert</b> beim Treiberstart ist kein Kriterium, um das remanente Variablenabbild zu restaurieren.</p>
Am Standby Server stoppen	<p>Einstellung für Redundanz bei Treibern, die nur eine Kommunikationsverbindung erlauben. Dazu wird der Treiber am Standby Server gestoppt und erst beim Hochstufen wieder gestartet.</p> <p><b>Achtung:</b> Ist diese Option aktiv, ist die lückenlose Archivierung nicht mehr gewährleistet.</p> <p>Aktiv: Versetzt den Treiber am nicht-prozessführenden Server automatisch in einen Stopp-ähnlichen Zustand. Im Unterschied zum Stoppen über Treiberkommando erhält die Variable nicht den Status <b>abgeschaltet</b> (<code>statusverarbeitung.chm: /24150.htm</code>), sondern einen leeren Wert. Damit wird verhindert, dass beim Hochstufen zum Server nicht relevante Werte in AML, CEL und Archiv erzeugt werden.</p>
Update Zeit Global	<p>Aktiv: Die eingestellte Globale Update Zeit in ms wird für alle Variablen im Projekt verwendet. Die bei den Variablen eingestellte Priorität wird nicht verwendet.</p>

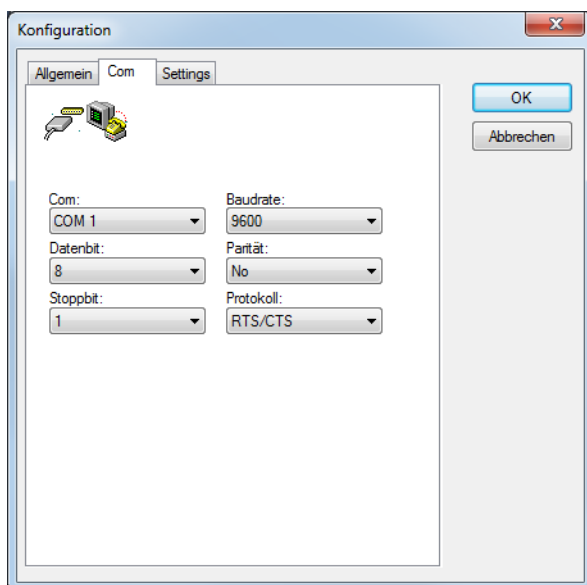
	Inaktiv: Die eingestellten Prioritäten werden für die einzelnen Variablen verwendet.
Priorität	Hier werden die Pollingzeiten der einzelnen Prioritäten eingestellt. Alle Variablen mit der entsprechenden Priorität werden in der eingestellten Zeit gepollt. Die Zuordnung zu den Variablen erfolgt separat bei jeder Variablen über die Einstellungen in den Variableneigenschaften. Mit den Prioritäten kann die Kommunikation der einzelnen Variablen auf die Wichtigkeit bzw. benötigte Aktualität abgestuft werden. Daraus ergibt sich eine verbesserte Verteilung der Kommunikationslast.
OK	Übernimmt Einstellungen aus allen Registerkarten und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

## UPDATE ZEIT ZYKLISCHE TREIBER

Für zyklische Treiber gilt:

Beim **Sollwert Setzen**, **Advisen** von Variablen und bei **Requests** wird sofort ein Lesezyklus für alle Treiber ausgelöst - unabhängig von der eingestellten Update Zeit. Damit wird sicher gestellt, dass der Wert nach dem Schreiben in der Visualisierung sofort zur Verfügung steht. Update-Zeiten können damit für zyklische Treiber kürzer ausfallen als eingestellt.

