zenon Treiber Handbuch

IEC870_10332

v.8.00

©2018 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. IEC870_10332 5

3. IEC870_10332 - Datenblatt 6

4. Treiber-Historie 8

5. Voraussetzungen 9

6. Konfiguration 9

6.1 Anlegen eines Treibers 10

6.2 Einstellungen im Treiberdialog 13

6.2.1 Allgemein 14

6.2.2 Com 18

6.2.3 Verbindungen 19

7. Variablen anlegen 26

7.1 Variablen im Editor anlegen 26

7.2 Adressierung 29

7.3 Treiberobjekte und Datentypen 30

7.3.1 Treiberobjekte 30

7.3.2 Zuordnung der Datentypen 33

7.4 Variablen anlegen durch Import 34

7.4.1 XML Import 35

7.4.2 DBF Import/Export 36

7.5 Kommunikationsdetails (Treibervariablen) 42

8. Treiberspezifische Funktionen 47

9. Funktion Treiberkommandos 51

10. Interoperability List 54

11. Fehleranalyse 62

11.1 Analysetool 62

11.2 Checkliste 63

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 Support-Team, das 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. IEC870_10332

Treiber für das IEC 60870-5-103 Protokoll (seriell oder TCP/IP).

Die Kommunikation zwischen Leitsystem und der Steuerung basiert auf IEC 60870-5-103. Das Leitsystem ('the control system' - primär Station nach IEC 60870-5-2) agiert auf der Protokollebene als Master und die Steuerung ('the protection equipment') als Slave.

Der Treiber sendet die ASDUs (Application Service Data Units) in sogenannter control direction, die Steuerung in monitor direction.

Im seriellen Fall agiert das Leitsystem als Master in unbalancierten Kommunikationsmodus. Der Kommunikationskanal kann zwischen einem 60870 Master und mehreren 60870 Slaves geteilt werden. Bei TCP/IP agiert das Leitsystem auf der Protokollebene als Master und auf TCP Ebene als Client.

Die Kommunikation ist Spontan. Zusätzlich, der Treiber sendet zyklisch General Interrogation nach der **GenInt** Einstellung in der Treiberkonfiguration.

3. IEC870_10332 - Datenblatt

Allgemein:	
Treiberdateiname	IEC870_10332.exe
Treiberbezeichnung	IEC60870 103
Steuerungs-Typen	IEC60870-5-103 kompatible Steuerungen
Steuerungs-Hersteller	IEC;

Treiber unterstützt:	
Protokoll	IEC 60870-5-103;
Adressierung: Adress-basiert	X
Adressierung: Namens-basiert	--
Kommunikation spontan	X
Kommunikation pollend	--
Online Browsing	--
Offline Browsing	--
Echtzeitfähig	--
Blockwrite	--
Modemfähig	--
Serielles Logging	X

RDA numerisch	--
RDA String	--
Hysterese	--
erweiterte API	--
Unterstützung von Statusbit **WR-SUC**	X
alternative IP-Adresse	--

Voraussetzungen:	
Hardware PC	Standard Netzwerkkarte; Serieller Anschluss RS232 od. RS 485.
Software PC	--
Hardware Steuerung	--
Software Steuerung	--
Benötigt v-dll	X

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

4. Treiber-Historie

Datum	Treiberversion	Änderung
07.07.08	400	Treiberdokumentation wurde neu erstellt
14.11.08	1200	- Treiber Objekttyp MONITORING TRANSIENT - INVALID-Bit ist verfügbar - Measurands als REAL - Übernahme des Zeitstempels vom externen Gerät - Einträge in LOG - Unterverzeichnisse (pro Netzadresse) für die Disturbance Data Dateien
23.03.09	1400	- Treiber Objekttyp CONNECTION STATE - Treiber Objekttyp TIME SYNC - PN-Bit für Kommandos ist verfügbar - SB-Bit ist verfügbar (für Variablen die nicht in GI geantwortet waren) - die Übertragungsursache (Cause of Transmission - COT) ist verfügbar

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. Voraussetzungen

Dieses Kapitel enthält Informationen zu den Voraussetzungen, die für die Verwendung des Treibers erforderlich sind.

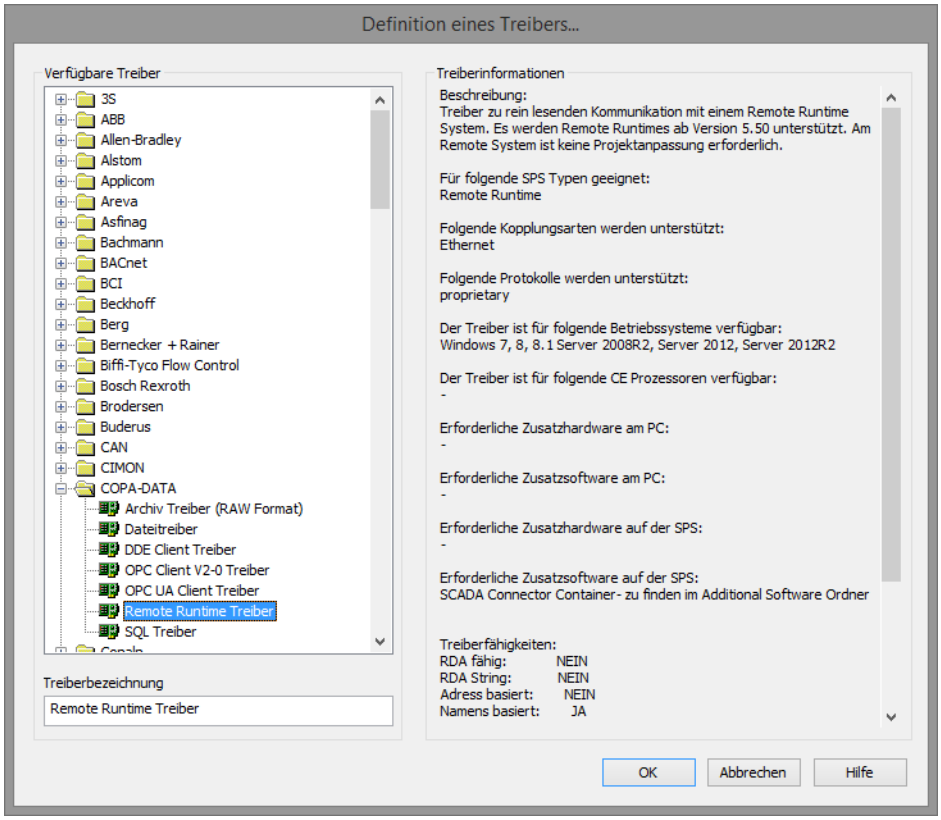
6. Konfiguration

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 (main.chm::/15247.htm) der Online-Hilfe.*

6.1 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 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].`

