



**COPADATA**  
do it your way

# zenon Treiber Handbuch

## OMR\_FINS

v.7.00





© 2012 Ing. Punzenberger COPA-DATA GmbH

Alle Rechte vorbehalten.

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

# Inhalt

<b>1. Willkommen bei der COPA-DATA Hilfe .....</b>	<b>4</b>
<b>2. OMR_FINS.....</b>	<b>4</b>
<b>3. OMR_FINS - Datenblatt.....</b>	<b>4</b>
<b>4. Treiber-Historie .....</b>	<b>6</b>
<b>5. Voraussetzungen .....</b>	<b>6</b>
5.1 PC .....	7
<b>6. Konfiguration .....</b>	<b>7</b>
6.1 Allgemein .....	8
6.2 Conn. TCP/IP .....	11
<b>7. Adressierung .....</b>	<b>14</b>
<b>8. Treiberobjekte und Datentypen .....</b>	<b>15</b>
<b>9. Zuordnung der Datentypen .....</b>	<b>15</b>
<b>10. Treiberspezifische Funktionen.....</b>	<b>16</b>
<b>11. Treiberkommandos .....</b>	<b>19</b>
<b>12. Test.....</b>	<b>21</b>
<b>13. Fehleranalyse .....</b>	<b>22</b>
13.1 Analysetool .....	22
13.2 Fehlernummern .....	23
13.3 Checkliste .....	24

# 1. Willkommen bei der COPA-DATA Hilfe

## **ALLGEMEINE HILFE**

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

## **PROJEKTUNTERSTÜTZUNG**

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

## **LIZENZEN UND MODULE**

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

## 2. OMR\_FINS

## 3. OMR\_FINS - Datenblatt

<b>Allgemein:</b>	
Treiberdateiname	OMR_FINS.exe
Treiberbezeichnung	Omron FINS Treiber
Steuerungs-Typen	OMRON SYSMAC
Steuerungs-Hersteller	Omron;

<b>Treiber unterstützt:</b>	
Protokoll	Omron FINS;
Adressierung: Adress-basiert	x
Adressierung: Namens-basiert	-
Kommunikation spontan	-
Kommunikation pollend	x
Online Browsing	-
Offline Browsing	-
Echtzeitfähig	-
Blockwrite	x
Modemfähig	-
Serielles Logging	x
RDA numerisch	-
RDA String	-

<b>Voraussetzungen:</b>	
Hardware PC	RS 232 serielle Schnittstelle, Kabeltyp: SYSMAC WAY; Standard Netzwerkkarte
Software PC	-
Hardware Steuerung	-
Software Steuerung	-
Benötigt v-dll	-

<b>Plattformen:</b>	
Betriebssysteme	Windows CE 5.0, CE 6.0; Windows XP, Vista, 7, Server 2003, Server 2008/R2;
CE Plattformen	x86; ARM; Pocket-PC;

## 4. Treiber-Historie

<b>Datum</b>	<b>Treiberversion</b>	<b>Änderung</b>
07.07.08	1300	Treiberdokumentation wurde neu erstellt

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

## 6.1 Allgemein

Konfiguration ☰

Algemein

 Modus:  
Hardware

Update-Liste im Speicher Halten

Ausgänge schreibbar

Variablenabbild remanent

Am Standby-Server stoppen

Update Zeit Global

Globale Updatezeit in ms:  
1000

Priontät

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

OK

Abbrechen

Hilfe



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

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

	wird nicht verwendet. Inaktiv: Die eingestellten Prioritäten werden für die einzelnen Variablen verwendet.
Priorität	Hier werden die Pollingzeiten der einzelnen Prioritäten eingestellt. Alle Variablen mit der entsprechenden Priorität werden in der eingestellten Zeit gepollt. Die Zuordnung zu den Variablen erfolgt separat bei jeder Variablen über die Einstellungen in den Variableneigenschaften. Mit den Prioritäten kann die Kommunikation der einzelnen Variablen auf die Wichtigkeit bzw. benötigte Aktualität abgestuft werden. Daraus ergibt sich eine verbesserte Verteilung der Kommunikationslast.

