



zenon
by COPA-DATA

zenon Treiber Handbuch OMR_FINS

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	4
2	OMR_FINS	4
3	OMR_FINS - Datenblatt	4
4	Treiber-Historie	6
5	Voraussetzungen.....	7
5.1	PC.....	7
6	Konfiguration	7
6.1	Allgemein.....	8
6.2	COM.....	12
6.3	Kommunikation	13
7	Adressierung	15
8	Treiberobjekte und Datentypen.....	17
8.1	Treiberobjekte.....	17
9	Zuordnung der Datentypen	18
10	Treiberspezifische Funktionen	19
11	Funktion Treiberkommandos.....	21
12	Test	26
13	Fehleranalyse.....	27
13.1	Analysetool.....	27
13.2	Treiberüberwachung.....	28
13.3	Fehlernummern	29
13.4	Checkliste	29

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 Hilfef Kapitel 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 OMR_FINS

3 OMR_FINS - Datenblatt

Allgemein:	
Treiberdateiname	OMR_FINS.exe

Allgemein:	
Treiberbezeichnung	Omron FINS Treiber
Steuerungs-Typen	OMRON SYSMAC
Steuerungs-Hersteller	Omron

Treiber unterstützt:	
Protokoll	Omron FINS
Adressierung: Adress-basiert	Address based
Adressierung: Namens-basiert	--
Kommunikation spontan	--
Kommunikation pollend	X
Online Browsing	--
Offline Browsing	--
Echtzeitfähig	--
Blockwrite	X
Modemfähig	--
RDA numerisch	--
RDA String	--
Hysterese	--
erweiterte API	--
Unterstützung von Statusbit WR-SUC	--
alternative IP-Adresse	--

Voraussetzungen:	
Hardware PC	RS 232 serielle Schnittstelle, Kabeltyp: SYSMAC WAY; Standard

Voraussetzungen:	
	Netzwerkkarte
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	1300	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 versionsunabhängig

5 Voraussetzungen

Dieses Kapitel enthält Informationen welche Voraussetzungen für die Verwendung des Treibers erforderlich sind.

5.1 PC

HARDWARE

Serielle Schnittstelle RS232 oder standard Netzwerkkarte (TCP/IP)

SOFTWARE

Die Treiber Datei OMR_FINS.exe in das aktuelle Installationsverzeichnis kopieren (wenn nicht bereits vorhanden) und ins TREIBER_DE.XML File über das Tool DriverInfo.exe eintragen.

Info

Windows CE wird derzeit nicht unterstützt.

6 Konfiguration

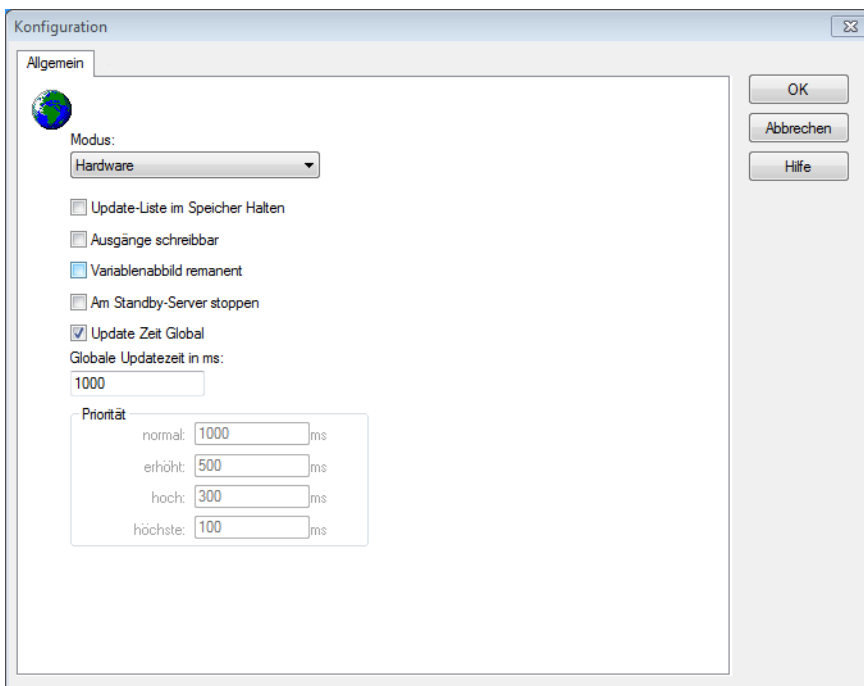
In diesem Kapitel lesen Sie, wie Sie den Treiber im Projekt anlegen und welche Einstellungen beim Treiber möglich sind.

 **Info**

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

6.1 Allgemein

Beim Anlegen eines Treibers wird der Konfigurationsdialog geöffnet. Um den Dialog später zum Bearbeiten zu öffnen, führen Sie einen Doppelklick auf den Treiber in der Liste aus oder klicken Sie auf die Eigenschaft **Konfiguration**.