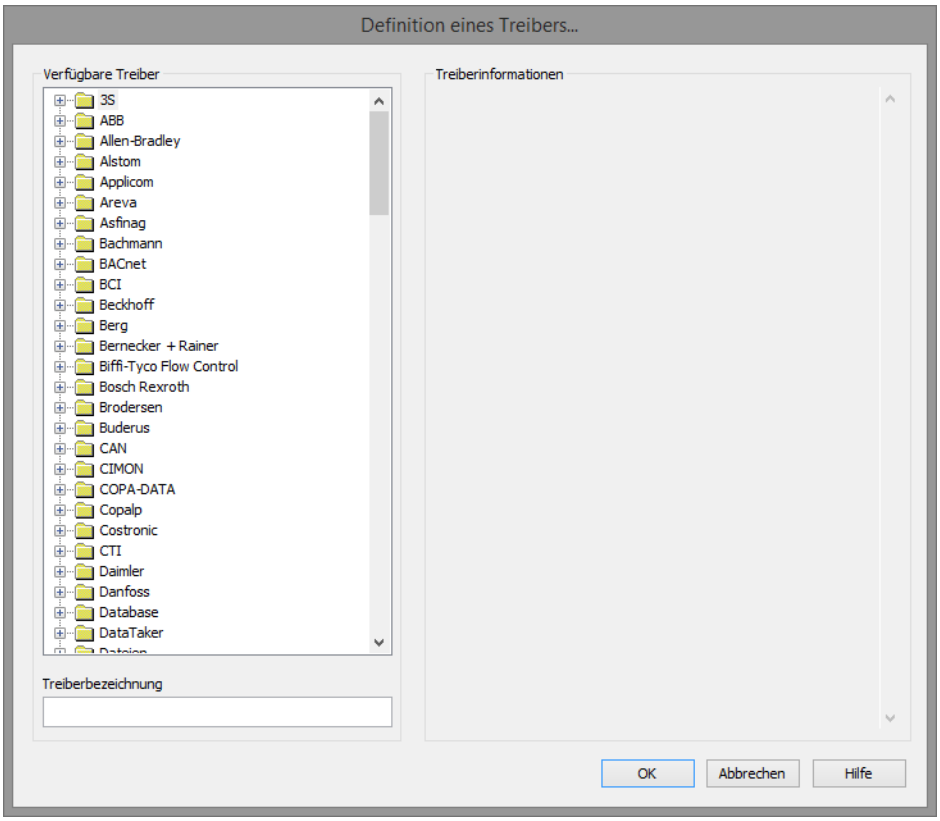
TREIBER NEU ANLEGEN

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:

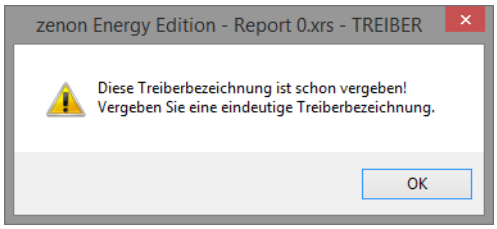
- Die **Treiberbezeichnung** muss eindeutig sein.
  Wird ein Treiber mehrmals im Projekt verwendet, so muss jeweils eine neue Bezeichnung vergeben werden.
  Dies wird durch Klick auf die Schaltfläche **OK** evaluiert. Ist die Treiber im Projekt bereits vorhanden wird dies mit einem Warndialog angezeigt.
- 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**

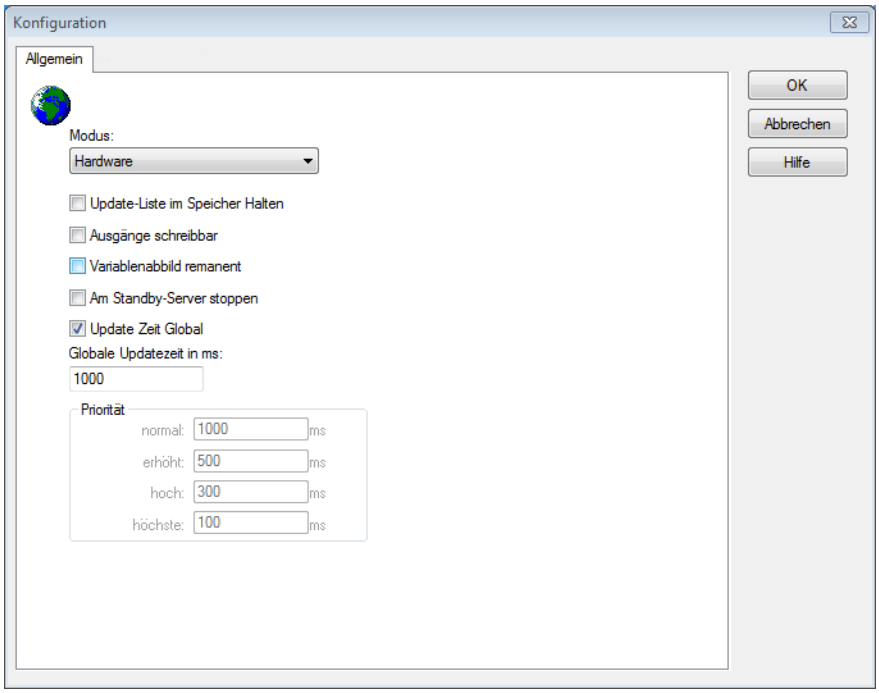
> **Info**
>
> In einem zenon Projekt müssen nur die benötigten Treiber vorhanden sein. Treiber können bei Bedarf zu einem späteren Zeitpunkt hinzugefügt werden.

6.2 Einstellungen im Treiberdialog

Folgende Einstellungen können Sie beim Treiber vornehmen:

6.2.1 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
Modus	Ermöglicht ein Umschalten zwischen Hardware und Simulationsmodus ▸ `Hardware`: Die Verbindung zur Steuerung wird hergestellt. ▸ Simulation - statisch: 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 gesetzen 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. ▸ `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. ▸ `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)
Update-Liste im Speicher Halten	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.
Ausgänge schreibbar	▸ `Aktiv`: Ausgänge können beschrieben werden. ▸ `Inaktiv`: Das Beschreiben der Ausgänge wird unterbunden. Hinweis: Steht nicht für jeden Treiber zur Verfügungen.
Variablenabbild remanent	Diese Option speichert und restauriert den aktuellen Wert, den Zeitstempel und die Status eines Datenpunkts. Grundvoraussetzung: Die Variable muss einen gültigen Wert und

	Zeitstempel besitzen. Das Variablenabbild wird im Modus Hardware gespeichert wenn: ▸ einer der Status S_MERKER_1(0) bis S_MERKER8(7), REVISION(9), AUS(20) oder ERSATZWERT(27) aktiv ist Das Variablenabbild wird immer gespeichert wenn: ▸ die Variable vom Objekttyp **Treibervariable** ist ▸ der Treiber im Simulationsmodus läuft. (nicht programmierte Simulation) Folgende Status werden beim Start der Runtime nicht restauriert: ▸ `SELECT(8)` ▸ `WR-ACK(40)` ▸ `WR-SUC(41)` Der Modus **Simulation - programmiert** beim Treiberstart ist kein Kriterium, um das remanente Variablenabbild zu restaurieren.
Am Standby Server stoppen	Einstellung für Redundanz bei Treibern, die nur eine Kommunikationsverbindung erlauben. Dazu wird der Treiber am Standby Server gestoppt und erst beim Hochstufen wieder gestartet. **Achtung:** Ist diese Option aktiv, ist die lückenlose Archivierung nicht mehr gewährleistet. ▸ `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 **abgeschaltet** (**statusverarbeitung.chm::/24150.htm**), sondern einen leeren Wert. Damit wird verhindert, dass beim Hochstufen zum Server nicht relevante Werte in AML , CEL und Archiv erzeugt werden. Default: `inaktiv` **Hinweis:** Nicht verfügbar, wenn CE Terminal als Datenserver dient. Weitere Informationen dazu erhalten Sie im Handbuch zenon Operator im Kapitel CE Terminal als Datenserver.
Update Zeit Global	Einstellung für globale Update-Zeiten in Millisekunden: ▸ `Aktiv`: Die eingestellte **Globale Update Zeit** wird für alle Variablen im Projekt verwendet. Die bei den Variablen eingestellte Priorität wird nicht verwendet.