## UPDATE ZEIT ZYKLISCHE TREIBER

Für zyklische Treiber gilt:

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

## 6.2 Conn. TCP/IP

### KONFIGURATIONSDATEI / ZUORDNUNGSDATEI

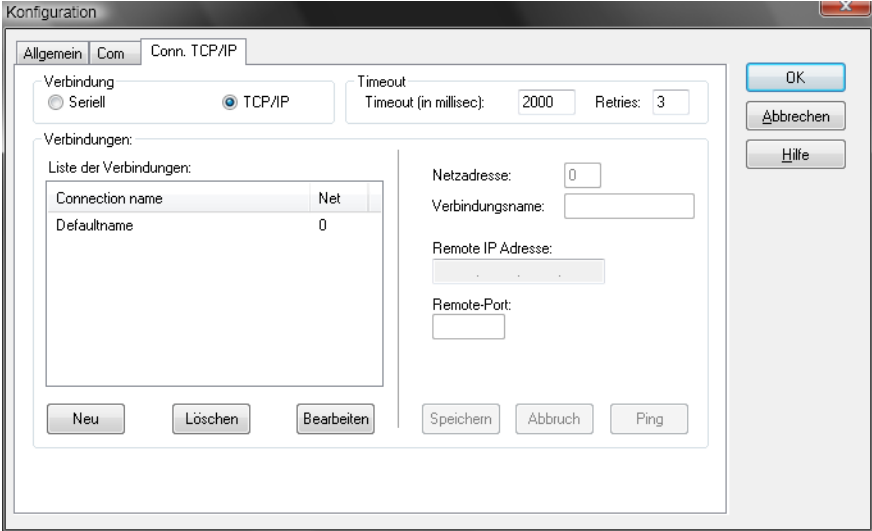
Der Filename der Zuordnungsdatei für TCP Verbindungen ist Conf\_[Treiberbezeichnung].tcp. Sie enthält folgende Sektion für jede konfigurierte HW-Adresse (xx ist die HW-Adresse):

[FETCHHWxx]

CONN\_NAME – Name der Verbindung

REMOTE\_IP – IP-Adresse der Steuerung

REMOTE\_PORT – Port der Steuerung



Parameter	Beschreibung
<b>Verbindung</b>	Auswahl der Verbindungsart.
Seriell	Aktiv: Serielle Verbindung wird benutzt.
TCP/IP	Aktiv: TCP/IP-Verbindung wird benutzt.
Timeout	Wartezeit für Verbindungsaufbau in Millisekunden.
Retries	Anzahl der Wiederholungen bei erfolgreichem Verbindungsaufbau.
<b>Verbindungen</b>	Einstellungen der Verbindungen.
<b>Liste der Verbindungen</b>	Liste aller definierten Verbindungen zu Steuerungen.
Netzadresse	Entspricht der Eigenschaft Netzadresse bei der Variablenkonfiguration.
Verbindungsname	Frei wählbarer Name.
Remote IP Adresse	Adresse der Steuerung.
Remote Port	Port-Adresse der Steuerung. Details finden Sie im Handbuch Ihrer Steuerung.
<b>Neu</b>	Erstellt eine neue Verbindung.
<b>Löschen</b>	Löscht markierte Verbindung aus Liste.
<b>Bearbeiten</b>	Öffnet markierte Verbindung zum Bearbeiten.
<b>Speichern</b>	Übernimmt alle Änderungen für bearbeitete Verbindung und schließt Editiermöglichkeit.
<b>Abbruch</b>	Verwirft alle Änderungen für bearbeitete Verbindung und schließt Editiermöglichkeit.
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.
OK	Änderungen im Dialog übernehmen und Dialog schließen.  Nur verfügbar, wenn keine Verbindung im Zustand "bearbeiten" ist.
<b>Abbrechen</b>	Verwirft alle Änderungen und schließt Dialog.

Hilfe	Öffnet Online-Hilfe.
-------	----------------------

 **Info**

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

## 7. Adressierung

### 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

### TREIBER OBJEKTTYPEN UND UNTERSTÜTZTE IEC DATENTYPEN FÜR PROZESSVARIABLEN IM LEITSYSTEM

Treiber Objekttypen	Kanaltyp	Untersützte Datentypen (Datenart)	Lesen	Schreiben	Kommentar
IR Area	64	BOOL, INT, UINT	J	J	
HR Area	65	BOOL, INT, DINT, REAL, UINT, UDINT, STRING	J	J	
AR Area	66	BOOL, INT, UINT	J	J	
TIM/CNT	68	BOOL	J	J	
DM Area	70	BOOL, INT, DINT, REAL, UINT, UDINT, STRING	J	J	
CIO Area	71	BOOL, INT, DINT, REAL, UINT, UDINT, STRING	J	J	
WORK Area	72	BOOL, INT, DINT, REAL, UINT, UDINT, STRING	J	J	