### 5.2.2 Com



Parameter	Beschreibung
Com	Auswahl der serielle Schnittstelle aus Dropdownliste: COM1 – COM64
Datenbit	Auswahl der Datenwortlänge in Bit aus Dropdownliste: <ul style="list-style-type: none"><li>▸ 5</li><li>▸ 6</li><li>▸ 7</li><li>▸ 8</li></ul>
Stopbit	Auswahl des Stopbits aus Dropdownliste: <ul style="list-style-type: none"><li>▸ 1,</li><li>▸ 1.5</li><li>▸ 2</li></ul>
Baudrate	Auswahl der Datenübertragungsrate aus Dropdownliste: <ul style="list-style-type: none"><li>▸ 110</li><li>▸ 300</li><li>▸ 1200</li><li>▸ 2400</li><li>▸ 4800</li><li>▸ 9600</li><li>▸ 19200</li><li>▸ 38400</li><li>▸ 57600</li><li>▸ 115200</li></ul>
Parität	Auswahl der Parität aus Dropdownliste: <ul style="list-style-type: none"><li>▸ No - keine</li></ul>

	<ul style="list-style-type: none"> <li>▸ Odd - ungerade</li> <li>▸ Even - gerade</li> </ul>
Protokoll	Auswahl des Protokolls aus Dropdownliste: <ul style="list-style-type: none"> <li>▸ RTS/CTS/</li> <li>▸ kein</li> </ul>
<b>OK</b>	Übernimmt alle Änderungen in allen Registerkarten und schließt den Dialog.
<b>Abbrechen</b>	Verwirft alle Änderungen in allen Registerkarten und schließt den Dialog.
<b>Hilfe</b>	Öffnet die Online-Hilfe.

#### DEFAULT KONFIGURATION

Parameter	Beschreibung
Com	COM1
Baudrate	Anpassen nach Einstellen auf Steuerung
Datenbit	8
Stoppbit	1
Parität	Anpassen nach Einstellung auf Steuerung
Protokoll	NO

### 5.2.3 Settings

Parameter	Beschreibung
Timeout	Wartezeit auf Kommunikation in Millisekunden. Default: 20000
Retries	Anzahl der Wiederholungen für den Kommunikationsaufbau. Default: 3
OK	Übernimmt alle Änderungen in allen Registerkarten und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen in allen Registerkarten und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

## 6. Variablen anlegen

So werden Variablen im zenon Editor angelegt:

### 6.1 Variablen im Editor anlegen

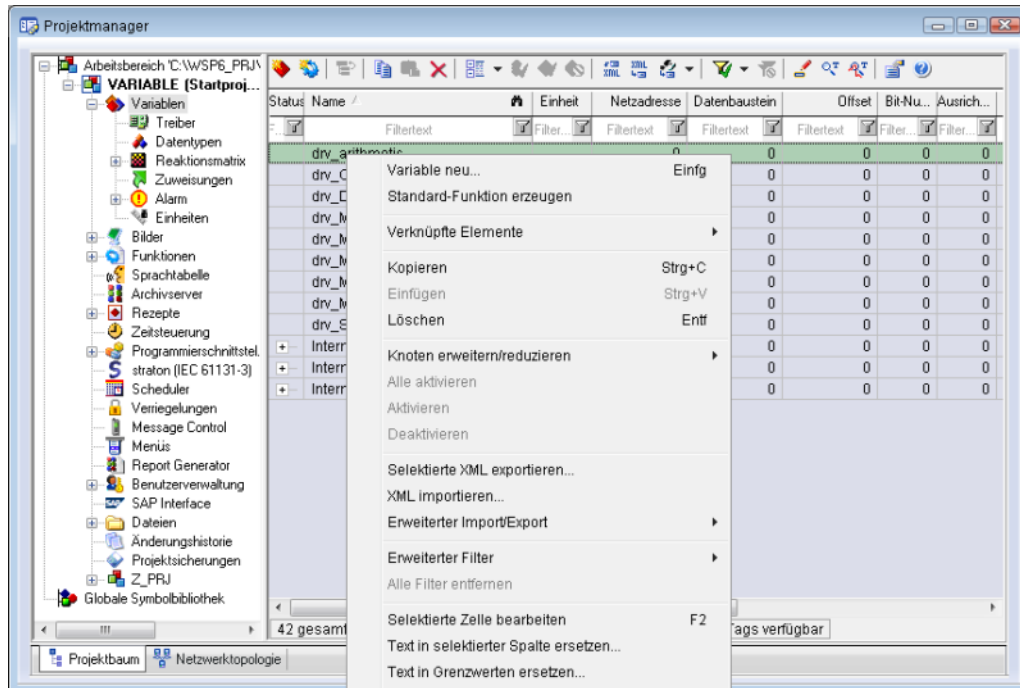
Variablen können angelegt werden:

- ▶ als einfache Variable
- ▶ in Arrays (main.chm::/15262.htm)
- ▶ als Struktur-Variablen (main.chm::/15278.htm)

#### DIALOG VARIABLE

Um eine neue Variable zu erstellen, gleich welchen Typs:

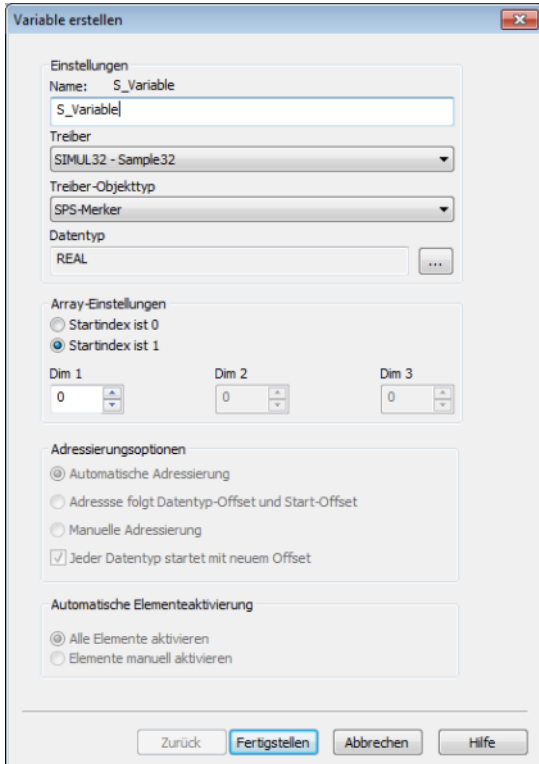
1. wählen Sie im Knoten **variablen** im Kontextmenü den Befehl **variable neu**



2. der Dialog zur Konfiguration der Variable wird geöffnet
3. konfigurieren Sie die Variable



#### 4. welche Einstellungen möglich sind, hängt ab vom Typ der Variablen



Eigenschaft	Beschreibung
Name	<p>Eindeutiger Name der Variablen. Ist eine Variable mit gleichem Namen im Projekt bereits vorhanden, kann keine weitere Variable mit diesem Namen angelegt werden.</p> <p>Maximale Länge: 128 Zeichen</p> <p><b>Achtung:</b> Die Zeichen # und @ sind für Variablennamen nicht erlaubt. Bei Verwendung nicht zugelassener Zeichen kann die Variablenerstellung nicht abgeschlossen werden, die Schaltfläche <b>Fertigstellen</b> bleibt inaktiv.</p>
Treiber	<p>Wählen Sie aus der Dropdownliste den gewünschten Treiber.</p> <p><b>Hinweis:</b> Sollte im Projekt noch kein Treiber angelegt sein, wird automatisch der Treiber für interne Variable (Intern.exe (Main.chm::/Intern.chm::/Intern.htm)) geladen.</p>
Treiber-Objekttyp (cti.chm::/28685.htm)	<p>Wählen Sie aus der Dropdownliste den passenden Treiber-Objekttyp aus.</p>

Datentyp	Wählen Sie den gewünschten Datentyp. Klick auf die Schaltfläche ... 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.

