# Modbus RTU Serial-Treiber

© 2025 PTC Inc. Alle Rechte vorbehalten.

# Inhaltsverzeichnis

**Modbus RTU Serial-Treiber** 1
**Inhaltsverzeichnis** 2
Modbus RTU Serial-Treiber 5
Übersicht 5
**Setup** 5
Kanaleigenschaften – Allgemein 6
Tag-Zähler 7
Kanaleigenschaften – Kommunikationsserialisierung 7
Kanaleigenschaften – Schreiboptimierungen 9
Kanaleigenschaften – Serielle Kommunikation 10
Kanaleigenschaften – Erweitert 12
Geräteeigenschaften – Allgemein 13
Betriebsmodus 14
Tag-Zähler 15
Geräteeigenschaften – Scan-Modus 15
Geräteeigenschaften – Ethernet-Kapselung 16
Geräteeigenschaften – Zeitvorgabe 17
Geräteeigenschaften – Automatische Herabstufung 18
Geräteeigenschaften – Tag-Generierung 19
Geräteeigenschaften – Blockgrößen 22
Geräteeigenschaften – Variablenimporteinstellungen 24
Geräteeigenschaften – Framing 24
Geräteeigenschaften – Fehlerbehandlung 25
Geräteeigenschaften – Redundanz 26
**Automatische Tag-Datenbankgenerierung** 27
Statistikelemente 28
**Datentypbeschreibung** 31
**Adressbeschreibungen** 33
Modbus-Adressierung 33
Magnetek-GPD-515-Laufwerksadressierung 37
Elliott-Mengenumwerter-Adressierung 37
Daniels-S500-Mengenumwerter-Adressierung 38
Adressierung dynamischer Flüssigkeitsmessgeräte 39
Omni-Mengenumwerter-Adressierung 41
Benutzerdefinierte Omni-Datenpakete 45
Omni-Rohdatenarchiv 48

Omni-Textberichte 53
Omni-Textarchiv 56
Beschreibung von Funktionscodes 58
Konfigurations-API-Dienst – Kanaleigenschaften 59
Konfigurations-API-Dienst – Geräteeigenschaften 59
**Ereignisprotokollmeldungen 61**
Ungültige Adresse im Blockbereich. | Adressbereich = <Start> bis <Ende>. 61
Ungültiges Array. | Array-Bereich = <Start> bis <Ende>. 61
Blockanfrage mit Ausnahmecode beantwortet. | Adressenbereich = <Start> bis <Ende>, Ausnahmecode = <Code>. 62
In Adresse kann nicht geschrieben werden, Gerät hat mit Ausnahmecode geantwortet. | Adresse = '<Adresse>', Ausnahmecode = <Code>. 62
Von Adresse kann nicht gelesen werden, Gerät hat mit Ausnahmecode geantwortet. | Adresse = '<Adresse>', Ausnahmecode = <Code>. 62
Tag-Import wegen zu wenig Speicherressourcen fehlgeschlagen. 62
Beim Tag-Import ist eine Dateiausnahme aufgetreten. 63
Fehler beim Parsen von Datensatz in Importdatei. | Datensatznummer = <Nummer>, Feld = <Name>. 63
Beschreibung für Datensatz in Importdatei abgeschnitten. | Datensatznummer = <Nummer>. 63
Importierter Tag-Name ist ungültig und wurde geändert. | Tag-Name = '<Tag>', geänderter Tag-Name = '<Tag>'. 63
Tag konnte nicht importiert werden, da der Datentyp nicht unterstützt wird. | Tag-Name = '<Tag>', nicht unterstützter Datentyp = '<Typ>'. 64
Omni-Textpuffer konnte aufgrund eines Fehlers bei der Speicherzuordnung nicht gelesen werden. 64
Keine Omni-Textarchivdaten im angegebenen Datumsbereich verfügbar. 64
Texteingabe im Omni-Textbericht abgeschnitten. | Berichtsnummer = <Nummer>. 65
Omni-Textbericht konnte aufgrund von Grenzwert für Paketanzahl nicht gelesen werden. | Berichtsnummer = <Nummer>. 65
Schreibvorgang fehlgeschlagen. Maximale Pfadlänge überschritten. | Tag-Adresse = '<Adresse>', Maximale Länge = <Anzahl>. 65
Fehler beim Schreiben der Omni-Textdaten in Datei. | Tag-Adresse = '<Adresse>', Grund = '<Grund>'. 65
Omni-Textausgabedatei konnte nicht geöffnet werden. | Tag-Adresse = '<Adresse>', Grund = '<Grund>'. 66
In Adresse kann nicht geschrieben werden. Unerwartete Zeichen in Antwort. | Tag-Adresse = '<Adresse>'. 66
Von Adresse kann nicht gelesen werden. Unerwartete Zeichen in Antwort. | Tag-Adresse = '<Adresse>'. 66
Blockadresse konnte nicht gelesen werden. Unerwartete Zeichen in Antwort. | Adressbereich = <Start> bis <Ende>. 66
Die Omni-Textausgabedatei konnte nicht geändert werden. | Tag-Adresse = '<address>', Grund = 66

Der angegeben Pfad ist nicht zulässig. .....
Die Omni-Textausgabedatei konnte nicht geändert werden. | Tag-Adresse = '<address>', Grund = Die angegebene Dateierweiterung muss '.txt' oder '.log' sein. .....67
Tag-Datenbank wird von Datei importiert. | Dateiname = '<Name>'. ..... 67
Fehlermaskendefinitionen ..... 67
Modbus-Ausnahmecodes ..... 68
**Index** ..... **70**

# Modbus RTU Serial-Treiber

Hilfeversion 1.095

## INHALT

**Übersicht**
Was ist Modbus RTU Serial-Treiber?

**Setup**
Wie konfiguriere ich Kanäle und Geräte für die Verwendung mit diesem Treiber?

**Automatische Tag-Datenbankgenerierung**
Wie kann ich Tags für Modbus RTU Serial-Treiber konfigurieren?

**Datentypbeschreibung**
Welche Datentypen unterstützt dieser Treiber?

**Adressbeschreibungen**
Wie adressiere ich eine Datenposition auf einem Modbus-Gerät?

**Ereignisprotokollmeldungen**
Welche Meldungen werden vom Modbus RTU Serial-Treiber erstellt?

## Übersicht

Modbus RTU Serial-Treiber bietet eine zuverlässige Möglichkeit, serielle Modbus-Geräte mit OPC-Client-Anwendungen, u.a. HMI, SCADA, Historian, MES, ERP und zahlreichen benutzerdefinierten Anwendungen, zu verbinden. Es ist für die Verwendung mit seriellen Geräten vorgesehen, die das Modbus-RTU-Protokoll unterstützen. Modbus RTU Serial-Treiber wurde entwickelt, um eine breite Palette von Modbus-RTU-kompatiblen Geräte zu unterstützen.

## Setup

### Kommunikationsprotokoll

Modbus-RTU-Protokoll

### Unterstützte Kommunikationsparameter

Baudrate: 1200, 2400, 9600 und 19200
Parität: "Ungerade", "Gerade" und "Keine"
Daten-Bits: 8
Stopp-Bits: 1 und 2

### Unterstützte Geräte

- Modbus-kompatible Geräte
- Elliott-Mengenumwerter
- Magnetek-GPD-515-Laufwerk
- Omni-Mengenumwerter
- Daniel-S500-Mengenumwerter

- Dynamisches Flüssigkeitsmessgerät (DFM) SFC3
- TSXCUSBMBP-USB-Adapter

### Kanal- und Gerätegrenzwerte

- Die von diesem Treiber unterstützte maximale Anzahl von Kanälen liegt bei 2048.
- Bei Verwendung einer **seriellen Verbindung** beträgt die empfohlene maximale Anzahl von Geräten pro Kanal 32 (es sei denn, es werden Hardware-Signalverstärkungsgeräte verwendet).
- Bei Verwendung von **Ethernet-Kapselung** beträgt die maximale Anzahl von Geräten pro Kanal 255.