## 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
U16	UINT
U32	UDINT
REAL	REAL
Boolean	BOOL

**Datenart:** Die Eigenschaft `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:

### INI EINTRÄGE

#### ZENON6.INI

Blockwrite

Durch einen Eintrag in der `zenon6.ini` kann Blockwrite aktiviert werden. Um dies zu aktivieren muss eine Sektion `[OMR_FINS]` mit dem Eintrag `BLOCKWRITE=1` angelegt werden.

#### PROJECT.INI

Serielles Logging



Um diese Option zu aktivieren ist in der Projekt-INI die Sektion [RS232LOG] einzufügen, darunter der Eintrag LOGCOMx=0 oder 1 zu machen. X steht für die Nummer der gewählten Schnittstelle und durch diese zu ersetzen.

Bei 0 wird das Log abgeschaltet, 1 schaltet das Log ein. Bei eingeschaltetem Logging wird eine Datei LOG\_COMxxx.TXT im Verzeichnis vom Treiber erstellt. X wird durch die Nummer der gewählten Schnittstelle ersetzt.

Beispiel für die Aktivierung der Protokollierung für COM 1:

```
[RS232LOG]
```

```
LOGCOM1=1
```



### Info

*Es wird eindringlich empfohlen die Protokollierung nur bei Problemen kurzfristig zu aktivieren. Bei aktiver Protokollierung benötigt diese eine erhebliche Rechnerleistung. Des weiteren benötigt die LOG-Datei innerhalb kürzester Zeit einen erheblichen Speicherplatz.*

## FEHLERDATEI

Der Treiber unterstützt die übliche Fehlerdatei, ab zenon 6.20 ist das zentrale logging möglich.

## ERWEITERTE FEHLERDATEI

Der Treiber unterstützt kein erweitertes logging

## SERIELLES LOGGING

Um diese Option zu aktivieren ist in der Projekt-INI eine Sektion RS232LOG einzufügen. Bei dieser ist dann ein

Eintrag LOGCOMx=0 oder 1 zu machen. X ist durch die Nummer der gewählten Schnittstelle zu ersetzen. Bei 0 wird

das Log abgeschaltet, 1 schaltet das Log ein. Bei eingeschaltetem Protokoll wird eine Datei LOG\_COMxxx.TXT im Verzeichnis vom Treiber erstellt. X wird durch die Nummer der gewählten Schnittstelle ersetzt.

Beispiel für die Aktivierung der Protokollierung für COM 1:

```
[RS232LOG]
```

```
LOGCOM1=1
```

Es wird eindringlich empfohlen die Protokollierung nur bei Problemen kurzfristig zu aktivieren. Bei aktiver Protokollierung benötigt diese eine erhebliche Rechnerleistung. Des weiteren benötigt die LOG-Datei innerhalb kürzester Zeit einen erheblichen Speicherplatz.

### **BLOCKWRITE**

Durch einen Eintrag in der zenon6.ini kann Blockwrite aktiviert werden. Um dies zu aktivieren muss eine Sektion [OMR\_FINS] mit dem Eintrag BLOCKWRITE=1 angelegt werden.

### **REDUNDANZ**

Ja

### **RDA**

-

### **ECHTZEITSTEMPELUNG**

Nein

### **BROWSEN**

Wird vom Treiber nicht unterstützt.

### **FEHLERWARTEZEIT**

Keine Einstellung möglich.

## ZUGRIFFSVERFAHREN

## POLLING

Zyklische Abfrage der Variablen, des Treibers.