### 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	
Name	<p>Frei vergebbarer Name.</p> <p><b>Achtung:</b> Je zenon Projekt muss der Name eindeutig sein.</p>
Kennung	Frei vergebbare Kennung, z. B. für Betriebsmittelkennung, Kommentar...
Adressierung	
Netzadresse	Adresse der Steuerung auf der sich diese Variable befindet
Datenbaustein	Nicht verwendet
Offset	Offset / Signalnummer der Variablen, die Speicheradresse der Variable in der Steuerung. Wertbereich von 0 bis 65535 .
Ausrichtung	<p>Auswahl des Bytes innerhalb des eingestellten Offsets.</p> <p><b>Achtung:</b> Der Treiber unterstützt kein Read-Modify-Write, Variablen vom Datentyp Byte können daher nicht geschrieben werden</p>
Bit-Nummer	<p>Nummer des Bits innerhalb des eingestellten Offsets.</p> <p>Mögliche Eingabe: 0 ... 65535</p> <p>Ein Wert größer 16 bedeutet, dass die BOOL Variable im ganzen Wort gespeichert wird (0 bzw. 0xFFFF)</p> <p><b>Achtung:</b> Der Treiber unterstützt kein Read-Modify-Write, Wird eine Variable vom Datentyp BOOL geschrieben, wird diese immer in das ganze Wort geschrieben (0 bzw. 0xFFFF).</p>
Stringlänge	Nur verfügbar bei String-Variablen: Maximale Anzahl von Zeichen, die die Variable aufnehmen kann.
Treiber Anbindung/Treiber-Objekttyp	Objekttyp der Variablen. Wird abhängig vom verwendeten Treiber beim Erstellen der Variablen ausgewählt und kann hier geändert werden.
Treiber Anbindung/Datentyp	<p>Datentyp der Variablen. Wird beim Erstellen der Variablen ausgewählt und kann hier geändert werden.</p> <p><b>ACHTUNG:</b> Wenn der Datentyp nachträglich geändert wird, müssen alle anderen Eigenschaften der Variablen überprüft bzw. angepasst werden.</p>

## 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:

Treiberobjekttyp	Kanaltyp	Lesen / Schreiben	Unterstützte Datentypen	Beschreibung
SPS-Merker	8	L/S	BOOL, SINT, USINT, INT, UINT	Variablen in der Steuerung
Treibervariablen	35	L/S	BOOL, SINT, USINT, INT, UINT, DINT, UDINT, REAL, STRING	<p>Variablen für die statische Analyse der Kommunikation; wird zwischen Treiber und Runtime übertragen (nicht zur SPS).</p> <p><b>Hinweis:</b> Die Adressierung und das Verhalten ist bei den meisten zenon Treibern gleich</p> <p>Weitere Infos finden Sie bei den Treibervariablen (auf Seite 29)</p>

### 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.

Steuerung	zenon	Datenart
BOOL	BOOL	8
USINT	USINT	9
SINT	SINT	10
UINT	UINT	2
-	INT	1
-	UDINT	4
-	DINT	3
-	ULINT	27
-	LINT	26
-	REAL	5
-	LREAL	6
-	STRING	12
-	WSTRING	21
-	DATE	18
-	TIME	17
-	DATE_AND_TIME	20
-	TOD (Time of Day)	19

**Datenart:** Die Eigenschaft `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 (main.chm::/13028.htm) im Abschnitt Variablen (main.chm::/13045.htm).*

### 6.4.1 XML Import

Für den Import/Export von Variablen gilt:

- ▶ Der Import/Export darf nicht aus dem Globalprojekt gestartet werden.
- ▶ Der Start erfolgt über:
  - Kontextmenü zu Variablen bzw. Datentyp im Projektbaum
  - oder Kontextmenü einer Variablen bzw. eines Datentyps
  - oder Symbol in der Symbolleiste Variablen

**Achtung**

*Beim Import/Überschreiben von existierenden Datentypen werden alle Variablen geändert, die auf diesem existierenden Datentyp basieren.*

**Beispiel:**

*Es existiert ein Datentyp XYZ abgeleitet vom Typ `INT` mit Variablen, die auf diesem Datentyp basieren. Ihre zu importierende XML-Datei enthält ebenfalls einen Datentyp mit Namen XYZ, allerdings abgeleitet vom Typ `STRING`. Wird dieser Datentyp importiert, so wird der existierende Datentyp überschrieben und bei allen auf ihm basierenden Variablen der Typ angepasst. D.h. die Variablen sind jetzt `STRING`- und keine `INT`-Variablen mehr.*

### 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 dem Importassistenten

Das Format der Datei ist im Kapitel Dateiaufbau beschrieben.