## Kanaleigenschaften – Allgemein

Dieser Server unterstützt die Verwendung von mehreren gleichzeitigen Kommunikationstreibern. Jedes Protokoll oder jeder Treiber, das/der in einem Serverprojekt verwendet wird, wird als Kanal bezeichnet. Ein Serverprojekt besteht unter Umständen aus vielen Kanälen mit demselben Kommunikationstreiber oder mit eindeutigen Kommunikationstreibern. Ein Kanal fungiert als grundlegender Baustein eines OPC-Links. Diese Gruppe wird verwendet, um allgemeine Kanaleigenschaften (wie z.B. die ID-Attribute und den Betriebsmodus) anzugeben.

### Identifikation

**Name**: Geben Sie die benutzerdefinierte ID dieses Kanals an. Bei jedem Serverprojekt muss jeder Kanalname eindeutig sein. Zwar können Namen bis zu 256 Zeichen lang sein, doch haben einige Client-Anwendungen beim Durchsuchen des Tag-Raums des OPC-Servers ein eingeschränktes Anzeigefenster. Der Kanalname ist ein Teil der OPC-Browserinformationen. Die Eigenschaft ist erforderlich, um einen Kanal zu erstellen.
*Informationen über reservierte Zeichen finden Sie in der Serverhilfe unter „So benennen Sie Kanäle, Geräte, Tags und Tag-Gruppen richtig“.*

**Beschreibung**: Geben Sie benutzerdefinierte Informationen über diesen Kanal an.
Viele dieser Eigenschaften, einschließlich der Beschreibung, verfügen über ein zugeordnetes System-Tag.

**Treiber**: Geben Sie das Protokoll/den Treiber für diesen Kanal an. Geben Sie den Gerätetreiber an, der während der Kanalerstellung ausgewählt wurde. Es ist eine deaktivierte Einstellung in den Kanaleigenschaften. Die Eigenschaft ist erforderlich, um einen Kanal zu erstellen.
**Hinweis**: Beim Online-Vollzeitbetrieb des Servers können diese Eigenschaften jederzeit geändert werden. Dies schließt das Ändern des Kanalnamens ein, um zu verhindern, dass Clients Daten am Server registrieren. Wenn ein Client bereits ein Element vom Server abgerufen hat, bevor der Kanalname geändert

wurde, sind die Elemente davon nicht beeinflusst. Wenn die Client-Anwendung das Element nach der Änderung des Kanalnamens freigibt und versucht, es mit dem alten Kanalnamen erneut abzurufen, wird das Element nicht akzeptiert. Es sollten keine Änderungen an den Eigenschaften erfolgen, sobald eine große Client-Anwendung entwickelt wurde. Verwenden Sie die richtige Benutzerrollen- und Berechtigungsverwaltung, um zu verhindern, dass Operatoren Eigenschaften ändern oder auf Serverfunktionen zugreifen.

### Diagnose

**Diagnoseerfassung**: Wenn diese Option aktiviert ist, stehen die Diagnoseinformationen des Kanals für OPC-Anwendungen zur Verfügung. Da für die Diagnosefunktionen des Servers eine minimale Mehraufwandsverarbeitung erforderlich ist, wird empfohlen, dass sie bei Bedarf verwendet werden und ansonsten deaktiviert sind. Die Standardeinstellung ist deaktiviert.

**Hinweise:** Diese Eigenschaft ist nicht verfügbar, wenn der Treiber Diagnosen nicht unterstützt.

*Weitere Informationen dazu finden Sie in der Serverhilfe unter "Kommunikationsdiagnosen".*

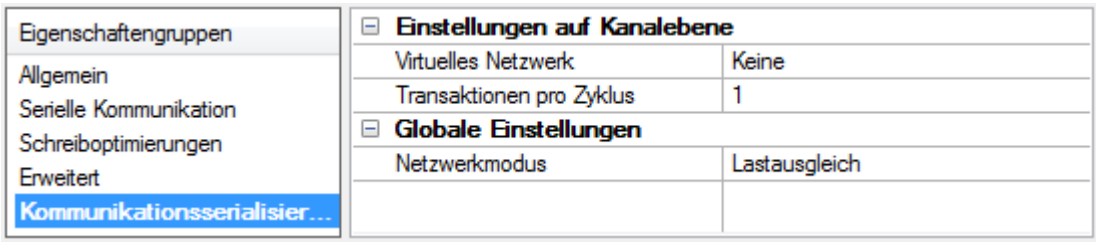
### Tag-Zähler

**Statische Tags**: Gibt die Gesamtanzahl der definierten statischen Tags auf dieser Ebene (Gerät oder Kanal) an. Diese Informationen können bei der Problembehandlung und beim Lastenausgleich hilfreich sein.

## Kanaleigenschaften – Kommunikationsserialisierung

Die Multithreading-Architektur des Servers ermöglicht Kanälen die parallele Kommunikation mit Geräten. Zwar ist das effizient, doch kann die Kommunikation in Fällen mit physischen Netzwerkeinschränkungen (wie Ethernet-Funksignale) serialisiert werden. Kommunikationsserialisierung schränkt die Kommunikation auf einen Kanal gleichzeitig innerhalb eines virtuellen Netzwerks ein.

Der Begriff "virtuelles Netzwerk" beschreibt eine Sammlung von Kanälen und zugeordneten Geräten, die dieselbe Pipeline für die Kommunikation verwenden. Beispielsweise ist die Pipeline eines Ethernet-Radios das Client-Radio. Alle Kanäle mit demselben Client-Radio werden demselben virtuellen Netzwerk zugeordnet. Kanäle dürfen jeweils nacheinander im Roundrobin-Verfahren kommunizieren. Standardmäßig kann ein Kanal eine Transaktion verarbeiten, bevor die Kommunikation an einen anderen Kanal übergeben wird. Eine Transaktion kann einen oder mehrere Tags einschließen Wenn der steuernde Kanal ein Gerät enthält, das nicht auf eine Anfrage antwortet, kann der Kanal die Steuerung erst bis zum Timeout der Transaktion freigeben. Dies hat Datenaktualisierungsverzögerungen für die anderen Kanäle im virtuellen Netzwerk zur Folge.




### Einstellungen auf Kanalebene

**Virtuelles Netzwerk**: Geben Sie den Kanalmodus der Kommunikationsserialisierung an. Zu den Optionen gehören "Keine" sowie "Netzwerk 1 – Netzwerk 500". Die Standardeinstellung ist "Keine". Es folgen Beschreibungen der Optionen:

- **Keine**: Mit dieser Option wird die Kommunikationsserialisierung für den Kanal deaktiviert.
- **Netzwerk 1 – Netzwerk 500**: Mit dieser Option wird das virtuelle Netzwerk angegeben, dem der Kanal zugewiesen wird.

**Transaktionen pro Zyklus**: Geben Sie die Anzahl einzelner blockierter/nicht blockierter Lese-/Schreibtransaktionen an, die auf dem Kanal vorkommen können. Wird einem Kanal die Möglichkeit zur Kommunikation gegeben, wird versucht, diese Anzahl von Transaktionen auszuführen. Der gültige Bereich liegt zwischen 1 und 99. Die Standardeinstellung ist 1.

## Globale Einstellungen

**Netzwerkmodus**: Mit dieser Eigenschaft wird gesteuert, wie die Kanalkommunikation delegiert wird. Im Modus **Lastausgleich** wird jedem Kanal die Möglichkeit gegeben, nacheinander zu kommunizieren. Im Modus **Priorität** wird Kanälen die Möglichkeit gegeben, nach den folgenden Regeln (von der höchsten zur niedrigsten Priorität) zu kommunizieren:

1. Kanäle mit ausstehenden Schreibvorgängen haben den höchsten Vorrang.
2. Kanäle mit ausstehenden expliziten Lesevorgängen (durch interne Plugins oder externe Client-Schnittstellen) werden je nach Priorität des Lesevorgangs priorisiert.
3. Gescannte Lesevorgänge und andere periodische Ereignisse (treiberspezifisch).

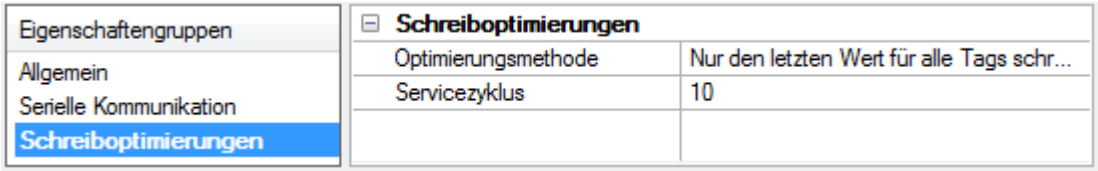
Die Standardeinstellung ist "Lastausgleich" und wirkt sich auf *alle* virtuellen Netzwerke und Kanäle aus.
Geräte, die sich auf unangeforderte Antworten verlassen, sollten nicht in ein virtuelles Netzwerk eingefügt werden. In Situationen, wo die Kommunikationen serialisiert werden muss, wird empfohlen, dass "Automatische Herabstufung" aktiviert wird.