	▸ Inaktiv: Die eingestellten Prioritäten werden für die einzelnen Variablen verwendet. **Ausnahmen:** Spontane Treiber ignorieren diese Option. Sie nutzen in der Regel die kürzest mögliche Update Zeit. Details siehe Abschnitt **Update Zeit spontane Treiber**.
Priorität	Hier werden die Pollingzeiten der einzelnen Prioritätsklassen eingestellt. Alle Variablen mit der entsprechenden Priorität werden in der eingestellten Zeit gepollt. Die Zuordnung der Variablen erfolgt separat bei jeder Variablen über die Einstellungen in den Variableneigenschaften. 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. **Achtung:** Prioritätsklassen werden nicht von jedem Treiber unterstützt, z.B. von spontan kommunizierenden zenon Treibern.

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.

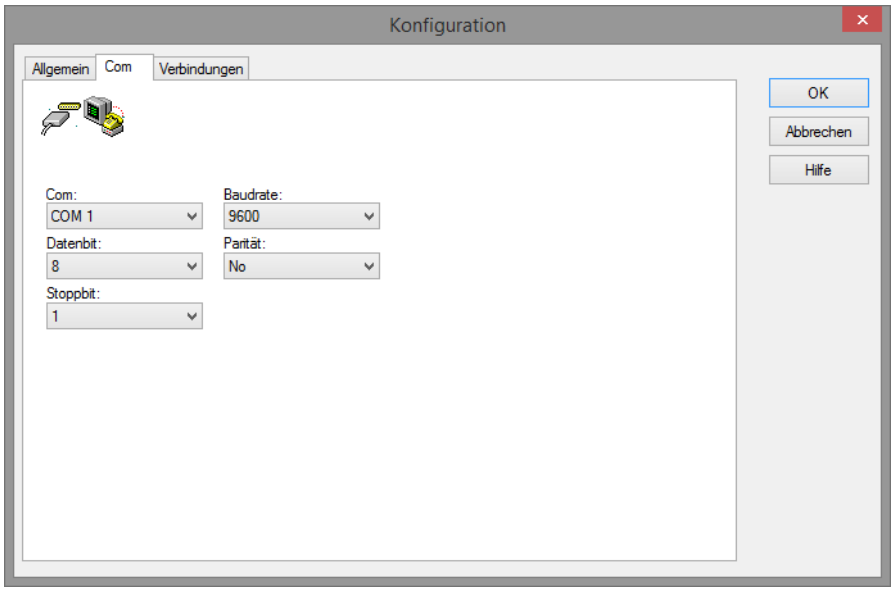
UPDATE ZEIT SPONTANE TREIBER

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**.

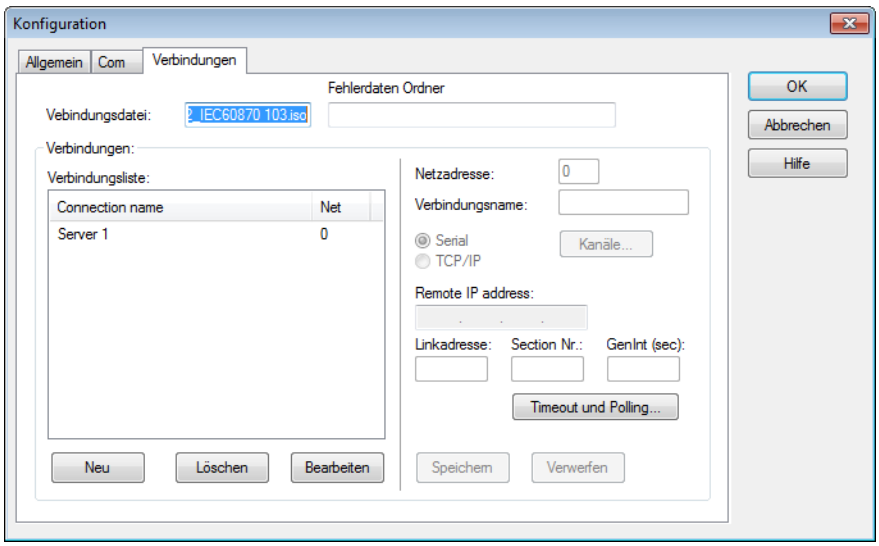
6.2.2 Com

Im seriellen Fall, wählen Sie die passende Kommunikations Einstellungen.



Parameter	Beschreibung
Com	COM-Port 1-16
Baudrate	Anpassen nach Einstellung auf Steuerung
Datenbits	8
Stopbit	1
Parity	Anpassen nach Einstellung auf Steuerung

6.2.3 Verbindungen



Optionen	Beschreibung
Verbindungsdatei	Namen der Datei für Konfigurationsdaten. Diese Datei ist für die Definition der Treiberverbindung notwendig.
Fehlerdaten Ordner	Name des Ordner am Runtime-Rechner, in dem die Dateien mit transferierten Fehlerdaten abgelegt werden sollen.

VERBINDUNGEN

Bereich für die Konfiguration beliebig vieler Steuerungen auf unterschiedliche Hardware Adressen.

Optionen	Beschreibung
Verbindungsliste	Zeigt alle konfigurierten Verbindungen an. Für jede Verbindung wird der Verbindungsname mit der dazugehörigen Netzadresse angezeigt. Für den ausgewählten Verbindungsnamen werden im Bereich neben der Liste die Verbindungsparameter angezeigt.
Neu	Klick auf die Schaltfläche erstellt einen neuen Eintrag in der Liste und ermöglicht es, die Verbindungsparameter einzutragen.
Löschen	Löscht die ausgewählte Verbindung aus der Liste.
Bearbeiten	Ermöglicht die Konfiguration der Optionen für die ausgewählte Verbindung.
Netzadresse	Entspricht der **Netzadresse** bei der Variablendefinition. Wert: `0` - `255`
Verbindungsname	Frei wählbarer Name der Verbindung, z. B. das Kennzeichen der Steuerung.
Serial	▸ `Aktiv`: Verbindung wird seriell hergestellt.
TCP/IP	▸ `Aktiv`: Verbindung wird über TCP/IP hergestellt. **Hinweis:** Der Treiber stellt die Verbindung entsprechend dem **IEC60870 Standard** über den Port `2404` her.
Kanäle	Klick auf die Schaltfläche öffnet den Dialog zum Einstellen der Kanäle (auf Seite 22) für das Comtrade-Format.
Remote IP address	IP-Adresse der Steuerung. **Hinweis:** Der Treiber stellt die Verbindung entsprechend dem **IEC60870 Standard** über den Port `2404` her.
Linkadresse	Linkadresse der Steuerung (des IEC60870-103 Slaves). Jedes Gerät wird auf der Runtime-Seite durch die Netzadresse identifiziert und auf der IEC60870-103 Protokoll Seite durch die Linkadresse. **Hinweis:** Die **Linkadresse** `255` wird als Broadcast Channel verwendet. Wird die **Linkadresse** einer Verbindung auf `255` gesetzt, können die **Kanäle** und die **Section** nicht mehr editiert werden.
Section Nr.	Sektion der Steuerung (des IEC60870-103 Slaves). Ein IEC60870-103 Gerät kann mehrere Sektoren enthalten.
GenInt (sec)	Das General Interrogation Intervall in Sekunden. Der Treiber sendet GI zyklisch nach dieser Einstellung. Empfehlung: Setzen Sie den Wert größer als 15 Minuten (900 s).

Timeout und Polling...	Klick auf die Schaltfläche öffnet den Dialog zur Konfiguration der Timeout-Einstellungen (auf Seite 24). Der Dialog wird im Modus **neu** oder **bearbeiten** zum Bearbeiten geöffnet, sonst nur zur Ansicht.
Speichern	Speichert die Verbindungseinstellungen für den ausgewählten Eintrag.
Verwerfen	Verwirft alle Änderungen und schließt die Optionen ohne zu speichern.