Option	Beschreibung
<p>Modus</p>	<p>Ermöglicht ein Umschalten zwischen Hardware und Simulationsmodus</p> <ul style="list-style-type: none"> ▶ <i>Hardware:</i> Die Verbindung zur Steuerung wird hergestellt. ▶ <i>Simulation - statisch:</i> Es wird keine Kommunikation zur Steuerung aufgebaut, die Werte werden vom Treiber simuliert. In diesem Modus bleiben die Werte konstant oder die Variablen behalten die über zenon Logic gesetzten Werte. Jede Variable hat seinen eigenen Speicherbereich. Zum Beispiel zwei Variablen vom

Option	Beschreibung
	<p>Typ Merker mit Offset 79, können zur Runtime unterschiedliche Werte haben und beeinflussen sich gegenseitig nicht. Ausnahme: Der Simulatortreiber.</p> <ul style="list-style-type: none"> ▶ <i>Simulation - zählend:</i> 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. ▶ <i>Simulation - programmiert:</i> 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.
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	<ul style="list-style-type: none"> ▶ <i>Aktiv:</i> Ausgänge können beschrieben werden. ▶ <i>Inaktiv:</i> Das Beschreiben der Ausgänge wird unterbunden. <p>Hinweis: 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 einer dieser Status aktiv ist:</p> <ul style="list-style-type: none"> ▶ Benutzerstatus M1 (0) bis M8 (7)

Option	Beschreibung
	<ul style="list-style-type: none"> ▶ <i>REVISION(9)</i> ▶ <i>AUS(20)</i> ▶ <i>ERSATZWERT(27)</i> <p>Das Variablenabbild wird immer gespeichert wenn:</p> <ul style="list-style-type: none"> ▶ die Variable vom Objekttyp Kommunikationsdetails ist ▶ der Treiber im Simulationsmodus läuft. (nicht programmierte Simulation) <p>Folgende Status werden beim Start der Runtime nicht restauriert:</p> <ul style="list-style-type: none"> ▶ <i>SELECT(8)</i> ▶ <i>WR-ACK(40)</i> ▶ <i>WR-SUC(41)</i> <p>Der Modus Simulation - programmiert beim Treiberstart ist kein Kriterium, um das remanente Variablenabbild zu restaurieren.</p>
<p>Am Standby Server stoppen</p>	<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>Achtung: Ist diese Option aktiv, ist die lückenlose Archivierung nicht mehr gewährleistet.</p> <ul style="list-style-type: none"> ▶ <i>Aktiv:</i> 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 abgeschaltet, sondern einen leeren Wert. Damit wird verhindert, dass beim Hochstufen zum Server nicht relevante Werte in AML , CEL und Archiv erzeugt werden. <p>Default: <i>inaktiv</i></p> <p>Hinweis: Nicht verfügbar, wenn CE Terminal als Datenserver dient. Weitere Informationen dazu erhalten Sie im Handbuch zenon Operator im Kapitel CE Terminal</p>

Option	Beschreibung
	als Datenserver.
Update Zeit Global	<p>Einstellung für globale Update-Zeiten in Millisekunden:</p> <ul style="list-style-type: none"> ▶ <i>Aktiv:</i> Die eingestellte Globale Update Zeit wird für alle Variablen im Projekt verwendet. Die bei den Variablen eingestellte Priorität wird nicht verwendet. ▶ <i>Inaktiv:</i> Die eingestellten Prioritäten werden für die einzelnen Variablen verwendet. <p>Ausnahmen: Spontane Treiber ignorieren diese Option. Sie nutzen in der Regel die kürzest mögliche Update Zeit. Details siehe Abschnitt Update Zeit spontane Treiber.</p>
Priorität	<p>Hier werden die Pollingzeiten der einzelnen Prioritätsklassen eingestellt. Alle Variablen mit der entsprechenden Priorität werden in der eingestellten Zeit gepollt.</p> <p>Die Zuordnung der Variablen erfolgt separat bei jeder Variablen über die Einstellungen in den Variableneigenschaften.</p> <p>Mit den Prioritätsklassen kann die Kommunikation der einzelnen Variablen auf die Wichtigkeit oder benötigte Aktualität abgestuft werden. Daraus ergibt sich eine verbesserte Verteilung der Kommunikationslast.</p> <p>Achtung: Prioritätsklassen werden nicht von jedem Treiber unterstützt, z.B. von spontan kommunizierenden zenon Treibern.</p>

DIALOG BEENDEN

Option	Beschreibung
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.