Aufgrund von Unterschieden in der Art und Weise, wie Treiber Daten lesen und schreiben (wie z.B. einzelne blockierte oder nicht blockierte Transaktionen) muss die Eigenschaft "Transaktionen pro Zyklus" der Anwendung möglicherweise angepasst werden. Berücksichtigen Sie dabei die folgenden Faktoren:

- Wie viele Tags müssen von jedem Kanal gelesen werden?
- Wie oft werden Daten in jeden Kanal geschrieben?
- Verwendet der Kanal einen seriellen oder einen Ethernet-Treiber?
- Liest der Treiber Tags in separaten Anfragen, oder werden mehrere Tags in einem Block gelesen?
- Wurden die Zeitvorgabe-Eigenschaften des Geräts (wie z.B. Anforderungs-Timeout und Fehlgeschlagen nach *x* aufeinander folgenden Timeouts) für das Kommunikationsmedium des virtuellen Netzwerks optimiert?

## Kanaleigenschaften – Schreiboptimierungen

Der Server muss sicherstellen, dass die von der Client-Anwendung geschriebenen Daten rechtzeitig auf das Gerät gelangen. In Anbetracht dieses Ziels stellt der Server Optimierungseigenschaften bereit, um die jeweiligen Anforderungen zu erfüllen bzw. die Reaktionsfähigkeit der Anwendungen zu verbessern.




### Schreiboptimierungen

**Optimierungsmethode**: Mit dieser Option wird gesteuert, wie Schreibdaten an den zugrunde liegenden Kommunikationstreiber weitergeleitet werden. Die Optionen sind:

- **Alle Werte für alle Tags schreiben**: Mit dieser Option wird der Server gezwungen, für jeden Wert einen Schreibvorgang auf dem Controller zu versuchen. In diesem Modus sammelt der Server weiterhin Schreibanforderungen und fügt sie der internen Schreibwarteschlange des Servers hinzu. Der Server verarbeitet die Schreibwarteschlange und versucht, sie zu leeren, indem er so schnell wie möglich Daten auf das Gerät schreibt. In diesem Modus wird sichergestellt, dass alles, was von den Client-Anwendungen geschrieben wird, an das Zielgerät gesendet wird. Dieser Modus sollte ausgewählt werden, wenn die Reihenfolge des Schreibvorgangs oder der Inhalt des Schreibelements eindeutig auf dem Zielgerät zu finden sein muss.
- **Nur den letzten Wert für nicht boolesche Tags schreiben**: Viele aufeinander folgende Schreibvorgänge für denselben Wert können sich aufgrund der Zeit, die tatsächlich zum Senden der Daten auf das Gerät erforderlich ist, in der Schreibwarteschlange ansammeln. Wenn der Server einen Schreibwert aktualisiert, der bereits in die Schreibwarteschlange eingefügt wurde, sind weitaus weniger Schreibvorgänge erforderlich, um denselben Endausgabewert zu erhalten. Auf diese Weise sammeln sich keine zusätzlichen Schreibvorgänge in der Warteschlange des Servers an. Wenn der Benutzer den Schiebeschalter nicht mehr verschiebt, erreicht der Wert im Gerät praktisch in derselben Zeit den richtigen Wert. Dem Modus entsprechend wird jeder Wert, der kein boolescher Wert ist, in der internen Warteschlange des Servers aktualisiert und bei der nächstmöglichen Gelegenheit an das Gerät gesendet. Dies kann die Anwendungsleistung erheblich verbessern.
  **Hinweis**: Mit dieser Option wird nicht versucht, Schreibvorgänge in Boolesche Werte zu optimieren. Dadurch können Benutzer den HMI-Datenvorgang optimieren, ohne Probleme mit Booleschen Operationen (z.B. eine vorübergehende Schaltfläche) zu verursachen.
- **Nur den letzten Wert für alle Tags schreiben**: Mit dieser Option wird die hinter der zweiten Optimierungsmethode stehende Theorie auf alle Tags angewendet. Sie ist besonders nützlich, wenn die Anwendung nur den letzten Wert an das Gerät senden muss. In diesem Modus werden alle Schreibvorgänge optimiert, indem die derzeit in der Schreibwarteschlange befindlichen Tags vor dem Senden aktualisiert werden. Dies ist der Standardmodus.

**Servicezyklus**: Wird verwendet, um das Verhältnis von Schreib- und Lesevorgängen zu steuern. Das Verhältnis basiert immer auf einem Lesevorgang für jeden zehnten Schreibvorgang. Für den Servicezyklus wird standardmäßig 10 festgelegt. Dies bedeutet, dass 10 Schreibvorgänge für jeden Lesevorgang erfolgen. Zwar führt die Anwendung eine große Anzahl fortlaufender Schreibvorgänge durch, doch muss sichergestellt werden, dass es für Lesedaten weiterhin Verarbeitungszeit gibt. Die Einstellung 1 hat zur Folge, dass ein Lesevorgang für jeden Schreibvorgang erfolgt. Wenn es keine durchzuführenden Schreibvorgänge gibt, werden Lesevorgänge fortlaufend verarbeitet. Dies ermöglicht eine Optimierung für Anwendungen mit fortlaufenden Schreibvorgängen gegenüber einem ausbalancierteren Datenzufluss und -abfluss.

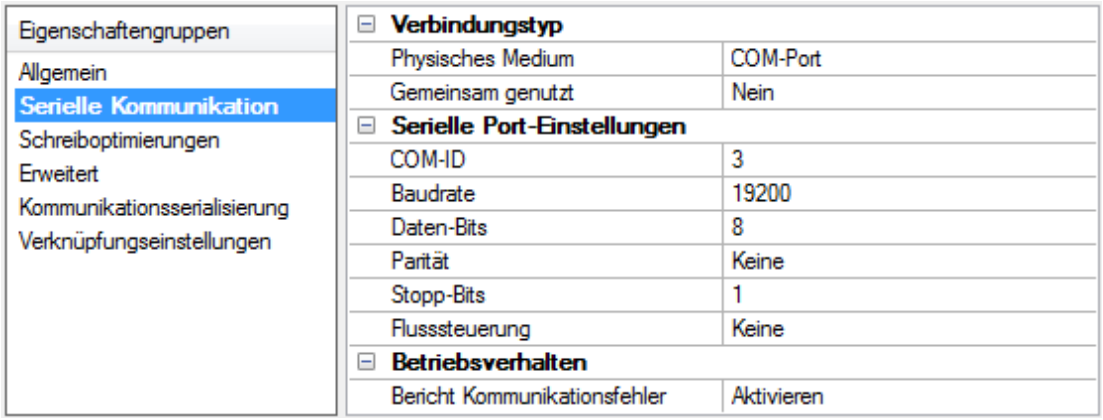
**Hinweis**: Es wird empfohlen, dass für die Anwendung die Kompatibilität mit den Verbesserungen zur Schreiboptimierung charakteristisch ist, bevor sie in einer Produktionsumgebung verwendet wird.

## Kanaleigenschaften – Serielle Kommunikation

Eigenschaften für serielle Kommunikation stehen seriellen Treibern zur Verfügung und sind je nach Treiber, Verbindungstyp und ausgewählten Optionen unterschiedlich. Unten finden Sie eine Übermenge der möglichen Eigenschaften.
Klicken Sie auf, um zu einem der Abschnitte zu springen: **Verbindungstyp**, **Serielle Port-Einstellungen**und **Betriebsverhalten**.

**Hinweise**:

- Während des Online-Vollzeitbetriebs des Servers können diese Eigenschaften jederzeit geändert werden. Verwenden Sie die richtige Benutzerrollen- und Berechtigungsverwaltung, um zu verhindern, dass Operatoren Eigenschaften ändern oder auf Serverfunktionen zugreifen.
- Benutzer müssen die spezifischen Parameter definieren, die verwendet werden sollen. Je nach Treiber können Kanäle identische Kommunikationsparameter gemeinsam nutzen. Es kann lediglich eine gemeinsam genutzte serielle Verbindung für ein virtuelles Netzwerk konfiguriert werden *(siehe **Kanaleigenschaften – Serielle Kommunikation**)*.