**Info**

*Beachten Sie:*

- ▶ Treiberobjekttyp und Datentyp müssen in der DBF-Datei an den Zieldriver 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 dem 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 <b>project.ini</b> 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 <b>project.ini</b> 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 <b>project.ini</b> 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 Treiber-Objektyp.
HWKANAL	Num	3	Bus-Adresse
BAUSTEIN	N	3	Datenbaustein-Adresse (nur bei Variablen aus den Datenbereich der SPS)
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-OBJEKTYP 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
MENTIEFE	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, 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 <b>project.ini</b> 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 Treiberobjektyp und Datentyp in der DBF-Datei an den Zieldreiber angepasst werden, damit Variablen importiert werden.*

## GRENZWERTDEFINITION

Grenzwertdefinition für Grenzwert 1 bis 4, bzw. 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)
SCHWERT1	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
GRENZTXT1	C	66	Grenzwertext
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 Treibervariablen

Das Treiberkit implementiert eine Reihe von Treibervariablen. Diese sind unterteilt in:

- ▶ Information
- ▶ Konfiguration
- ▶ Statistik und
- ▶ Fehlermeldungen

Die Definitionen der im Treiberkit implementierten Variablen sind in der Importdatei `drvvar.dbf` (auf der CD im Verzeichnis: `CD_Laufwerk:/Predefined/Variables`) verfügbar und können von dort importiert werden.

**Hinweis:** Variablennamen müssen in zenon einzigartig sein. Soll nach einem Import der Treibervariablen aus `drvvar.dbf` ein erneuter Import durchgeführt werden, müssen die zuvor importierten Variablen umbenannt werden.



### Info

*Nicht jeder Treiber unterstützt alle Treibervariablen.*

*Zum Beispiel werden:*

- ▶ Variablen für Modem-Informationen nur von modemfähigen Treibern unterstützt
- ▶ Treibervariablen für den Polling-Zyklus nur für rein pollenden Treibern
- ▶ verbindungsbezogene Informationen wie ErrorMessage nur von Treibern, 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 Servicepack-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.
TreiberStop	BOOL	28	Treiber gestoppt

			Bei <b>Treiberstop</b> , hat die Variable den Wert <b>TRUE</b> und ein <b>OFF</b> -Bit. Nach dem Treiberstart, hat die Variable den Wert <b>FALSE</b> und kein <b>OFF</b> -Bit.
SimulRTState	UDINT	60	Informiert über Status der Runtime bei Treibersimulation.

## KONFIGURATION

Name aus Import	Typ	Offset	Erklärung
ReconnectInRead	BOOL	27	Wenn <b>TRUE</b> , dann wird beim Lesen automatisch ein Neuaufbau der Verbindung durchgeführt.
ApplyCom	BOOL	36	Änderungen an den Einstellungen der seriellen Schnittstelle zuweisen. Das Schreiben auf diese Variable hat unmittelbar den Aufruf der Methode <b>SrvDrvVarApplyCom</b> 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 <b>SrvDrvVarApplyModem</b> zur Folge. Diese schließt die aktuelle Verbindung und öffnet eine neue entsprechend den Einstellungen <b>PhoneNumberSet</b> und <b>ModemHwAdrSet</b> .
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	<b>TRUE</b> , wenn die Updatezeit global ist.
TreiberSimul	BOOL	5	<b>TRUE</b> , 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.
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 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 <b>Normal</b> in Millisekunden (ms).
MaxUpdateTimeHigher	UDINT	57	Zeit seit letzter Aktualisierung der Prioritätsgruppe <b>Höher</b> in Millisekunden (ms).
MaxUpdateTimeHigh	UDINT	58	Zeit seit letzter Aktualisierung der Prioritätsgruppe <b>Hoch</b> in Millisekunden (ms).