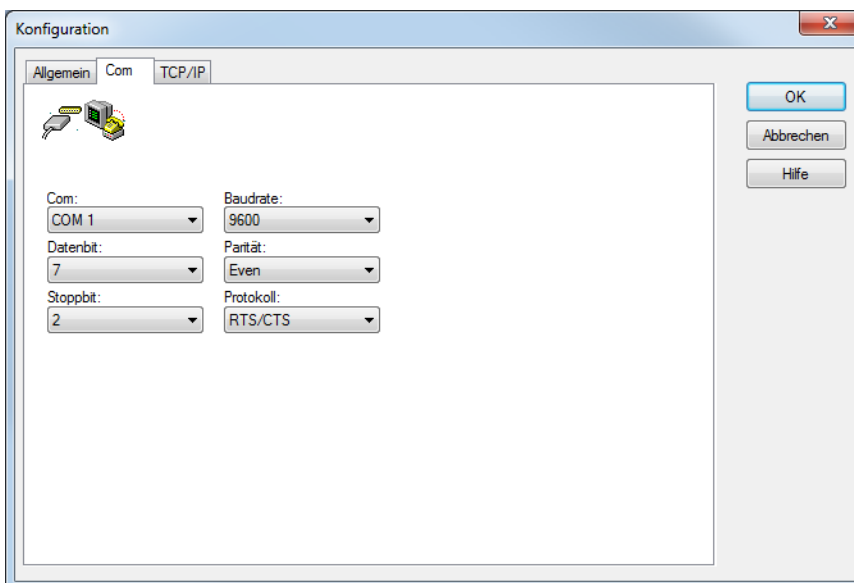
UPDATE ZEIT SPONTANE TREIBER

Bei spontanen Treibern wird beim **Sollwert Setzen, Advisen** von Variablen und bei **Requests** sofort ein Lesezyklus ausgelöst - unabhängig von der eingestellten Update Zeit. Damit wird sicher gestellt, dass der Wert nach dem Schreiben in der Visualisierung sofort zur Verfügung steht. In der Regel beträgt die Updatezeit 100 ms.

Spontane Treiber sind **ArchDrv, BiffiDCM, BrTcp32, DNP3, Esser32, FipDrv32, FpcDrv32, IEC850, IEC870, IEC870_103, Otis, RTK9000, S7DCOS, SAIA_Slave, STRATON32** und **Trend32**.

6.2 COM

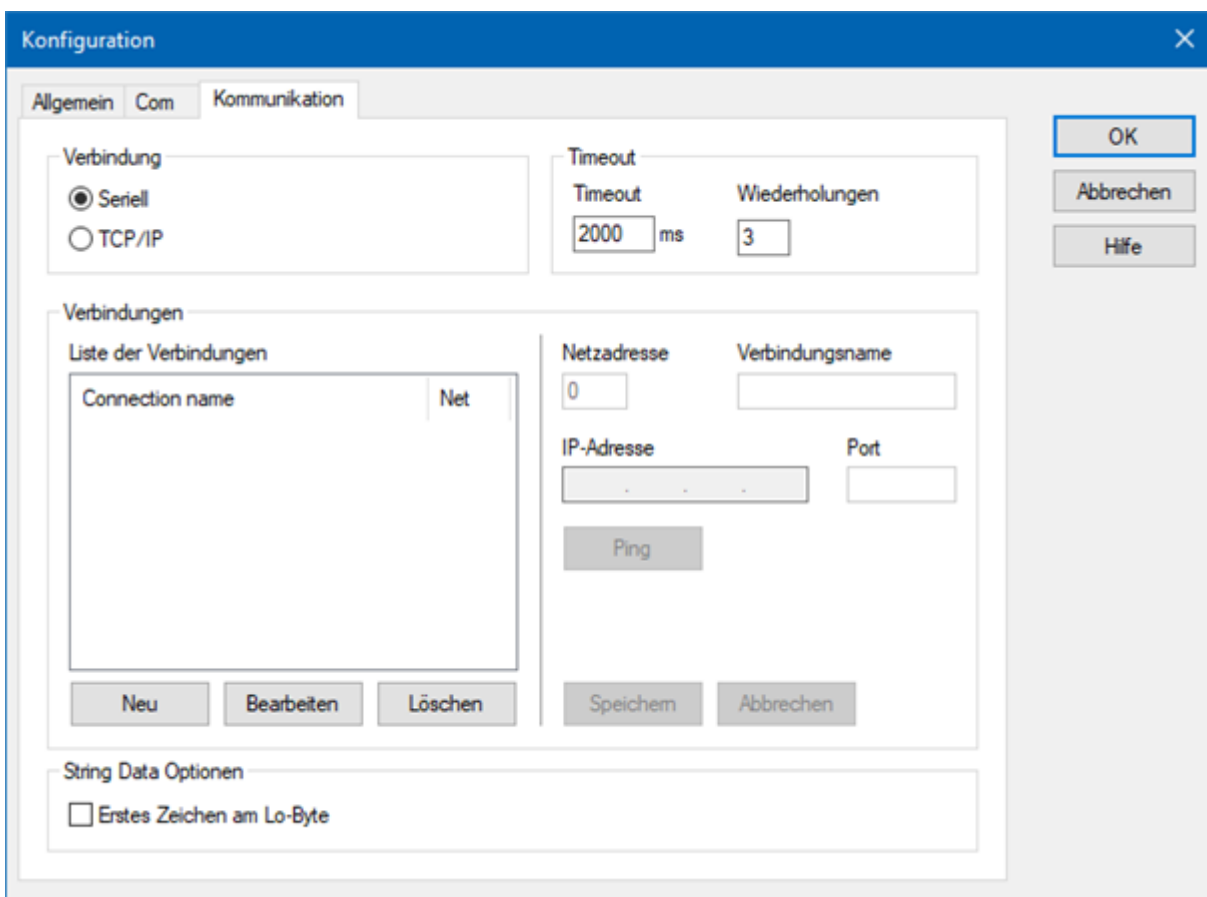
Einstellung der Kommunikationsparameter für die Serielle Schnittstelle