### Verbindungstyp

**Physisches Medium**: Wählen Sie den Hardware-Gerätetyp für Datenkommunikation. Zu den Optionen gehören Modem, COM-Port und Keine. Die Standardeinstellung ist COM-Port.

1. **Keine**: Wählen Sie "Keine" aus, um anzugeben, dass keine physische Verbindung vorhanden ist. Dadurch wird der Abschnitt **Operation ohne Kommunikation** angezeigt.
2. **COM-Port**: Wählen Sie "COM-Port" aus, um den Abschnitt **Serielle Port-Einstellungen** anzuzeigen und zu konfigurieren.
3. **Modem**: Wählen Sie "Modem" aus, wenn für die Kommunikation Telefonleitungen verwendet werden. Dies wird im Abschnitt **Modemeinstellungen** konfiguriert.
4. **Gemeinsam genutzt**: Überprüfen Sie, ob für die Verbindung korrekt angegeben ist, dass die aktuelle Konfiguration mit einem anderen Kanal gemeinsam genutzt wird. Dies ist eine schreibgeschützte Eigenschaft.

### Serielle Port-Einstellungen

**COM-ID**: Gibt die zu verwendende Kommunikations-ID bei der Kommunikation mit dem Kanal zugewiesenen Geräten an. Der gültige Bereich ist 1 bis 9991 bis 16. Die Standardeinstellung ist 1.

**Baudrate**: Geben Sie die Baudrate an, die zur Konfiguration des ausgewählten Kommunikationsports verwendet werden soll.

**Daten-Bits**: Geben Sie die Anzahl der Daten-Bits pro Datenwort an. Zu den Optionen gehören 5, 6, 7 oder 8.

**Parität**: Geben Sie den Paritätstyp für die Daten an. Zu den Optionen gehören "Ungerade", "Gerade" oder "Keine".

**Stopp-Bits**: Geben Sie die Anzahl der Stopp-Bits pro Datenwort an. Zu den Optionen gehören 1 oder 2.

**Flusssteuerung**: Wählen Sie aus, wie die RTS- und DTR-Steuerleitungen verwendet werden. Flusssteuerung ist für die Kommunikation mit einigen seriellen Geräten erforderlich. Es gibt folgende Optionen:

- **Keine**: Mit dieser Option werden keine Steuerleitungen umgeschaltet oder in den aktiven Zustand gebracht.
- **DTR**: Mit dieser Option wird die DTR-Leitung in den aktiven Zustand gebracht, wenn der Kommunikationsport geöffnet ist und es auch bleibt.
- **RTS**: Mit dieser Option wird angegeben, dass die RTS-Leitung hoch ist, wenn Byte für die Übertragung zur Verfügung stehen. Nachdem alle gepufferten Byte gesendet wurden, ist die RTS-Leitung niedrig. Dies wird normalerweise mit der RS232/RS485-Konverter-Hardware verwendet.
- **RTS, DTR**: Diese Option ist eine Kombination aus DTR und RTS.
- **RTS immer**: Mit dieser Option wird die RTS-Leitung in den aktiven Zustand gebracht, wenn der Kommunikationsport geöffnet ist und es auch bleibt.
- **RTS manuell**: Mit dieser Option wird die RTS-Leitung basierend auf den für RTS-Leitungssteuerung eingegebenen Zeitvorgaben-Eigenschaften in den aktiven Zustand gebracht. Sie steht nur zur Verfügung, wenn der Treiber manuelle RTS-Leitungssteuerung unterstützt (oder wenn die Eigenschaften gemeinsam benutzt werden und mindestens einer der Kanäle zu einem Treiber gehört, der diese Unterstützung bereitstellt). Durch "RTS manuell" wird die Eigenschaft **RTS-Leitungssteuerung** mit den folgenden Optionen hinzugefügt:
    - **Anstieg**: Geben Sie an, wie lang die RTS-Leitung vor der Datenübertragung ansteigt. Der gültige Bereich liegt zwischen 0 und 9999 Millisekunden. Die Standardeinstellung ist 10 Millisekunden.
    - **Abfall**: Geben Sie die Zeitdauer an, während der die RTS-Leitung nach der Datenübertragung hoch bleibt. Der gültige Bereich liegt zwischen 0 und 9999 Millisekunden. Die Standardeinstellung ist 10 Millisekunden.
    - **Abrufverzögerung**: Geben Sie die Zeit an, um die der Abruf für die Kommunikation verzögert ist. Der gültige Bereich liegt zwischen 0 und 9999. Die Standardeinstellung ist 10 Millisekunden.

**Tipp**: Bei Verwendung von doppeladrigen RS-485-Kabeln können "Echos" in den Kommunikationsleitungen auftreten. Da diese Kommunikation keine Echounterdrückung unterstützt, wird empfohlen, Echos zu deaktivieren oder einen RS-485-Konverter zu verwenden.

## Betriebsverhalten

- **Kommunikationsfehler melden**: Aktivieren oder deaktivieren Sie die Berichterstellung über geringfügige Kommunikationsfehler. Wenn diese Option aktiviert ist, werden geringfügige Fehler beim

Auftreten im Ereignisprotokoll angezeigt. Wenn diese Option deaktiviert ist, werden dieselben Fehler nicht angezeigt, selbst wenn es normale Anforderungsfehler sind. Die Standardeinstellung ist "Aktivieren".

- **Inaktive Verbindung schließen**: Wählen Sie diese Option, um die Verbindung zu schließen, wenn es keinerlei Tags mehr gibt, die von einem Client im Kanal referenziert werden. Die Standardeinstellung ist "Aktivieren".
- **Inaktivitätsdauer bis Schließen**: Geben Sie an, wie lang der Server warten soll, bis alle Tags vor dem Schließen des COM-Ports entfernt wurden. Der Standardwert ist 15 Sekunden.

### Modemeinstellungen

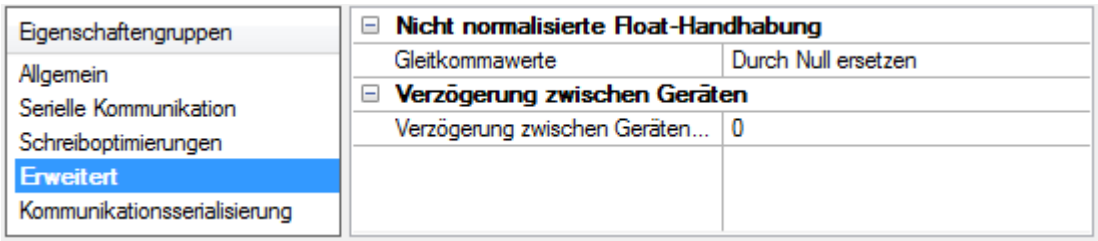
- **Modem**: Geben Sie das installierte Modem an, das für die Kommunikation verwendet werden soll.
- **Verbindungs-Timeout**: Diese Eigenschaft gibt an, wie lang auf das Herstellen von Verbindungen gewartet werden soll, bevor ein Lese- oder Schreibvorgang fehlschlägt. Der Standardwert ist 60 Sekunden.
- **Modemeigenschaften**: Konfigurieren Sie die Modem-Hardware. Durch Klicken auf diese Schaltfläche werden händlerspezifische Modemeigenschaften geöffnet.
- **Automatisches Wählen**: Ermöglicht das automatische Wählen von Einträgen im Telefonbuch. Die Standardeinstellung ist "Deaktivieren". *Weitere Informationen finden Sie unter "Modem Auto-Dial" in der Serverhilfe.*
- **Kommunikationsfehler melden**: Aktivieren oder deaktivieren Sie die Berichterstellung über geringfügige Kommunikationsfehler. Wenn diese Option aktiviert ist, werden geringfügige Fehler beim Auftreten im Ereignisprotokoll angezeigt. Wenn diese Option deaktiviert ist, werden dieselben Fehler nicht angezeigt, selbst wenn es normale Anforderungsfehler sind. Die Standardeinstellung ist "Aktivieren".
- **Inaktive Verbindung schließen**: Wählen Sie diese Option, um die Modemverbindung zu schließen, wenn es keinerlei Tags mehr gibt, die von einem Client im Kanal referenziert werden. Die Standardeinstellung ist "Aktivieren".
- **Inaktivitätsdauer bis Schließen**: Geben Sie an, wie lang der Server warten soll, bis alle Tags vor dem Schließen der Modemverbindung entfernt wurden. Der Standardwert ist 15 Sekunden.