MaxUpdateTimeHighest	UDINT	59	Zeit seit letzter Aktualisierung der Prioritätsgruppe <b>Höchste</b> 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.
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

Dieser Treiber unterstützt folgende Funktionen:

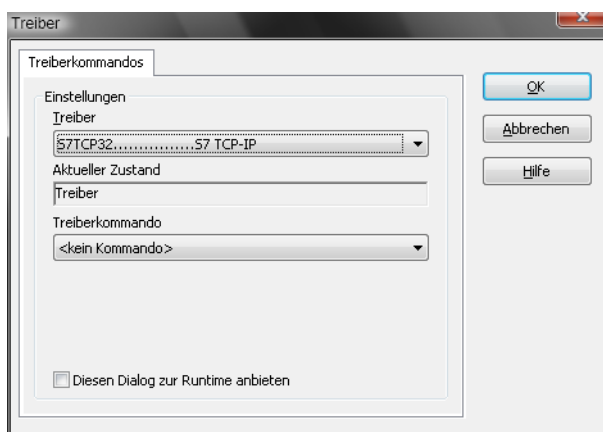
Der Treiber wertet die Bits `Disabled` und `Out-Of-Range` aus und schreibt das jeweilige Bit auf das `INVALID` (main.chm::/24148.htm) Statusbit bzw. `OV_870` (main.chm::/28115.htm) Statusbit der zenon Variable.

## 8. Treiberkommandos

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.

Treiberkommandos dienen dazu, Treiber über zenon zu beeinflussen, z. B. starten und stoppen. Die Projektierung erfolgt über die Funktion **Treiber Kommandos**. Dazu:

- ▶ legen Sie eine neue Funktion an
- ▶ wählen Sie *Variablen -> Treiberkommandos*
- ▶ der Dialog zur Konfiguration wird geöffnet



Parameter	Beschreibung
Treiber	Dropdownliste mit allen im Projekt geladenen Treibern.
Aktueller Zustand	Fixer Eintrag, in aktuellen Versionen ohne Funktion.
<b>Treiberkommando</b>	Dropdownliste zur Auswahl des Kommandos.
▶ Treiber starten (Online-Modus)	Treiber wird neu initialisiert und gestartet.

▶ Treiber stoppen (Offline-Modus)	Treiber wird angehalten, es werden keine neuen Daten angenommen.  <b>Hinweis:</b> 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 treiberspezifischer Kommandos. Öffnet Eingabefeld für die Eingabe eines Kommandos.
▶ Treiber Sollwertsetzen aktivieren	Sollwert setzen auf Treiber ist erlaubt.
▶ 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).
Diesen Dialog zur Runtime anbieten	Dialog wird zur Runtime für Änderungen angeboten.

## TREIBERKOMMANDOS IM NETZWERK

Wenn sich der Rechner, auf dem die Funktion **Treiberkommandos** ausgeführt wird, im zenon Netzwerk befindet, werden zusätzliche Aktionen ausgeführt. Ein spezielles Netzwerkkommando wird vom Rechner zum Server des Projekts gesendet, der dann die gewünschte Aktion auf seinem Treiber durchführt. 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 Hilfen, 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 Diagnose Viewer (main.chm::/12464.htm), das mit zenon mitinstalliert wird. Sie finden es unter *Start/Alle Programme/zenon/Tools 7.11 -> Diagviewer*.

zenon Treiber protokollieren alle Fehler in Log-Dateien. Der Standardordner für die Log-Dateien ist der Ordner `LOG` unterhalb des Ordners `ProgramData`, zum Beispiel: `C:\ProgramData\zenon\zenon7.11\LOG` für die zenon Version 7.11. Log-Dateien sind Textdateien mit einer speziellen Struktur.