Hinweis: Meldungen des Treibers können mit dem Diagnosis Viewer (main.chm::/12464.htm) ausgelesen werden.

NEUE VERBINDUNG ANLEGEN

1. Klicken Sie auf die Schaltfläche **Neu**.
2. Tragen Sie die Verbindungsdetails ein.
3. Klicken Sie auf **Speichern**.

VERBINDUNG BEARBEITEN

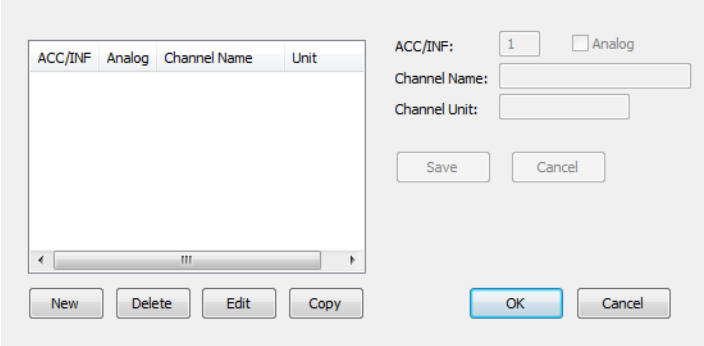
1. Wählen Sie in der Verbindungsliste die gewünschte Verbindung.
2. Klicken Sie auf die Schaltfläche **Bearbeiten**.
3. Ändern Sie die Verbindungsparameter.
4. Schließen Sie mit **Speichern** ab.

VERBINDUNG LÖSCHEN

1. Wählen Sie in der Verbindungsliste die gewünschte Verbindung.
2. Klicken Sie auf die Schaltfläche **Löschen**.
3. Die Verbindung wird aus der Liste gelöscht.

Einstellen der Kanäle

Der IEC870_10332 Treiber schreibt Disturbance Data als Comtrade-Dateien nach dem Standard von 1999. Damit zenon diese Informationen verarbeiten kann, sind zusätzliche Informationen über die Disturbance Kanäle notwendig. Sie legen die Kanäle in folgendem Dialog fest.



Hinweis: Dieser Dialog steht nur in englischer Sprache zur Verfügung.

Parameter	Beschreibung
Listenfeld (linke Seite)	Auflistung aller bereits definierter Kanäle.
New	Aktiviert Konfiguration für neuen Kanal.
Delete	Löscht ausgewählten Kanal aus der Liste.
Edit	Öffnet Konfiguration für ausgewählten Kanal.
Copy	Kopiert Einträge einer anderen Hardware-Adresse.
Konfiguration (rechte Seite)	
ACC/INF	Tragen Sie hier die entsprechenden Kanalnummer zur Identifizierung des entsprechenden Comtrade-Kanals ein. ACC: analog INF: digital
Analog	Durch Aktivierung der Checkbox legen Sie den Kanal als analog fest. Sonst ist der Kanal digital.
Channel Name	Geben Sie einen Namen für den Kanal an. Der Name darf kein Koma enthalten.
Unit	Nur verfügbar, wenn **Analog** ausgewählt. Geben Sie eine Einheit für den Kanal an.
Save	Speichert die Einstellungen für den Kanal.
Cancel	Verwirft ungesicherte Änderungen und bricht die die Bearbeitung des Kanals ab.

Nachdem Sie die gewünschten Änderungen vorgenommen haben, verlassen Sie den Dialog mit **OK**, um die Änderungen zu übernehmen. Ein Klick auf **Cancel** schließt den Dialog ohne die Änderung zu übernehmen.

DATENSPEICHERUNG

- Der Treiber speichert die Disturbance Daten in einer **Comtrade.cfg** und einer **Comtrade.dat** Datei.
- Der Name des Kanals wird bei analogen und digitalen Kanälen in **ch_id** eingesetzt, die Einheit bei analogen in **uu**.
- Die Kanalinformationen werden in der Treiberverbindungsdatei für jede Hardwareadresse abgelegt (xxx ist die jeweilige Hardwareadresse):

  [FETCH_HWxxx]

  CH_COUNT = (Anzahl der Kanäle)

- Für jeden Kanal gibt es eine Section (ccc ist die fortlaufende Nummer der Kanäle 0-(CH_COUNT-1)):

  [HWxxxCHANNELccc]

  CH_ANALOG = (1 Analog, 0 Digital)

  CH_NO = (ACC/Information Number)

  CH_NAME = (Name des Kanals)

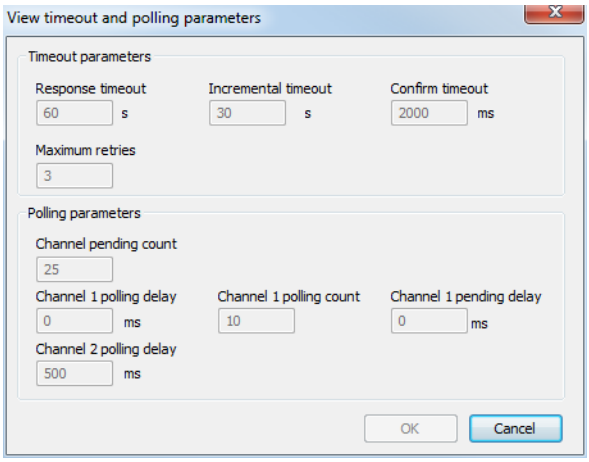
  CH_UNIT = (Einheit des Kanals)

- Empfängt der Treiber einen Kanal, der in der Liste nicht definiert wurde, bleiben die Felder in **Comtrade.cfg** leer.

Timeout und Polling

Die Timeout-Einstellungen werden in diesem Dialog festgelegt. Manche Timeouts werden in Sekunden, andere in Millisekunden konfiguriert.

> **Achtung**
>
> ***Response timeout*** *kann jederzeit individuell angepasst werden.*
>
> *Alle anderen Einstellungen für Timeout und Polling sollten auf den vorgegebenen Werten belassen werden. Änderungen dieser Einstellungen sind Experten vorbehalten und können bei Fehlkonfiguration die Kommunikation beeinträchtigen und zu unerwünschtem Verhalten führen.*



Hinweis: Dieser Dialog steht nur in englischer Sprache zur Verfügung.

Parameter	Beschreibung
Timeout parameters	
Response timeout	Genereller Response Timeout in Sekunden. Default: 60 s
Incremental timeout	Maximale Zeit in Sekunden, die ein Gerät auf Antwort eines anderen Geräts wartet, wenn ein Kommando ansteht. Default: 30 s
Confirm timeout	Maximale Zeit, die auf eine Bestätigung gewartet wird. Default: 2000 ms
Maximum retries	Maximale Anzahl der Verbindungsversuche für serielle Kommunikation. Default: 3
Polling Parameters	Polling-Einstellungen
Channel pending count	Maximale Anzahl der auf einander folgenden Anfragen an ein Gerät. Hat keine Auswirkung, wenn nur ein Channel konfiguriert wurde. Default: 25
Channel 1 polling delay	Verzögerung beim Polling für Kanal 1 in Millisekunden. Default: 0 ms
Channel 1 polling count	Anzahl der Polling-Versuche für Kanal 1. Default: 10
Channel 1 pending delay	Verzögerung für Kanal 1 in Millisekunden. Default: 0 ms
Channel 2 polling delay	Verzögerung beim Polling für Kanal 2 in Millisekunden. Default: 500 ms

DIALOG BEENDEN

Option	Beschreibung
OK	Übernimmt Einstellungen und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.