### Operation ohne Kommunikation

- **Leseverarbeitung**: Wählen Sie aus, welche Maßnahmen ergriffen werden sollen, wenn ein expliziter Gerätelesevorgang angefordert wird. Zu den Optionen gehören Ignorieren und Fehlgeschlagen. Bei Ignorieren geschieht nichts, bei Fehlgeschlagen wird das Fehlschlagen dem Client durch eine Aktualisierung angezeigt. Die Standardeinstellung ist Ignorieren.

## Kanaleigenschaften – Erweitert

Diese Gruppe wird verwendet, um erweiterte Kanaleigenschaften anzugeben. Nicht alle Treiber unterstützen alle Eigenschaften; so wird die Gruppe "Erweitert" für jene Geräte nicht angezeigt.



**Nicht normalisierte Float-Handhabung**: Ein nicht normalisierter Wert wird als "Unendlich", "Nichtzahlenwert (NaN)" oder als "Denormalisierte Zahl" definiert. Die Standardeinstellung ist Durch Null ersetzen. Für Treiber, die eine native Float-Handhabung aufweisen, wird standardmäßig unter Umständen "Nicht geändert" verwendet. Durch Behandlung nicht normalisierter Gleitkommazahlen können Benutzer festlegen, wie ein Treiber mit nicht normalisierten IEEE-754-Gleitkommadaten umgeht. Es folgen Beschreibungen der Optionen:

- **Durch Null ersetzen**: Diese Option ermöglicht es einem Treiber, nicht normalisierte IEEE-754-Gleitkommawerte durch Null zu ersetzen, bevor sie an Clients übertragen werden.
- **Nicht geändert**: Diese Option ermöglicht es einem Treiber, denormalisierte, normalisierte IEEE-754-Nichtzahlenwerte und unendliche IEEE-754-Werte ohne jegliche Konvertierung oder Änderungen an Clients zu übertragen.

**Hinweis:** Diese Eigenschaft ist deaktiviert, wenn der Treiber keine Gleitkommawerte unterstützt, oder wenn er nur die angezeigte Option unterstützt. Gemäß der Float-Normalisierungseinstellung des Kanals unterliegen nur Echtzeit-Treiber-Tags (wie z.B. Werte und Arrays) der Float-Normalisierung. Beispielsweise werden EFM-Daten nicht durch diese Einstellung beeinflusst.

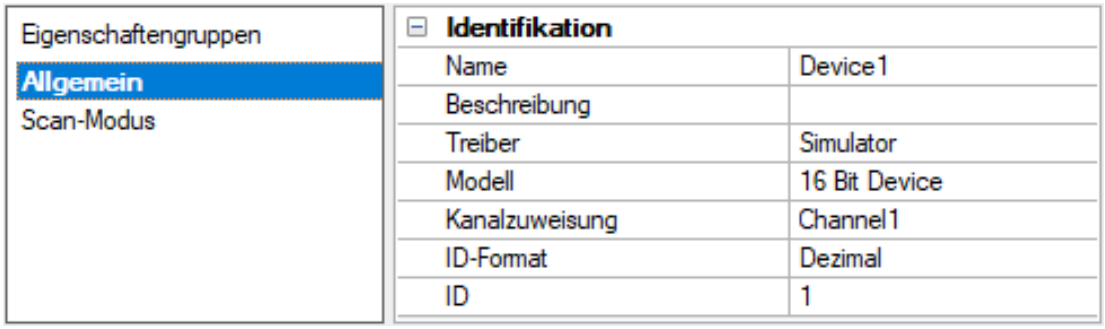
*Weitere Informationen über die Gleitkommawerte finden Sie unter "Mit nicht normalisierten Gleitkommawerten arbeiten" in der Serverhilfe.*

**Verzögerung zwischen Geräten**: Geben Sie die Zeitdauer an, in der der Kommunikationskanal das Senden einer Anforderung an das nächste Gerät verzögert, nachdem Daten vom aktuellen Gerät in demselben Kanal empfangen wurden. Null (0) deaktiviert die Verzögerung.

**Hinweis:** Diese Eigenschaft ist nicht für alle Treiber, Modelle und abhängige Einstellungen verfügbar.

## Geräteeigenschaften – Allgemein

Ein Gerät stellt ein einzelnes Ziel in einem Kommunikationskanal dar. Wenn der Treiber mehrere Controller unterstützt, müssen Benutzer eine Geräte-ID für jeden Controller eingeben.




### Identifikation

**Name**: Geben Sie den Namen des Geräts an. Es ist ein logischer, benutzerdefinierter Name, der bis zu 256 Zeichen lang sein und auf mehreren Kanälen verwendet werden kann.

**Hinweis**: Zwar sind beschreibende Namen allgemein eine gute Idee, doch haben einige OPC-Client-Anwendungen beim Durchsuchen des Tag-Raums des OPC-Servers möglicherweise ein eingeschränktes Anzeigefenster. Der Geräte- und Kanalname werden ebenfalls Teil der Informationen zum Durchsuchen der Hierarchiebaumstruktur. Innerhalb eines OPC-Clients würde die Kombination aus Kanalname und Gerätename als "ChannelName.DeviceName" angezeigt werden.

*Weitere Informationen dazu finden Sie in der Serverhilfe unter "So benennen Sie Kanäle, Geräte, Tags und Tag-Gruppen richtig".*

**Beschreibung**: Geben Sie die benutzerdefinierten Informationen über dieses Gerät an.
Viele dieser Eigenschaften, einschließlich der Beschreibung, verfügen über ein zugeordnetes System-Tag.

**Kanalzuweisung**: Geben Sie den benutzerdefinierten Namen des Kanals an, zu dem dieses Gerät derzeit gehört.

**Treiber**: Ausgewählter Protokolltreiber für dieses Gerät.

**Modell**: Geben Sie den Gerätetyp an, der dieser ID zugeordnet ist. Der Inhalt des Dropdown-Menüs hängt vom Typ des verwendeten Kommunikationstreibers ab. Modelle, die von einem Treiber nicht unterstützt werden, sind deaktiviert. Wenn der Kommunikationstreiber mehrere Gerätemodelle unterstützt, kann die Modellauswahl nur geändert werden, wenn keine Client-Anwendungen mit dem Gerät verbunden sind.
**Hinweis:** Wenn der Kommunikationstreiber mehrere Modelle unterstützt, sollten Benutzer versuchen, die Modellauswahl mit dem physischen Gerät abzugleichen. Wenn das Gerät im Dropdown-Menü nicht dargestellt wird, wählen Sie ein Modell aus, das dem Zielgerät am ehesten entspricht. Einige Treiber unterstützen die Modellauswahl "Offen", wodurch Benutzer kommunizieren können, ohne bestimmte Details des Zielgeräts zu kennen. *Weitere Informationen dazu finden Sie in der Dokumentation für den Treiber.*

**ID**: Geben Sie die treiberspezifische Station oder den treiberspezifischen Knoten für das Gerät an. Der Typ der eingegebenen ID hängt vom verwendeten Kommunikationstreiber ab. Für viele Kommunikationstreiber ist die ID ein numerischer Wert. Treiber, die eine numerische ID unterstützen, stellen Benutzern die Option zum Eingeben eines numerischen Werts bereit, dessen Format den Anforderungen der Anwendung oder der Charakteristik des ausgewählten Kommunikationstreibers entsprechend angepasst werden kann. Das Format wird standardmäßig durch den Treiber festgelegt. Zu den Optionen gehören "Dezimal", "Oktal" und "Hexadezimal".

## Betriebsmodus

**Datensammlung**: Diese Eigenschaft steuert den aktiven Status des Geräts. Zwar sind Gerätekommunikationen standardmäßig aktiviert, doch kann diese Eigenschaft verwendet werden, um ein physisches Gerät zu deaktivieren. Kommunikationen werden nicht versucht, wenn ein Gerät deaktiviert ist. Vom Standpunkt eines Clients werden die Daten als ungültig markiert und Schreibvorgänge werden nicht akzeptiert. Diese Eigenschaft kann jederzeit durch diese Eigenschaft oder die System-Tags des Geräts geändert werden.