Parameter	Beschreibung
Com	Auswahl der seriellen Schnittstelle, an der die Steuerung angeschlossen ist.
Baudrate	Auswahl Baudrate. Anpassen an Steuerung. Auswahl aus Dropdownliste. Default: 9600 Eingabebereich: 110 bis 256000
Datenbit	Datenwortlänge in Bit: Default: 7
Parität	Einstellungen zur Parität der Verbindung Default: Even

Parameter	Beschreibung
Stopbit	Anzahl der Stopbits der Verbindung. Default: 2
Protokoll	Protokoll der Verbindung. Default: <i>RTS/CTS</i>

6.3 Kommunikation



VERBINDUNG

Parameter	Beschreibung
Verbindung	Verbindungsart. Auswahl aus Optionsfeld: ▶ <i>Seriell:</i>

Parameter	Beschreibung
	Kommunikation über serielle Verbindung. <ul style="list-style-type: none"> ▶ <i>TCP/IP</i>: Kommunikation via TCP/IP Protokoll.

TIMEOUT

Optionen für Fehlerwartezeit.

Parameter	Beschreibung
Timeout	Wartezeit für Verbindungsaufbau in Millisekunden.
Wiederholungen	Anzahl der Wiederholungen bei erfolglosem Verbindungsaufbau.

VERBINDUNGEN

Einstellungen der Verbindungen.

Parameter	Beschreibung
Liste der Verbindungen	Liste aller definierten Verbindungen zu Steuerungen.
Netzadresse	Entspricht der Eigenschaft Netzadresse bei den Variableneigenschaften.
Verbindungsname	Frei wählbarer Name.
IP Adresse	IP-Adresse der Steuerung.
Port	Port-Adresse der Steuerung. Details finden Sie im Handbuch Ihrer Steuerung.
Ping	Sendet einen Ping zur IP-Adresse, die für diese Verbindung konfiguriert wurde. Ermöglicht das Testen der Verbindung zum Gerät. Wird der Ping negativ abgeschlossen, prüfen Sie die IP-Adresse und ob das Gerät online ist.
Neu	Erstellt eine neue Verbindung.
Bearbeiten	Öffnet markierte Verbindung zum Bearbeiten.
Löschen	Löscht markierte Verbindung aus Liste.
Speichern	Übernimmt alle Änderungen für bearbeitete

Parameter	Beschreibung
	Verbindung und schließt Editiermöglichkeit.
Abbrechen	Verwirft alle Änderungen für bearbeitete Verbindung und schließt Editiermöglichkeit.
OK	Änderungen im Dialog übernehmen und Dialog schließen. Nur verfügbar, wenn keine Verbindung im Zustand "bearbeiten" ist.
Abbrechen	Verwirft alle Änderungen und schließt Dialog.
Hilfe	Öffnet Online-Hilfe.

STRING DATA OPTIONEN

Einstellung für STRINGS.

Parameter	Beschreibung
Erstes Zeichen am Lo-Byte	Mit dieser Checkbox stellen Sie die Byte-Order für STRINGS in der Steuerung ein. <ul style="list-style-type: none"> ▶ <i>Aktiv</i>: Byte-Order wird umgekehrt. Wird zum Beispiel von Omron NJ benötigt. Default: <i>inaktiv</i>

Info

Maximale Anzahl der Verbindungen: 256 (0-255).

7 Adressierung

Gruppe/Eigenschaft	Beschreibung
Allgemein	Gruppe mit allgemeinen Eigenschaften.
Name	Frei vergebbarer Name. Achtung: Je zenon Projekt muss der Name eindeutig sein.