## SPONTAN

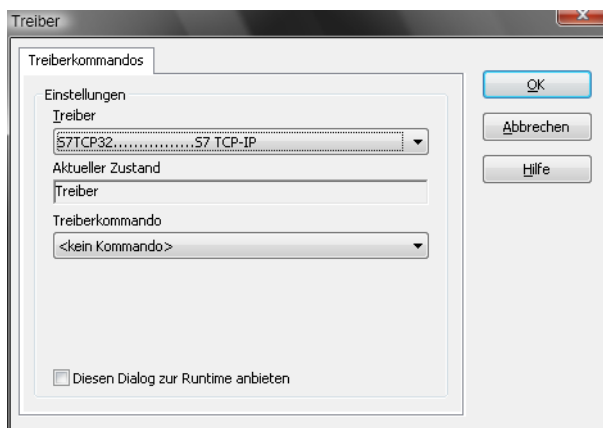
-

# 11. Treiberkommandos

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

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

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



Parameter	Beschreibung
Treiber	Dropdownliste mit allen im Projekt geladenen Treibern.
Aktueller Zustand	Fixer Eintrag, in aktuellen Versionen ohne Funktion.
<b>Treiberkommando</b>	Dropdownliste zur Auswahl des Kommandos.
▶ Treiber starten (Online-Modus)	Treiber wird neu initialisiert und gestartet.
▶ Treiber stoppen (Offline-Modus)	Treiber wird angehalten, es werden keine neuen Daten angenommen.  <b>Hinweis:</b> Ist der Treiber im Offline-Modus, erhalten alle Variablen, die für diesem Treiber angelegt wurden, den Status Abgeschaltet (OFF; Bit 20).
▶ Treiber in Simulationsmodus	Treiber wird in den Simulationsmodus gesetzt. Die Werte aller Variablen des Treibers werden vom Treiber simuliert. Es werden keine Werte von der angeschlossenen Hardware (z.B. SPS, Bussystem, ...) angezeigt.
▶ Treiber in Hardwaremodus	Treiber wird in den Hardwaremodus gesetzt. Für die Variablen des Treibers werden die Werte von der angeschlossenen Hardware (z.B. SPS, Bussystem, ...) angezeigt.
▶ Treiberspezifisches Kommando	Eingabe treiberspezifischer Kommandos. Öffnet Eingabefeld für die Eingabe eines Kommandos.
▶ Treiber Sollwertsetzen aktivieren	Sollwert setzen auf Treiber ist erlaubt.
▶ Treiber Sollwertsetzen deaktivieren	Sollwert setzen auf Treiber wird verhindert.
▶ Verbindung mit Modem aufbauen	Verbindung aufbauen (für Modem-Treiber). Öffnet Eingabefelder für Hardware-Adresse und Eingabe der zu wählenden Nummer.
▶ Verbindung mit Modem trennen	Verbindung beenden (für Modem-Treiber).
Diesen Dialog zur Runtime anbieten	Dialog wird zur Runtime für Änderungen angeboten.

## TREIBERKOMMANDOS IM NETZWERK

Wenn sich der Rechner, auf dem die Funktion **Treiberkommandos** ausgeführt wird, im zenon Netzwerk befindet, werden zusätzliche Aktionen ausgeführt. Ein spezielles Netzwerkkommando wird vom Rechner zum Server des Projekts gesendet, der dann die gewünschte Aktion auf seinem Treiber durchführt. Zusätzlich sendet der Server das gleiche Treiberkommando zum Standby des Projekts. Der Standby führt die Aktion auch auf seinem Treiber aus.

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

## 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 Hilfen, 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 Diagnose Viewer (main.chm::/12464.htm), das mit zenon mitinstalliert wird. Sie finden es unter *Start/Alle Programme/zenon/Tools 7.00 -> Diagviewer*.

zenon Treiber protokollieren alle Fehler in Log-Dateien. Der Standardordner für die Log-Dateien ist der Ordner `LOG` unterhalb des Ordners `ProgramData`, zum Beispiel: `C:\ProgramData\zenon\zenon700\LOG` für die zenon Version 7.00 SP0. Log-Dateien sind Textdateien mit einer speziellen Struktur.

**Achtung:** Mit den Standardeinstellungen zeichnet ein Treiber nur Fehlerinformationen auf. Mit dem `Diagnose Viewer` kann bei den meisten Treibern die Diagnose-Ebene auf „Debug“ und „Deep Debug“ erweitert werden. Damit protokolliert der Treiber auch alle anderen wesentlichen Aufgaben und Ereignisse.

Im Diagnose Viewer kann man auch:

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

#### Hinweise:

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

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



### Info

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

## 13.2 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 bitte der **FINS Dokumentation Section 5-1-3 „End Codes“**. Hier sind sämtliche Fehlernummern dokumentiert.

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