7. Variablen anlegen

So werden Variablen im zenon Editor angelegt:

7.1 Variablen im Editor anlegen

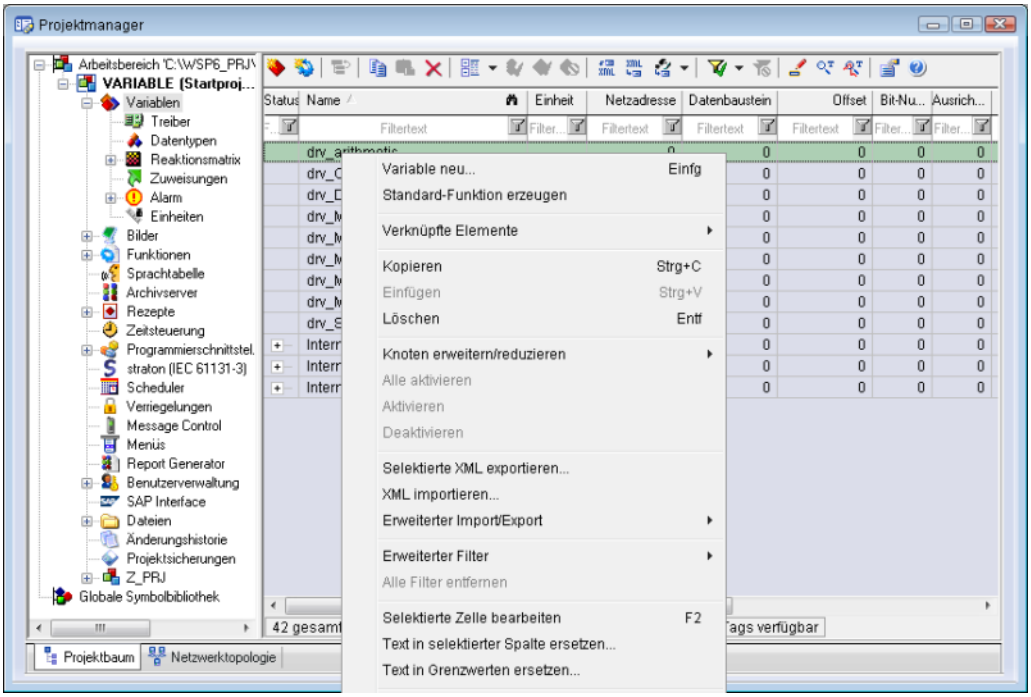
Variablen können angelegt werden:

- 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:

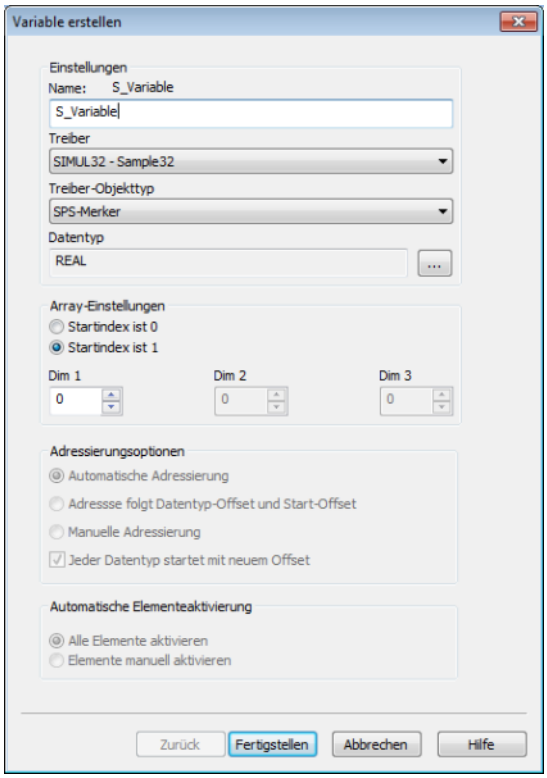
1. Wählen Sie im Knoten **Variablen** im Kontextmenü den Befehl **Variable neu**.



   Der Dialog zur Konfiguration der Variable wird geöffnet.

2. Konfigurieren Sie die Variable.

3. Welche Einstellungen möglich sind, hängt ab vom Typ der Variablen.



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** (**Main.chm::/Intern.chm::/Intern.htm**)) geladen.
Treiberobjekttyp (cti.chm::/28685.htm)	Wählen Sie aus der Dropdownliste den passenden Treiberobjekttyp aus.

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.

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.

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.

7.2 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	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 `Erweiterter Datenbaustein` 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	Nummer des Bits innerhalb des eingestellten Offsets. Mögliche Eingabe: `0` bis `65535`.
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/Treiberobjekttyp	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.

Der Treiber kommuniziert adressbasiert.

Die Variablen werden zugeordnet über:

- die Netzadresse - eingegeben bei Treiberkonfiguration (auf Seite 19) für Verbindung,
- den Function Type und die Information Number - entspricht **FUN** und **INF** aus der Norm
- sowie bei Measurands zusätzlich noch dem Index - entspricht der Reihenfolge von MEA in ASDU <3> oder <9>

Die Variable des Zeitsynchronisation-Kommandos (Treiber Objekttyp TIME SYNC) hat den Function Type 0 und die Information Number 0, ebenso die Variable mit dem Verbindungsstatus (Treiber Objekttyp CONNECTION STATE).

Beispiel

Measurand P von Measurands II (ASDU <9> in monitor direction)

Datentyp: `REAL`

Treiber Objekttyp: MONITORING

Function Type: 128

Information Number: 148

Index: 6

7.3 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.

7.3.1 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
COMMAND	8	--	X	BOOL, INT, UINT, SINT, USINT	Für 'General command' - ASDU<20> in control direction In Norm entspricht DCO (Double command); Empfehlung: SINT, USINT Bei fehlgeschlagenem Senden wird die Variable ein PN-Bit bekommen.
TIME SYNC	67	--	X	BOOL	Für Zeitsynchronisation Kommando - ASDU<6> in control direction Ein an diese Variable (Function Type 0 und Information Number 0) abgesetzter Sollwert 1 wird als Zeitsynchronisation an die Steuerung gesendet. Bei fehlgeschlagenem Senden wird die Variable ein PN-Bit bekommen.
MONITORING	9	X	--	BOOL, REAL, INT, UINT, SINT, USINT	In Norm entspricht DPI (Double-point information); Empfehlung: SINT, USINT oder MVAL (F13); Empfehlung: REAL oder SCL (R32.23); Empfehlung: REAL
MONITORING TRANSIENT	65	X	--	BOOL, INT, UINT, SINT, USINT	Für Indikatoren die nicht in GI enthalten sind. In Norm entspricht DPI (Double-point information); Empfehlung:

					SINT, USINT *) Der kommende Wert wird vom Treiber gleich wieder auf 0 gesetzt.
CONNECTION STATE	66	X	--	BOOL	der Verbindungsstatus (Function Type 0 und Information Number 0) Ist der Wert der Variable 1, so besteht eine laufende Verbindung (wenigstens auf Data-Link Ebene).
DISTURBANCE DATA	64	X	X	STRING	Um Disturbance Data Dateien abzurufen müssen zwei Variablen mit gleichen Function Type projektiert werden. Variable mit Information Number 0 ist die Befehls- und Statusvariable, Information Number 1 ist die Directory Variable. Der Index wird bei diesen Variablen ignoriert.
Kommunikationsdetails	35	X	X	BOOL, SINT, USINT, INT, UINT, DINT, UDINT, REAL, STRING	Variablen für die statische Analyse der Kommunikation; wird zwischen Treiber und Runtime übertragen (nicht zur SPS). **Hinweis**: Die Adressierung und das Verhalten ist bei den meisten zenon Treibern gleich. Weitere Informationen dazu finden Sie im Kapitel Kommunikationsdetails (Treibervariablen) (auf Seite 42).
Generic Data	69	X	X	STRING	
Generic Data Trigger	68	X	X	INT, UINT	

Legende:

x: wird unterstützt

--: wird nicht unterstützt

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

GENERIC DATA TRIGGERED UND GENERIC DATA

Für jede Generic Data Variable muss eine korrespondierende Generic Data Trigger Variable existieren.