**Simuliert**: Versetzen Sie das Gerät in den Simulationsmodus, oder beenden Sie den Modus. In diesem Modus versucht der Treiber nicht, mit dem physischen Gerät zu kommunizieren, aber der Server gibt weiterhin gültige OPC-Daten zurück. Durch Auswählen von "Simuliert" wird die physische Kommunikation mit dem Gerät angehalten, OPC-Daten können jedoch als gültige Daten dem OPC-Client zurückgegeben werden. Im Simulationsmodus behandelt der Server alle Gerätedaten als reflektierend: was auch immer in das simulierte Gerät geschrieben wird, wird zurückgelesen, und jedes OPC-Element wird einzeln behandelt. Die

Daten werden nicht gespeichert, wenn der Server das Element entfernt (z.B., wenn der Server neu initialisiert wird). Die Standardeinstellung ist "Nein".

**Hinweise:**

1. Aktualisierungen werden erst nach dem Trennen von Clients und nach dem Wiederherstellen von deren Verbindung angewendet.
2. Das System-Tag (_Simulated) ist schreibgeschützt und kann für den Laufzeitschutz nicht geschrieben werden. Das System-Tag ermöglicht es, dass diese Eigenschaft vom Client überwacht wird.
3. Im Simulationsmodus basiert die Speicherzuordnung des Elements auf Client-Aktualisierungsraten (Gruppenaktualisierungsrate für OPC-Clients oder Scan-Intervall für native und DDE-Schnittstellen). Das bedeutet, dass zwei Clients, die dasselbe Element mit unterschiedlichen Aktualisierungsraten referenzieren, verschiedene Daten zurückgeben.
4. Wird ein Gerät simuliert, so werden Aktualisierungen möglicherweise nicht schneller als innerhalb einer Sekunde auf dem Client angezeigt.

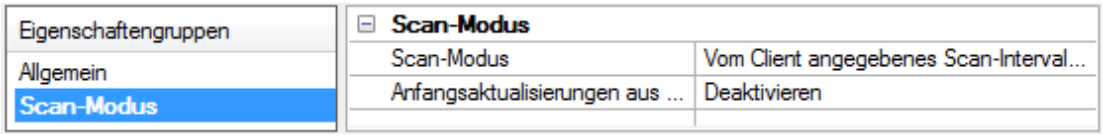
Der Simulationsmodus ist nur für Test- und Simulationszwecke. Es sollte niemals in einer Produktionsumgebung nie verwendet werden.

### Tag-Zähler

**Statische Tags**: Gibt die Gesamtanzahl der definierten statischen Tags auf dieser Ebene (Gerät oder Kanal) an. Diese Informationen können bei der Problembehandlung und beim Lastenausgleich hilfreich sein.

## Geräteeigenschaften – Scan-Modus

Der Scan-Modus gibt das vom abonnierten Client angeforderte Scan-Intervall für Tags an, die Gerätekommunikation erfordern. Synchrone und asynchrone Lese- und Schreibvorgänge des Geräts werden so bald wie möglich verarbeitet; unbeeinflusst von den Eigenschaften für den Scan-Modus.




**Scan-Modus**: Geben Sie an, wie Tags im Gerät für an abonnierende Clients gesendete Aktualisierungen gescannt werden. Es folgen Beschreibungen der Optionen:

- **Vom Client angegebenes Scan-Intervall berücksichtigen**: Dieser Modus verwendet das vom Client angeforderte Scan-Intervall.
- **Datenanfrage nicht schneller als Scan-Intervall**: Dieser Modus gibt den Wert an, der als maximales Scan-Intervall festgelegt wurde. Der gültige Bereich liegt zwischen 10 und 99999990 Millisekunden. Die Standardeinstellung ist 1000 Millisekunden.
  **Hinweis**: Wenn der Server über einen aktiven Client und Elemente für das Gerät verfügt und der Wert für das Scan-Intervall erhöht wird, werden die Änderungen sofort wirksam. Wenn der Wert für

das Scan-Intervall verringert wird, werden die Änderungen erst wirksam, wenn alle Client-Anwendungen getrennt wurden.

- **Alle Datenanfragen im Scan-Intervall**: Dieser Modus erzwingt, dass Tags im angegebenen Intervall nach abonnierten Clients gescannt werden. Der gültige Bereich liegt zwischen 10 und 99999990 Millisekunden. Die Standardeinstellung ist 1000 Millisekunden.
- **Nicht scannen, nur Abruf anfordern**: In diesem Modus werden Tags, die zum Gerät gehören, nicht periodisch abgerufen, und es wird auch kein Lesevorgang durchgeführt, um den Anfangswert eines Elements abzurufen, sobald es aktiv wird. Es liegt in der Verantwortung des OPC-Clients, Aktualisierungen abzurufen, entweder durch Schreiben in das _DemandPoll-Tag oder durch Ausgeben expliziter Lesevorgänge des Geräts für einzelne Elemente. *Weitere Informationen finden Sie unter "Geräte-Bedarfsabruf" in der Serverhilfe*.
- **Durch Tag angegebenes Scan-Intervall berücksichtigen**: Dieser Modus erzwingt das Scannen statischer Tags im Intervall, das in ihrer statischen Konfiguration Tag-Eigenschaften angegeben wurde. Dynamische Tags werden in dem vom Client angegebenen Scan-Intervall gescannt.

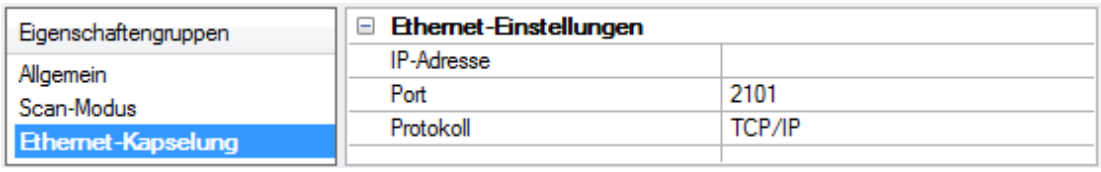
**Anfangsaktualisierungen aus Cache**: Wenn diese Option aktiviert ist, kann der Server die ersten Aktualisierungen für neu aktivierte Tag-Referenzen aus gespeicherten (Cache-)Daten zur Verfügung stellen. Cache-Aktualisierungen können nur bereitgestellt werden, wenn die neue Elementreferenz dieselben Eigenschaften für Adresse, Scan-Intervall, Datentyp, Client-Zugriff und Skalierung gemeinsam nutzt. Ein Lesevorgang des Geräts wird nur für die Anfangsaktualisierung für die erste Client-Referenz verwendet. Der Standardeinstellung ist "Deaktiviert"; immer wenn ein Client eine Tag-Referenz aktiviert, versucht der Server, den Anfangswert vom Gerät zu lesen.

## Geräteeigenschaften – Ethernet-Kapselung

Ethernet-Kapselung ist für die Kommunikation mit seriellen Geräten vorgesehen, die im Ethernet-Netzwerk mit Terminalservern verbunden sind. Ein Terminalserver ist im Wesentlichen ein virtueller serieller Port. Er wandelt TCP/IP-Meldungen im Ethernet-Netzwerk in serielle Daten um. Sobald die Meldung in ein serielles Format umgewandelt wurde, können Benutzer Standardgeräte verbinden, die eine serielle Kommunikation mit dem Terminalserver unterstützen.

*Weitere Informationen dazu finden Sie in der Serverhilfe unter "So verwenden Sie Ethernet-Kapselung".*

Ethernet-Kapselung ist für den Treiber transparent; konfigurieren Sie die übrigen Eigenschaften, als ob direkt über einen lokalen seriellen Port eine Verbindung mit dem Gerät hergestellt wird.




**IP-Adresse**: Geben Sie die Vier-Feld-IP-Adresse des Terminalservers ein, mit dem das Gerät verbunden ist. IP-Adressen werden als YYY.YYY.YYY.YYY angegeben. YYY bestimmt die IP-Adresse: Jedes YYY-Byte sollte im Bereich von 0 bis 255 liegen. Jedes serielle Gerät kann seine eigene IP-Adresse haben; jedoch können Geräte dieselbe IP-Adresse haben, wenn es mehrere Geräte gibt, die von einem einzelnen Terminalserver mehrfach abgelegt wurden.