Gruppe/Eigenschaft	Beschreibung
Kennung	Frei vergebbare Kennung. Z. B. für Betriebsmittelkennung , Kommentar usw.
Adressierung	
Netzadresse	Netzadresse der Variablen. Diese Adresse bezieht sich auf die Netzadresse der Verbindungsprojektierung im Treiber. Damit wird ausgewählt auf welcher Steuerung sich die Variable befindet.
Datenbaustein	Für Variablen vom Objekttyp <i>Erweiterter Datenbaustein</i> muss hier die Datenbaustein-Nummer angegeben werden. Einstellbar von 0 bis 4294967295. Den genauen maximalen Bereich für Datenbausteine entnehmen Sie dem Handbuch für die Steuerung.
Offset	Offset der Variablen. Entspricht der Speicheradresse der Variablen in der Steuerung. Einstellbar von 0 bis 4294967295.
Ausrichtung	Wird für diesen Treiber nicht verwendet.
Bitnummer	Offset der Variablen. Entspricht der Speicheradresse der Variablen in der Steuerung. Einstellbar von 0 bis 4294967295.
Stringlänge	Nur verfügbar bei String-Variablen. Maximale Anzahl von Zeichen, die die Variable aufnehmen kann.
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	Einstellen der Prioritätsklasse. Damit wird die Variable der Prioritätsklasse zugeordnet, so wie sie im Treiberdialog in der Registerkarte Allgemein konfiguriert wurde. Die Prioritätsklassen werden nur dann verwendet, wenn die Update Zeit Global deaktiviert ist. Bei aktivierter Option Update Zeit Global und Verwendung der Prioritätsklassen erfolgt ein Fehlereintrag in die Log-Datei des Systems.

Gruppe/Eigenschaft	Beschreibung
	Der Treiber verwendet die höchstmögliche Priorität.

VARIABLENADRESSIERUNG ÜBER

Adresse

Die Variablen werden über ihren Offset einem Speicherbereich in der Steuerung zugeordnet.

Info

Die Adressierung der SPS erfolgt als WORD. Bei ungerader Stringlänge wird immer ein Zeichen mehr gelesen bzw. geschrieben.

8 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.

8.1 Treiberobjekte

TREIBER OBJEKTTYPEN UND UNTERSTÜTZTE IEC DATENTYPEN FÜR PROZESSVARIABLEN IM LEITSYSTEM

Treiber Objekttypen	Kanaltyp	Unterstützte Datentypen (Datenart)	Lesen	Schreiben	Kommentar
IR Area	64	<i>BOOL, INT, UINT</i>	X	X	
HR Area	65	<i>BOOL, INT, DINT, REAL, UINT, UDINT, STRING</i>	X	X	
AR Area	66	<i>BOOL, INT, UINT</i>	X	X	

Treiber Objekttypen	Kanaltyp	Unterstützte Datentypen (Datenart)	Lesen	Schreiben	Kommentar
TIM/CNT	68	BOOL	X	X	
EM Area	73	BOOL, INT, DINT, REAL, UINT, UDINT, STRING	X	X	Bank Nr. 0...12 in Datenbaustein definieren. Offset 0...32767
DM Area	70	BOOL, INT, DINT, REAL, UINT, UDINT, STRING	X	X	
DM Area BCD	71	INT, DINT, UINT, UDINT	X	X	
CIO Area	71	BOOL, INT, DINT, REAL, UINT, UDINT, STRING	X	X	
WORK Area	72	BOOL, INT, DINT, REAL, UINT, UDINT, STRING	X	X	

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.

9 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.

BEISPIELE FÜR ALLE MÖGLICHEN IEC DATENTYPEN

SPS	zenon
I16	INT
I32	DINT

SPS	zenon
U16	UINT
U32	UDINT
REAL	REAL
Boolean	BOOL

DATENART

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

10 Treiberspezifische Funktionen

Dieser Treiber unterstützt folgende Funktionen:

Treiberspezifische Funktion	Beschreibung
Blockwrite	X Details siehe Abschnitt Blockwrite .
Browsen	--
Echtzeitstempleung	--
Erweiterte Fehlerdatei	--
Fehlerdatei	Der Treiber unterstützt die übliche Fehlerdatei, ab zenon 6.20 ist zentrales Logging möglich.
Fehlerwartezeit	Keine Einstellung möglich.
RDA	--
Redundanz	X
Seriellles Logging	X Details siehe Abschnitt Seriellles Logging .
Zugriffsverfahren	<ul style="list-style-type: none"> ▶ Polling: Zyklische Abfrage der Variablen/des Treibers.

Treiberspezifische Funktion	Beschreibung
	▶ Spontan: Keine Unterstützung

Legende:

- ▶ **X:** wird unterstützt
- ▶ **--:** keine Unterstützung

BLOCKWRITE

Um **Blockwrite** zu aktivieren, muss in der **project.ini** ein Eintrag gesetzt sein:

- ▶ Abschnitt:
[OMR_FINS]
- ▶ Eintrag:
BLOCKWRITE=1

SERIELLES LOGGING

Um serielles Logging zu aktivieren, muss in der **project.ini** ein Eintrag gesetzt sein:

- ▶ Abschnitt:
[RS232LOG]
- ▶ Eintrag:
LOGCOMx=
 - x:** Nummer der gewählten Schnittstelle
 - ▶ **0:** kein Logging
 - ▶ **1:** serielles Logging aktiv
Es wird eine Datei **LOG_COMxxx.TXT** im Ordner des Treibers erstellt.