**Achtung:** Mit den Standardeinstellungen zeichnet ein Treiber nur Fehlerinformationen auf. Mit dem **Diagnose 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 Diagnose Viewer kann man auch:

- ▶ eben erstellte Einträge live mitverfolgen
- ▶ die Aufzeichnungseinstellungen anpassen
- ▶ den Ordner, in dem die Log-Dateien gespeichert werden, ändern

#### Hinweise:

1. Unter Windows CE werden aus Ressourcegründen auch Fehler standardmäßig nicht protokolliert.
2. Der Diagnose Viewer zeigt alle Einträge in UTC (Koordinierter Weltzeit) an und nicht in der lokalen Zeit.
3. Der Diagnose 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.

4. 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**.
5. 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.
6. 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 **Diagnose Viewers** weiter aufgezeichnet.

**Info**

Weitere Informationen zum **Diagnose Viewer** finden Sie im Kapitel **Diagnose Viewer** ([main.chm::/12464.htm](#)).

## 9.2 Checkliste

Fragen und Hinweise zur Fehlereingrenzung:

### ALLGEMEINE FEHLERSUCHE

- ▶ Ist die Steuerung an die Stromversorgung angeschlossen?
- ▶ Analyse mit Hilfe des **Diagnose Viewers** (auf Seite 37):  
-> Welche Meldungen werden angezeigt?
- ▶ Wird für die Verbindung von Steuerung und PC das korrekte, vom Hersteller empfohlene Kabel verwendet?

- ▶ Wurde der richtige COM Port ausgewählt?
- ▶ Stimmen die seriellen Kommunikationsparameter (Baudrate, Parität, Start/Stop Bits,...) überein?
- ▶ Wird der COM Port durch eine andere Anwendung blockiert?
- ▶ Wurde die Netzadresse in den Adresseigenschaften der Variable korrekt eingestellt?
  - Stimmt die Adressierung mit der Konfiguration im Treiberdialog überein?
  - Entspricht die Netzadresse der Adresse der Zielstation?
- ▶ Wird in der Variable der richtige Objekttyp verwendet?

**Beispiel:** Treibervariablen sind reine Statistikvariablen und kommunizieren nicht mit der Steuerung. (Siehe Kapitel Treibervariablen (auf Seite 29).)

- ▶ Stimmt die Offset-Adressierung der Variable mit der in der Steuerung überein?

## **MANCHE VARIABLEN MELDEN INVALID**

- ▶ INVALID Bits beziehen sich immer auf eine Netzadresse.
- ▶ Mindestens eine Variable der Netzadresse ist gestört.

## **WERTE WERDEN NICHT ANGEZEIGT, ZAHLENWERTE BLEIBEN LEER**

Treiber läuft nicht. Überprüfen Sie die:

- ▶ Installation von zenon
  - ▶ Installation des Treibers
  - ▶ Installation aller Komponenten
- > Achten Sie auf Fehlermeldungen beim Start der Runtime.

## **VARIABLEN WERDEN MIT BLAUEM PUNKT ANGEZEIGT**

Die Kommunikation im Netzwerk ist gestört:

- ▶ Bei einem Netzwerkprojekt:
  - Läuft das Netzwerkprojekt auch auf dem Server?

- Bei einem Standalone-Projekt oder Netzwerkprojekt, das auf dem Server läuft:  
Deaktivieren Sie die Eigenschaft `Nur von Standby Server anfordern im Knoten`  
`Treiber Anbindung/Adressierung`.

## **WERTE WERDEN FALSCH ANGEZEIGT**

Überprüfen Sie die Angaben zur Berechnung im Knoten `Wertberechnung der Variablen-Eigenschaften`?

## **TREIBER FALLT SPORADISCH AUS**

Analyse mit Hilfe des `Diagnose Viewers` (auf Seite 37):

-> Welche Meldungen werden angezeigt?