Die Adressen der Variablen werden wie folgt gemappt:

- **Function Type:** `kod` (üblicherweise 1)
- **Information Number:** `0` (wird nicht verwendet)
- **Index:** `gin`

Der Wert der Triggervariable, der geschrieben wird entscheidet über die durchzuführende Aktion:

- `244`: Get Generic Data – die Variable mit `gin/kod` wird angefordert
- `245`: Generic Data GI – eine Generalabfrage für `Generic Data` wird abgeschickt
- `248`: Write GenericData – die Variable mit `gin/kod` wird geschrieben
- `250`: Write and Execute – die Variable mit `gin/kod` wird geschrieben und ein **execute** angestoßen

Die Daten der Generic Data Variablen werden als Hexdump in einem String ausgegeben.

7.3.2 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	INT	1
-	UDINT	4
-	DINT	3
-	ULINT	27
-	LINT	26
REAL	REAL	5
-	LREAL	6
STRING	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.

7.4 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).

7.4.1 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.

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** (**main.chm::/13046.htm**).*

7.4.2 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 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 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 **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)
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, 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
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.

7.5 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** (auf der Installationsmedium im Ordner `\Predefined\Variables`) verfügbar und können von dort importiert werden.

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 werden:

- Variablen für Modem-Informationen nur von modemfähigen Treibern unterstützt
- Treibervariablen für den Polling-Zyklus nur für rein pollenden Treibern
- verbindungsbezogene Informationen wie ErrorMSG 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 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.
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 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.
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 **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.
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.

8. Treiberspezifische Funktionen

Dieser Treiber unterstützt folgende Funktionen:

VERBINDUNGSSTATUS

Über eine BOOL-Variable vom Treiber Objekttyp **CONNECTION STATE** kann der aktuelle Verbindungsstatus abgefragt werden. Die Function Type und Information Number dieser Variablen müssen 0 sein. Ist der Wert dieser Variable 1, so besteht eine laufende Verbindung (zumindest auf Data-Link Ebene).

GENERALABFRAGE (GENERAL INTERROGATION - GI)

Der Treiber sendet GI (ASDU <7> in control direction) zyklisch je nach Verbindungeinstellung **GenInt** in der Treiberkonfiguration (auf Seite 19). Auch gleich nach Verbindungwiederherstellung (falls die ausgefallen war) wird ein GI ausgeführt.

Der Treiber nimmt auch kommende Werte von Variablen an, die in der Norm als nicht in GI vorhanden gekennzeichnet sind, falls die Steuerung solche doch kommuniziert.

Eine Generalabfrage kann auch als Broadcast abgesetzt werden. Dafür wird die Linkadresse (auf Seite 19) `255` als Broadcast Channel verwendet. Dabei gilt:

- General Interrogation auf einer Verbindung wird nur dann abgesetzt, wenn das Intervall dafür ungleich `0` ist (Einstellung **GenInt** (auf Seite 19)). Ausnahme: Initiale GI, diese wird immer gesendet.
- General Interrogation auf der Verbindung mit Linkadresse `255` wird als Broadcast GI gesendet, wenn das Intervall ungleich `0` ist (Einstellung **GenInt** (auf Seite 19)).

ZEITSYNCHRONIZATION

Der Treiber sendet Zeitsynchronisation Befehl (ASDU <6> in control direction) automatisch bei jedem Treiberstart, also bei Start der zenon Runtime und nach Stopp/Start des Treibers (mit zenon Funktionen - Treiber Kommandos: Treiber Stoppen und Treiber Starten).

Es kann nachträglich erneut einer Uhrzeitsynchronisation ausgelöst werden. Dazu ist eine Variable vom Treiber Objekttyp TIME SYNC zu beschreiben. Die Function Type und Information Number dieser Variablen müssen 0 sein. Bekommt die Variable den Wert 1, wird die aktuelle Uhrzeit des PCs zur Steuerung gesendet. Konnte die Uhrzeitsynchronisation nicht korrekt ausgeführt werden, so erhält diese Variable PN-Bit.

Zeitsynchronisation kann auch als Broadcast abgesetzt werden. Dafür wird die Linkadresse (auf Seite 19) `255` als Broadcast Channel verwendet. Dabei gilt:

- Für die Zeitsynchronisation muss eine Variable vom Typ `TIME SYNC` angelegt werden.
- Die Variable muss die dieselbe Netzadresse haben, die für die Broadcast Linkadresse `255` verwendet wird.
- Wird diese Variable gesetzt, dann wird ein Broadcast Zeitsynchronisation gesendet.

KOMMANDOS (GENERAL COMMAND)

Um General command an die Steuerung zu senden (ASDU <20> in control direction), muss eine Variable von Treiber Objekttyp COMMAND (auf Seite 29) mit dem entsprechenden Function Type und Information Number projektiert werden. Ein an diese Variable abgesetzter Sollwert (DCO: 1-off, 2-on) wird als General Command an die Steuerung gesendet. Bei fehlgeschlagenem Senden des Kommandos wird das PN-Bit der Variable auf 1 gesetzt.

ÜBERTRAGUNGSURSACHE

Die COT in den Status der Variable ist verfügbar und kann z. B. über ein Rema ausgewertet werden.

Die Rückmeldevariablen bekommen die COT von Steuerung und die wird von Treiber an Leitsystem weitergegeben, z. B. in der Norm die COT 1 bedeutet 'spontaneous' ,COT 9 - 'general interrogation'.

Die Übertragungsursache des Kommandos wird jedoch für Leitsystem von Treiber selbst gesetzt. In Richtung Steuerung wird natürlich die COT nach der Norm 60870-103 gesendet aber für Leitsystem, um die Kompatibilität mit Befehlsgabe zu sichern, für Übertragungsursache Ablauf des Kommandos verwendet der Treiber die COT von der Norm 61870-101/104 (z. B. COT Wert 7 - COT_actcon). Somit wird Sollwertvorgabe mit der Laufzeitüberwachung über Übertragungsursache unterstützt.

INVALID STATUS

Der Treiber setzt das Invalid Bit nach dem entsprechenden Time-out, wenn eine Störung in der Kommunikation (TCP/IP oder Seriell) mit dem Treiber vorliegt.

INVALID wird auch für Measurands Werte gesetzt, in deren Struktur (MEA) der Quality descriptor den Wert ER (MVAL invalid) aktiviert ist.

OV-BIT STATUS

OV-Bit wird auch für Measurands Werte gesetzt, in deren Struktur (MEA) der Quality descriptor den Wert OV (Overflow) aktiviert ist.

VARIABLEN, DIE NICHT IN GI VORHANDEN SIND

Manche Steuerungen nicht für alle Indikatoren (Datenpunkte) senden eine Zurücksetzung des Wertes (von ON zu OFF). Auch nicht alle Indikatoren standardweise in GI vorhanden sind (z. B. Statusindikator 'LED reset').

Für solche Variablen ist Treiber Objekttyp MONITORING TRANSIENT (auf Seite 29) empfohlen. Der kommende Wert wird vom Treiber gleich wieder auf 0 gesetzt und kann durch Anlegen von Alarm oder Funktion für Grenzwertverletzung bei Grenzwert 1 und/oder bei Grenzwert 2 ausgewertet werden (somit kann ein Zähler für Wert OFF o. ON gebaut werden).