Beispiel für die Aktivierung der Protokollierung für **COM 1:**

[RS232LOG]

LOGCOM1=1



Info

Aktivieren Sie die Protokollierung nur bei Problemen und nur kurzfristig. Die Protokollierung benötigt erhebliche Ressourcen. Außerdem belegt die LOG-Datei innerhalb kurzer Zeit erheblichen Speicherplatz.

11 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!

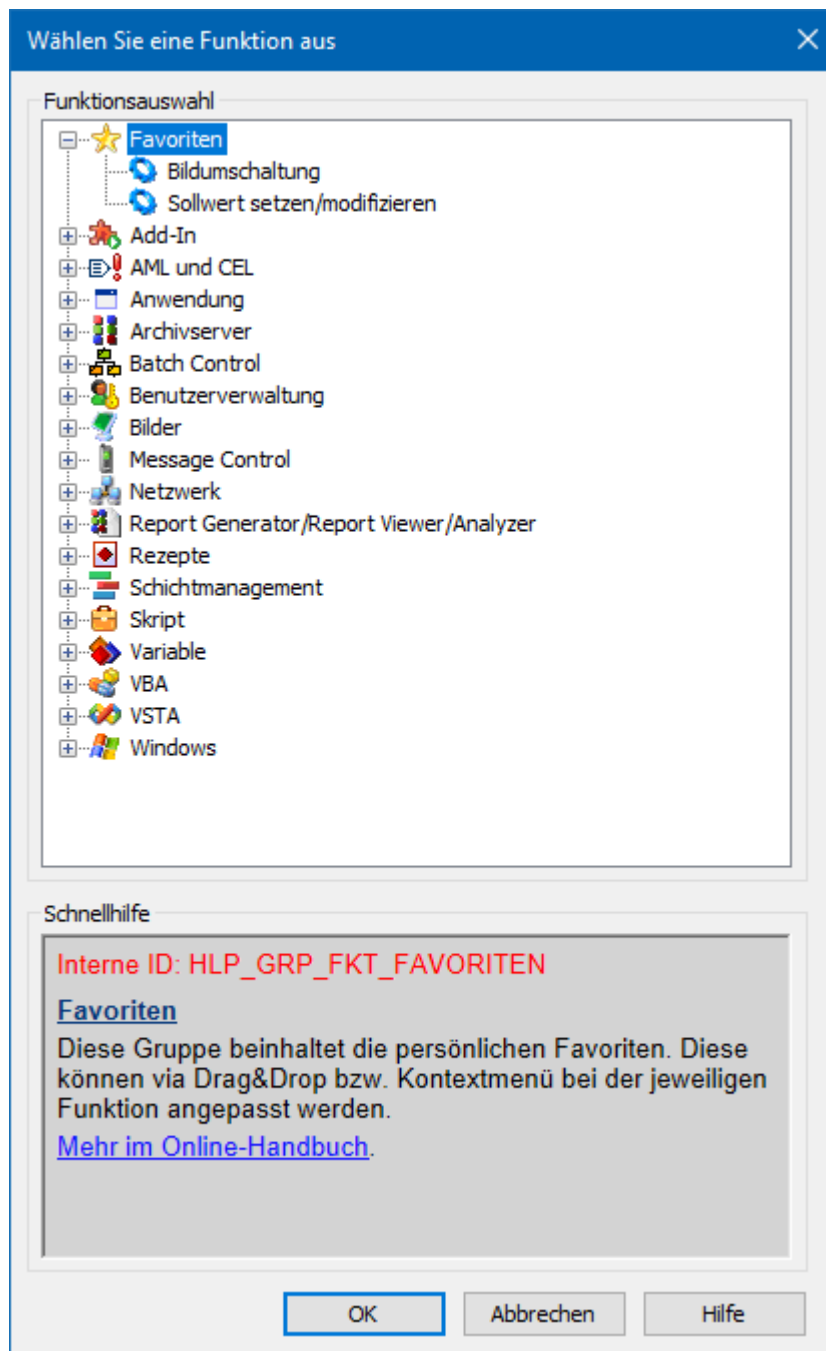
PROJEKTIERUNG DER FUNKTION

Die Projektierung erfolgt über die Funktion **Treiberkommandos**.