**Port**: Konfigurieren Sie den Ethernet-Port, der beim Herstellen der Verbindung mit einem Remote-Terminalserver verwendet werden soll.

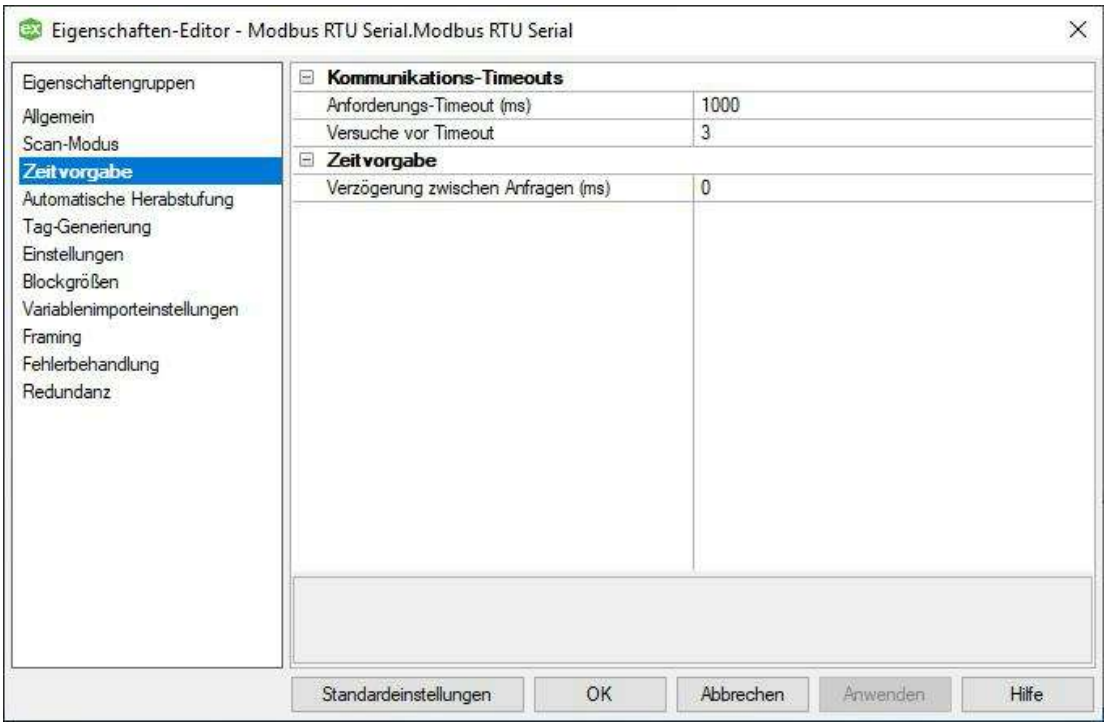
**Protokoll**: Legen Sie die TCP/IP- oder UDP-Kommunikation fest. Die Auswahl hängt von der Art des verwendeten Terminalservers ab. Die Standardprotokollauswahl ist TCP/IP. Weitere Informationen über verfügbare Protokolle finden Sie in der Hilfedokumentation des Terminalservers.

**Hinweise**

1. Während des Online-Vollzeitbetriebs des Servers können diese Eigenschaften jederzeit geändert werden. Verwenden Sie die richtige Benutzerrollen- und Berechtigungsverwaltung, um zu verhindern, dass Operatoren Eigenschaften ändern oder auf Serverfunktionen zugreifen.
2. Der gültige IP-Adressbereich liegt zwischen größer als (>) 0.0.0.0 und kleiner als (<) 255.255.255.255.

## Geräteeigenschaften – Zeitvorgabe

Mithilfe der Zeitvorgabe-Eigenschaften des Geräts kann die Antwort des Treibers auf Fehlerbedingungen so angepasst werden, dass sie den Anforderungen der Anwendung entspricht. In vielen Fällen erfordert die Umgebung für eine optimale Leistung Änderungen an diesen Eigenschaften. Faktoren wie elektrisch generiertes Rauschen, Modemverzögerungen und fehlerhafte physische Verbindungen können beeinflussen, wie viele Fehler oder Timeouts ein Kommunikationstreiber feststellt. Zeitvorgabe-Eigenschaften sind für jedes konfigurierte Gerät spezifisch.




### Kommunikations-Timeouts

**Anforderungs-Timeout**: Mit dieser Eigenschaft wird ein von allen Treibern verwendetes Intervall festgelegt, um zu bestimmen, wie lange der Treiber abschließend auf eine Antwort vom Zielgerät wartet. Der gültige Bereich liegt zwischen 50 und 9.999.999 Millisekunden (167,6667 Minuten). Die Standardeinstellung ist im Allgemeinen 1000 Millisekunden, kann jedoch abhängig vom Treiber unterschiedlich sein. Das Standard-Timeout für die meisten seriellen Treiber basiert auf einer Baudrate von 9600 Baud oder besser. Wenn ein Treiber bei niedrigeren Baudraten verwendet wird, erhöhen Sie das Timeout, um die erhöhte Zeit auszugleichen, die zum Abrufen von Daten erforderlich ist.

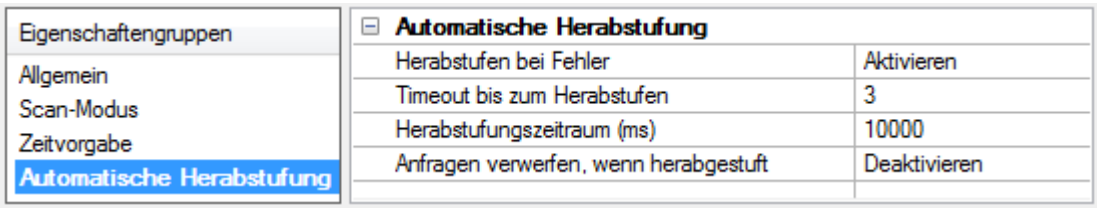
**Versuche vor Timeout**: Mit dieser Eigenschaft wird festgelegt, wie oft der Treiber eine Kommunikationsanforderung wiederholt, bevor er die Anforderung als fehlgeschlagen und das Gerät als fehlerhaft erachtet. Der gültige Bereich liegt zwischen 1 und 10. Die Standardeinstellung ist normalerweise 3, kann sich jedoch abhängig vom jeweiligen Treiber ändern. Die Anzahl der für eine Anwendung konfigurierten Wiederholungen hängt größtenteils von der Kommunikationsumgebung ab. Diese Eigenschaft trifft sowohl auf Verbindungsversuche als auch auf Anforderungsversuche zu.

### Zeitvorgabe

**Verzögerung zwischen Anfragen**: Mit dieser Eigenschaft wird festgelegt, wie lange der Treiber wartet, bevor er die nächste Anforderung an das Zielgerät sendet. Sie setzt das dem Gerät zugewiesene normale Tag-Abfrageintervall sowie einmalige Lese- und Schreibvorgänge außer Kraft. Diese Verzögerung kann bei Geräten mit langsamen Durchlaufzeiten und in Situationen nützlich sein, in denen die Netzwerklast problematisch ist. Das Konfigurieren einer Verzögerung für ein Gerät wirkt sich auf die Kommunikation mit allen anderen Geräten im Kanal aus. Es wird empfohlen, dass Benutzer jedes Gerät trennen, das eine Verzögerung zwischen Anfragen für einen separaten Kanal erfordert (sofern möglich). Andere Kommunikationseigenschaften (wie z.B. Kommunikationsserialisierung) können diese Verzögerung verlängern. Der gültige Bereich liegt zwischen 0 und 300000 Millisekunden; jedoch können einige Treiber ggf. den maximalen Wert wegen einer Funktion ihrer spezifischen Konstruktion beschränken. Die Standardeinstellung ist 0. Dies weist darauf hin, dass es keine Verzögerung zwischen Anfragen mit dem Zielgerät gibt.
**Hinweis**: Nicht alle Treiber unterstützen Verzögerung zwischen Anfragen. Diese Einstellung wird nicht angezeigt, wenn sie nicht zur Verfügung steht.