Falls eine Variable von Treiber Objekttyp MONITORING wird auf Generalanfrage (GI) nicht geantwortet (z. B. Time-trigged measurand - ASDU<4>) bekommt sie Wert 0 und der Treiber setzt auch das SB-Bit (Substituted).

VARIABLEN ZEITSTEMPEL

Monitoring-Variablen:

- Time-tagged message (ASDU<1>)
- Time-tagged message with relative time (ASDU<2>)
- Time-tagged measurands with relative time (ASDU<4>)

kommen von IEC 60870-5-103 Steuerungen signiert mit CP32Time2a Zeitstempel. Der Treiber verwendet diese Zeit für den Zeitstempel der Variable und setzt das Statusbit T_EXTERN (Echtzeit extern). Alle Module des Leitsystems verwenden nun diesen Zeitstempel, z. B. Archiv, Alarmliste...

Die Zähler RET und FAN von ASDUs <2> u.<4> werden nicht vom Treiber berücksichtigt.

DISTURBANCE DATA ABRUFEN

Die Disturbance Data Dateien werden im angegeben Verzeichnis (in der Treiberkonfiguration) in Unterverzeichnisse pro Netzadresse gespeichert (d.h. bei Netzadresse 1 gibt es ein Unterverzeichnis '1', in dem die Dateien abgelegt werden).

Um Disturbance Data Dateien für einen Function Type abzurufen bzw. die verfügbaren Daten anzusehen müssen zwei String Variablen projektiert werden von Treiber Objekttyp DISTURBANCE DATA (auf Seite 29) und entsprechenden Function Type.
Die Variable mit Information Number 0 ist die Befehls- und Statusvariable, mit Information Number 1 ist die Directory Variable. Der Index wird bei diesen Variablen ignoriert.

Wird bei Information Number 0 der Wert der Variable mit 'DIR' gesetzt, so werden die Directory Informationen aktualisiert und in der Variable mit Information Number 1 angezeigt. Nach Ausführung der Operation bekommt die Variable mit Information Number 0 den Wert '`DIR OK`' oder '`DIR ERROR`'.

Mit absetzen von '`GET [FaultNo]`' auf die Variable mit Information Number 0, wird die entsprechende Disturbance Data Datei angefordert und im projektierten Verzeichnis als Comtrade Dateien '[FktType]-[FaultNo].cfg' und '[FktType]-[FaultNo].dat' abgelegt (z. B. '128-3.cfg, 128-3.dat'). Nach Ausführung der Operation bekommt die Variable mit Information Number 0 den Wert '`GET OK`' oder '`GET ERROR`'.

SPEZIFISCHE LOG EINTRÄGE

- '`Callback Status: Variable: , Status: I870CHNL_RESP_STATUS_TIMEOUT`'

Der Treiber konnte die Kommunikation mit Steuerung nicht herzustellen, die Kommunikation wurde unterbrochen oder die Steuerung antwortet nicht in vorgeseher Zeit.

- `'LOG:ConvertTime: Invalid TimeStamp'`

Der von der Steuerung kommende Zeitstempel entspricht nicht dem CP32Time2a Format.

9. 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

Achtung: Die zenon Funktion **Treiberkommandos** ist nicht ident mit Treiberkommandos, die bei Energy-Treibern zur Runtime ausgeführt werden können!

> **Info**
>
> *Dieses Kapitel beschreibt Standardfunktionalitäten, die für die meisten zenon Treiber gültig sind.*
> *Aber 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.*

PROJEKTIERUNG DER FUNKTION

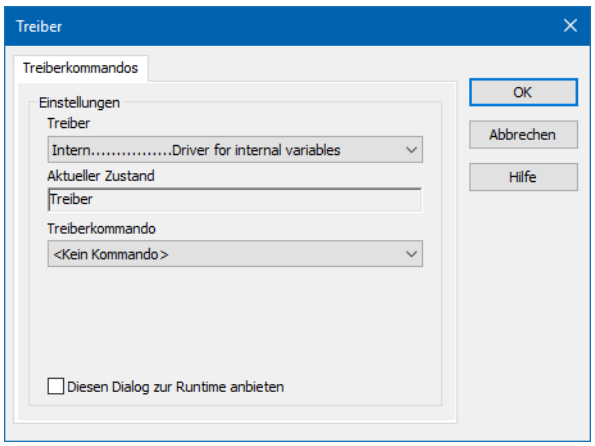
Die Projektierung erfolgt über die Funktion **Treiberkommandos**.
Dazu:

1. Legen Sie im zenon Editor eine neue Funktion an.
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 zur 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, in aktuellen Versionen ohne Funktion.
Treiberkommando	Dropdownliste zur Auswahl des Kommandos:
<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.
`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 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).
`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 bei `0`.
`Treiber in Simulationsmodus statisch`	Treiber wird in den zählenden Simulationsmodus gesetzt. Alle Werte werden mit `0` initialisiert.
`Treiber in Simulationsmodus programmiert`	Treiber wird in den zählenden Simulationsmodus gesetzt. 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.
Diesen Dialog zur Runtime anbieten	Dialog wird zur Runtime für Änderungen angeboten.

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, 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.

10. Interoperability List

This companion standard presents sets of parameters and alternatives from which subsets must be selected to implement particular telecontrol systems. Certain parameter values, such as the choice of 'structured' or'unstructured' fields of the informationobject address of ASDUs represent mutually exclusive alternatives. This means that only one value of the defined parameters is admitted per system. Other parameters, such as the listed set of different process information incommand and in monitor direction allow the specification of the complete set orsubsets, as appropriate for given applications. This clause summarizes theparameters of the previous clauses to facilitate a suitable selection for aspecific application. If a system is composed of equipment stemming from different manufacturers, it is necessary that all partners agree on the selected parameters.

The interoperability list is defined as in IEC 60870-5-101 and extended with parameters used in this standard. The text descriptions of parameters which are not applicable to this companion standard are strike-through (corresponding check box is marked black).

NOTE In addition, the full specification of a system may require individual selection of certain parameters for certain parts of the system, such as the individual selection of scaling factors for individually addressable measured values.

The selected parameters should bemarked in the white boxes as follows:

`[ ]` Function or ASDU is not used
`[ X ]` Function or ASDU is used as standardized (default)
`[ R ]` Function or ASDU is used in reverse mode
`[ B ]` Function or ASDU is used in standard and reverse mode

The possible selection (blank, `X` , `R` , or `B)` is specified for each specific clause or parameter.

A black check box indicates that the option cannot be selected in this companion standard.

1. PHYSICAL LAYER

ELECTRICAL INTERFACE

[X] EIA RS-485

[X] Number of loads 32

OPTICAL INTERFACE

[X] Glass Fibre

[X] Plaxtic Fibre

[X] F-SMA type connector

[X] BFOC/2,5 type connector

TRANSMISSION SPEED

[X] 9600 bit/s

[X] 12900 bit/s

2. LINK LAYER

There are no choices for the Link Layer.