Um die Funktion zu projektieren:

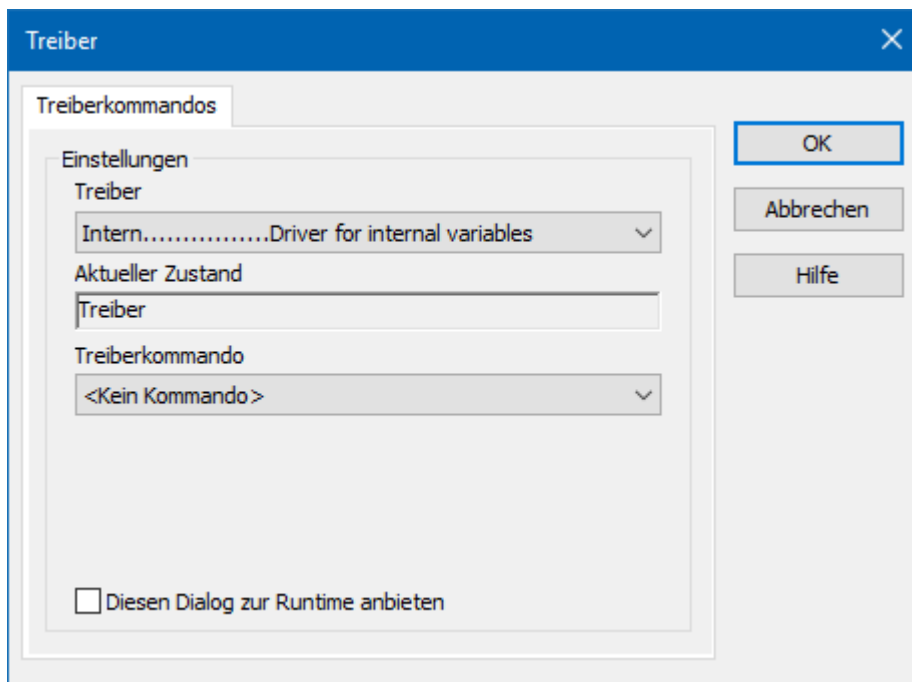
1. Legen Sie im zenon Editor eine neue Funktion an.

Der Dialog zur Auswahl einer Funktion wird geöffnet.



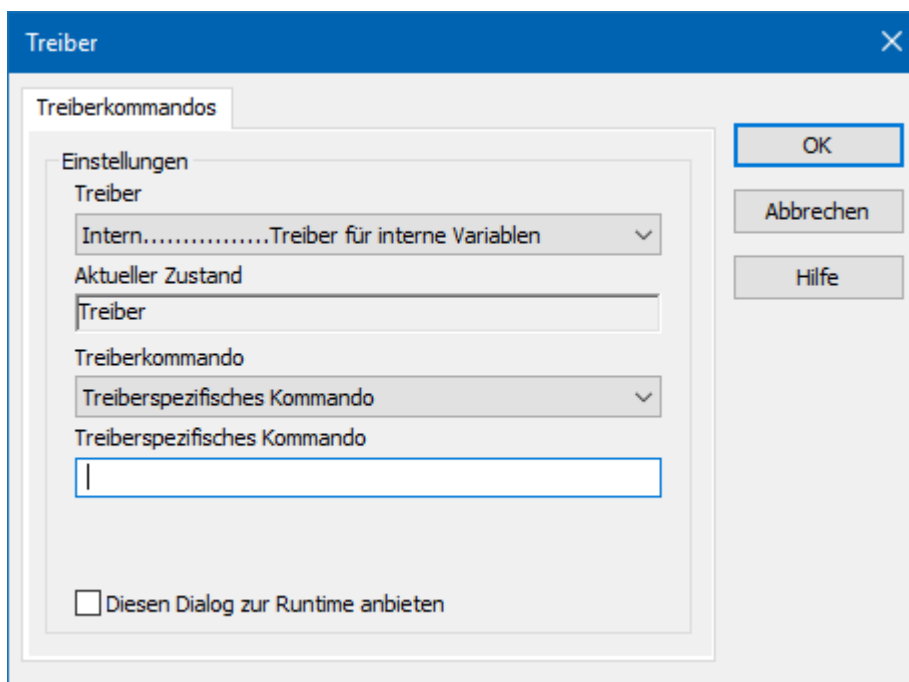
2. Navigieren Sie zum Knoten **Variable**.
3. Wählen Sie den Eintrag **Treiberkommandos**.