3. APPLICATION LAYER

TRANSMISSION MODE FOR APPLICATION DATA

Mode 1

COMMON ADDRESS OF ASDU

[X] One COMMON ADDRESS of ASDU (identical with station address)

[ ] More than one COMMON ADDRESS of ASDU

SELECTION OF STANDARD INFORMATION NUMBERS IN MONITOR DIRECTION

System functions in monitor direction		
	Inf	**Semantics**
[X]	<0>	End of general interrogation
[X]	<0>	Time synchronization
[X]	<2>	Reset FCB
[X]	<3>	Reset CU
[X]	<4>	Start/restart
[X]	<5>	Power on

Status indications in monitor direction		
	Inf	**Semantics**
[X]	<16>	Auto-recloser active
[X]	<17>	Teleprotection active
[X]	<18>	Protection active
[X]	<19>	LED reset
[X]	<20>	Monitor direction blocked
[X]	<21>	Test mode
[X]	<22>	Local parameter setting
[X]	<23>	Characteristic 1
[X]	<24>	Characteristic 2
[X]	<25>	Characteristic 3
[X]	<26>	Characteristic 4
[X]	<27>	Auxiliary input 1
[X]	<28>	Auxiliary input 2
[X]	<29>	Auxiliary input 3
[X]	<30>	Auxiliary input 4

Supervision indications in monitor direction		
	Inf	**Semantics**

[X]	<32>	Measurand supervision I
[X]	<33>	Measurand supervision V
[X]	<35>	Phase sequence supervision
[X]	<36>	Trip circuit supervision
[X]	<37>	I>> back-up operation
[X]	<38>	VT fuse failure
[X]	<39>	Teleprotection disturbed
[X]	<46>	Group warning
[X]	<47>	Group alarm

Earth fault indications in monitor direction		
	Inf	**Semantics**
[X]	<48>	Earth fault L_1
[X]	<49>	Earth fault L_2
[X]	<50>	Earth fault L_3
[X]	<51>	Earth fault forward, i.e. line
[X]	<52>	Earth fault reverse, i.e. busbar

Fault indications in monitor direction		
	Inf	**Semantics**

[X]	<64>	Start /pick-up L_1
[X]	<65>	Start /pick-up L_2
[X]	<66>	Start /pick-up L_3
[X]	<67>	Start /pick-up N
[X]	<68>	General trip
[X]	<69>	Trip L_1
[X]	<70>	Trip L_2
[X]	<71>	Trip L_3
[X]	<72>	Trip I>> (back-up operation)
[X]	<73>	Fault location X in ohms
[X]	<74>	Fault forward/line
[X]	<75>	Fault reverse/busbar
[X]	<76>	Teleprotection signal transmitted
[X]	<77>	Teleprotection signal received
[X]	<78>	Zone 1
[X]	<79>	Zone 2
[X]	<80>	Zone 3
[X]	<81>	Zone 4
[X]	<82>	Zone 5
[X]	<83>	Zone 6

[X]	<84>	General start/pick-up
[X]	<85>	Breaker failure
[X]	<86>	Trip measuring system L_1
[X]	<87>	Trip measuring system L_2
[X]	<88>	Trip measuring system L_3
[X]	<89>	Trip measuring system E
[X]	<90>	Trip I>
[X]	<91>	Trip I>>
[X]	<92>	Trip IN>
[X]	<93>	Trip IN>>

Auto-reclosure indications in monitor direction		
	Inf	**Semantics**
[X]	<128>	CB 'on' by AR
[X]	<129>	CB 'on' by long-time AR
[X]	<130>	AR blocked

Measurands in monitor direction		
	Inf	**Semantics**
[X]	<144>	Measurand I
[X]	<145>	Measurands I, V
[X]	<146>	Measurands I, V, P, Q
[X]	<147>	Measurands I_N , V_{EN}
[X]	<148>	Measurands $I_{L1,2,3}$, $V_{L1,2,3}$, P, Q, f

Generic functions in monitor direction		
	Inf	**Semantics**

[ ]	<240>	Read headings of all defined groups
[ ]	<241>	Read values or attributes of all entries of one group
[ ]	<243>	Read directory of a single entry
[ ]	<244>	Read value or attribute of a single entry
[ ]	<245>	End of general interrogation of generic data
[ ]	<249>	Write entry with confirmation
[ ]	<250>	Write entry with execution
[ ]	<251>	Write entry aborted

SELECTION OF STANDARD INFORMATION NUMBERS IN CONTROL DIRECTION

System functions in control direction		
	Inf	**Semantics**
[X]	<0>	Initiation of general interrogation
[X]	<0>	Time synchronization

General commands in control direction		
	Inf	**Semantics**
[X]	<16>	Auto-recloser on/off
[X]	<17>	Teleprotection on/off
[X]	<18>	Protection on/off
[X]	<19>	LED reset
[X]	<23>	Activate characteristic 1
[X]	<24>	Activate characteristic 2
[X]	<25>	Activate characteristic 3
[X]	<26>	Activate characteristic 4

Generic functions in control direction		
	Inf	**Semantics**

[ ]	<240>	Read headings of all defined groups
[ ]	<241>	Read values or attributes of all entries of one group
[ ]	<243>	Read directory of a single entry
[ ]	<244>	Read value or attribute of a single entry
[ ]	<245>	General interrogation of generic data
[ ]	<248>	Write entry
[ ]	<249>	Write entry with confirmation
[ ]	<250>	Write entry with execution
[ ]	<251>	Write entry abort

BASIC APPLICATION FUNCTIONS

[X]	Test mode
[X]	Blocking of monitor direction
[X]	Disturbance data
[ ]	Generic services
[ ]	Private data

MISCELLANEOUS

Measurands are transmitted with ASDU 3 as well as with ASDU 9. As defined in 7.2.6.8, the maximum MVAL can either

be 1,2 or 2,4 times the rated value. No different rating shall be used in ASDU 3 and ASDU 9, i.e., for each measurand

there is only one choice.

Measurand	Max. MVAL	= rated value times
Current L_1	1,2	or 2,4
Current L_2	[X]	[X]
Current L_3	[X]	[X]
Voltage L_{1-E}	[X]	[X]
Voltage L_{2-E}	[X]	[X]
Voltage L_{3-E}	[X]	[X]
Active power P	[X]	[X]
Reactive power Q	[X]	[X]
Frequency f	[X]	[X]
Voltage L_1 - L_2	[X]	[X]

11. Fehleranalyse

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

11.1 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 (main.chm::/12464.htm), das mit zenon mitinstalliert wird. Sie finden es unter *Start/Alle Programme/zenon/Tools 8.00 -> 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 (main.chm::/12464.htm).

11.2 Checkliste

- TCP/IP:
  • Ist das Gerät, mit dem versucht wird eine Kommunikation herzustellen, an das Stromversorgungsnetz angeschlossen ?
  • Sind der PC bzw. das Gerät an das Netzwerk angeschlossen ?
  • Ist das TCP/IP Protokoll installiert?
  • Ist die IP Adresse von einer anderen Anwendung belegt?
- Seriell:
  • Ist der COM Port von einer anderen Anwendung belegt, bzw. stimmen seine Einstellungen ?
  • Ist das Gerät (die Steuerung), mit dem versucht wird eine Kommunikation herzustellen, an das Stromversorgungsnetz angeschlossen ?
  • Ist das Verbindungskabel zwischen Steuerung und PC bzw. IPC korrekt angeschlossen ?

- Allgemein
  - Wurde mit dem Diagnosis Viewer bereits analysiert, welche Fehler sind aufgetreten?
  - Sind die Treibereinstellungen (auf Seite 19) (Verbindungeinstellungen) korrekt, z. B. Linkadresse und Sektion Nr.?
  - Stimmt die Variablenadressierung (auf Seite 29), z. B. entspricht die Netzadresse der in der Treibereinstellung, sind die Indikatoren/Datenpunkte mit den eingegebenen FUN (Function Type) und INF (Information Number) in Steuerung vorhanden?
  - Wurde das Zeitinterval der General Interrogation (GI) lange genug gewählt (Empfehlung als Mindestinterval 15 Minuten)?