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 <i>Treiberspezifisches Kommando</i> gewählt wurde.
Diesen Dialog zur Runtime anbieten	Konfiguration, ob die Konfiguration in der Runtime geändert werden kann: <ul style="list-style-type: none"> ▶ <i>Aktiv:</i> 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. ▶ <i>Inaktiv:</i> Die Editor-Konfiguration wird in der Runtime beim Ausführen der Funktion angewendet. Default: <i>inaktiv</i>

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
<i>Kein Kommando</i>	Es wird kein Kommando gesendet. Damit kann auch ein bereits bestehendes Kommando aus einer projektierten Funktion entfernt werden.
<i>Treiber starten (Onlinemodus)</i>	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.
<i>Treiber stoppen (Offlinemodus)</i>	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 <i>Abgeschaltet (OFF; Bit 20)</i> .
<i>Treiber in Simulationsmodus</i>	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.
<i>Treiber in Hardwaremodus</i>	Treiber wird in den Hardwaremodus gesetzt. Für die Variablen des Treibers werden die Werte von der angeschlossenen Hardware (z. B. SPS, Bussystem, ...) angezeigt.
<i>Treiberspezifisches Kommando</i>	Eingabe eines treiberspezifischen Kommandos. Öffnet Eingabefeld für die Eingabe eines Kommandos.
<i>Treiber Sollwertsetzen aktivieren</i>	Sollwert setzen auf Treiber ist möglich.
<i>Treiber Sollwertsetzen deaktivieren</i>	Sollwert setzen auf Treiber wird verhindert.
<i>Verbindung mit Modem aufbauen</i>	Verbindung aufbauen (für Modem-Treiber). Öffnet Eingabefelder für Hardware-Adresse und Eingabe der zu wählenden Nummer.
<i>Verbindung mit Modem trennen</i>	Verbindung beenden (für Modem-Treiber).
<i>Treiber in Simulationsmodus zählend</i>	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.
<i>Treiber in Simulationsmodus statisch</i>	Es wird keine Kommunikation zur Steuerung aufgebaut. Alle Werte werden mit 0 initialisiert.
<i>Treiber in Simulationsmodus programmiert</i>	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.

12 Test

GETESTET MIT FOLGENDER HARD- UND SOFTWARE

omron SYSMAC CJ1M /CPU11 (Programmable Controller)

TESTAUFBAU

Getestet mit serieller Verbindung (serielles Kabel, KEIN Nullmodemkabel, Kabeltyp: SYSMAC WAY), Parameter: 9600, 7, 2, even, no; und TCP/IP Verbindung mit default Port 9600. Die IP-Adresse der Steuerung kann über Web-Interface eingestellt werden z. B. [http://\[IP-Adresse der Steuerung\]/0](http://[IP-Adresse der Steuerung]/0) oder über einen DM Speicherbereich mittels der seriellen Verbindung (Offset: $m = D30000 + (100 \times \text{unit number}) + 98$; in diesen 4 Byte wird die IP-Adresse eingetragen, die die Steuerung dann verwendet).

Details finden Sie auch in der Omron Dokumentation
'W441E101_Ethernet+CPUs_Operation_Manual.pdf'

EINSCHRÄNKUNGEN

Mit unserem Testaufbau war schreiben auf HR und AR nicht möglich. Bei Treiber Objekttyp IR war weder schreiben noch lesen möglich.

13 Fehleranalyse

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

13.1 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.

13.2 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 Zurücksetzen 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 angemeldet 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
<i>Communication with driver:<drvExe>/<drvDesc>(id:<drvId>) timed out. No communication for <time> ms.</i>	Keine Kommunikation mit Treiber innerhalb der angegeben Zeit. <ul style="list-style-type: none"> ▶ <time>: Zeitangabe in Millisekunden ▶ <drvDesc>: Treibername ▶ <drvExe>: Treiber EXE-Name ▶ <drvId>: Treiber-ID im zenon Projekt
<i>Communication with %s timed out. Invalid-Bit will be set.</i>	Die Kommunikation zum Treiber %s konnte bei 5 Versuchen nicht innerhalb von 60 Sekunden aufgebaut werden. Bei der Variable wird das <i>INVALID</i> -Bit gesetzt.
<i>Communication with %s timed out. Timeout happened %d times</i>	Die Kommunikation zum Treiber %s konnte %d Mal nicht innerhalb von 60 Sekunden aufgebaut werden.

13.3 Fehlernummern

Im Error-Logfile des Treibers wird bei Kommunikationsproblemen ein Eintrag erstellt, in welchem die Fehlerursache mittels Nummer angegeben wird.

Info

Die Bedeutung der Fehlernummern entnehmen Sie der FINS Dokumentation Section 5-1-3 „End Codes“. Hier sind sämtliche Fehlernummern dokumentiert.

13.4 Checkliste

- ▶ Ist die Steuerung korrekt angeschlossen (serielles Verbindungskabel – KEIN Nullmodemkabel bei serieller Verbindung) und die Verbindung korrekt konfiguriert (Parameter: 9600,7,2,even,no)?
- ▶ Ist die IP-Adresse der Steuerung korrekt konfiguriert (bei Verbindung über TCP/IP)? Kann über Web-Interface die Steuerung oder z. B. Ping in der Treiberkonfiguration/Conn. TCP/IP angesprochen wrden

- ▶ Wurde die Fehlerdatei (bis zenon 6.01) bzw das zentrale Logfile (ab zenon 6.20) analysiert