## Geräteeigenschaften – Automatische Herabstufung

Die Eigenschaften für automatische Herabstufung können ein Gerät vorübergehend in den Nicht-Scan-Modus versetzen, falls das Gerät nicht antwortet. Dadurch, dass ein nicht reagierendes Gerät für einen bestimmten Zeitraum offline gestellt wird, kann der Treiber weiterhin seine Kommunikation mit anderen Geräten in demselben Kanal optimieren. Nach Ablauf dieses Zeitraums versucht der Treiber die Kommunikation mit dem nicht reagierenden Gerät erneut. Wenn das Gerät reagiert, wird es wieder zum Scannen freigegeben. Andernfalls wird sein Nicht-Scan-Zeitraum erneut gestartet.




**Herabstufen bei Fehler**: Wird diese Option aktiviert, wird das Gerät automatisch in den Nicht-Scan-Modus versetzt, bis es wieder antwortet.
**Tipp**: Ermitteln Sie, wenn sich ein Gerät im Nicht-Scan-Modus befindet, indem Sie seinen herabgestuften Status mit dem _AutoDemoted-System-Tag überwachen.

**Timeout bis zum Herabstufen**: Legen Sie fest, wie viele aufeinander folgende Zyklen von Anforderungs-Timeouts und Wiederholungen vorkommen, bevor das Gerät in den Nicht-Scan-Modus versetzt wird. Der gültige Bereich ist 1 bis 30 aufeinander folgende Fehlschläge. Die Standardeinstellung ist 3.

**Herabstufungszeitraum**: Gibt an, wie lange das Gerät im Nicht-Scan-Modus sein sollte, wenn der Timeout-Wert erreicht wird. Während dieses Zeitraums werden keine Leseanforderungen an das Gerät gesendet, und für alle den Leseanforderungen zugeordneten Daten wird schlechte Qualität festgelegt. Wenn dieser Zeitraum abgelaufen ist, versetzt der Treiber das Gerät in den Scan-Modus und ermöglicht einen weiteren

Kommunikationsversuch. Der gültige Bereich liegt zwischen 100 und 3600000 Millisekunden. Die Standardeinstellung ist 10000 Millisekunden.

**Anfragen verwerfen, wenn herabgestuft**: Durch Aktivieren dieser Option wird ausgewählt, ob Schreibanforderungen während des Nicht-Scan-Zeitraums versucht werden sollten. Deaktivieren Sie diese Option, damit Schreibanforderungen unabhängig vom Herabstufungszeitraum immer gesendet werden. Aktivieren Sie diese Option, um Schreibvorgänge zu verwerfen; auf dem Server schlägt jede von einem Client empfangene Schreibanforderung automatisch fehl, und es wird keine Meldung im Ereignisprotokoll angezeigt.

## Geräteeigenschaften – Tag-Generierung

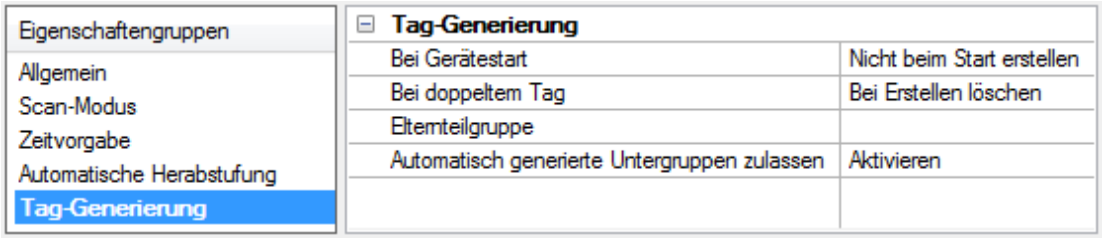
Mithilfe der Funktionen zur automatischen Tag-Datenbankgenerierung wird die Einrichtung einer Anwendung zu einem Plug-and-Play-Vorgang. Ausgewählte Kommunikationstreiber können so konfiguriert werden, dass automatisch eine Liste von Tags erstellt wird, die gerätespezifischen Daten entsprechen. Diese automatisch generierten Tags (die von der Art des unterstützenden Treibers abhängen) können von den Clients durchsucht werden.

*Nicht alle Geräte und Treiber unterstützen die automatische Tag-Datenbankgenerierung und nicht alle unterstützen die gleichen Datentypen. Lesen Sie für weitere Informationen die Beschreibungen der Datentypen oder die Liste der unterstützten Datentypen für jeden Treiber.*

Wenn das Zielgerät seine eigene lokale Tag-Datenbank unterstützt, liest der Treiber die Tag-Informationen des Geräts und verwendet die Daten zum Generieren von Tags innerhalb des Servers. Wenn das Gerät benannte Tags nicht nativ unterstützt, erstellt der Treiber eine Liste von auf treiberspezifischen Informationen basierenden Tags. Ein Beispiel dieser beiden Bedingungen sieht wie folgt aus:

1. Wenn ein Datenerfassungssystem seine eigene lokale Tag-Datenbank unterstützt, verwendet der Kommunikationstreiber die im Gerät gefundenen Tag-Namen, um die Tags des Servers zu erstellen.
2. Wenn ein Ethernet-E/A-System die Erkennung seiner eigenen verfügbaren E/A-Modultypen unterstützt, generiert der Kommunikationstreiber automatisch Tags auf dem Server, die auf den E/A-Modultypen im Ethernet-E/A-Rack basieren.

**Hinweis**: Der Betriebsmodus zur automatischen Tag-Datenbankgenerierung ist komplett konfigurierbar. *Weitere Informationen finden Sie in den nachfolgenden Eigenschaftsbeschreibungen.*




**Bei Eigenschaftsänderung**: Unterstützt das Gerät automatische Tag-Generierung, wenn bestimmte Eigenschaften geändert werden, so wird die Option **Bei Eigenschaftsänderung** angezeigt. Diese Option ist standardmäßig auf **Ja** festgelegt, kann jedoch auf **Nein** gesetzt werden, um zu steuern, wann eine Tag-Generierung stattfindet. In diesem Fall muss die Aktion **Tags erstellen** manuell aufgerufen werden, damit eine Tag-Generierung stattfindet.

**Bei Gerätestart**: Geben Sie an, wann OPC-Tags automatisch generiert werden. Es folgen Beschreibungen der Optionen:

- **Nicht beim Start erstellen**: Mit dieser Option wird verhindert, dass der Treiber irgendwelche OPC-Tags dem Tag-Raum des Servers hinzufügt. Dies ist die Standardeinstellung.
- **Immer beim Start erstellen**: Das Auswählen dieser Option hat zur Folge, dass der Treiber das Gerät für Tag-Informationen bewertet. Es werden auch jedes Mal, wenn der Server gestartet wird, Tags dem Tag-Raum des Servers hinzugefügt.
- **Beim ersten Start erstellen**: Das Auswählen dieser Option hat zur Folge, dass der Treiber das Zielgerät für Tag-Informationen bewertet, wenn das Projekt zum ersten Mal ausgeführt wird. Es werden bei Bedarf auch sämtliche OPC-Tags dem Tag-Raum des Servers hinzugefügt.

**Hinweis**: Wenn die Option zum automatischen Generieren von OPC-Tags ausgewählt wird, müssen sämtliche Tags, die dem Tag-Raum des Servers hinzugefügt werden, mit dem Projekt gespeichert werden. Benutzer können das Projekt konfigurieren, um automatisch über das Menü **Tools** | **Optionen** zu speichern.

**Bei doppeltem Tag**: Wenn die automatische Tag-Datenbankgenerierung aktiviert wird, muss der Server wissen, wie mit Tags, die er möglicherweise zuvor hinzugefügt hat, oder mit Tags, die nach dem Kommunikationstreiber seit ihrer ursprünglichen Erstellung hinzugefügt oder geändert wurden, zu verfahren ist. Mit dieser Einstellung wird gesteuert, wie der Server OPC-Tags behandelt, die automatisch generiert wurden und derzeit im Projekt vorhanden sind. Es wird auch verhindert, dass sich automatisch generierte Tags auf dem Server ansammeln.

Beispiel: Wenn ein Benutzer die E/A-Module im Rack mit dem für **Immer beim Start erstellen** konfigurierten Server ändert, würden neue Tags jedes Mal dem Server hinzugefügt werden, wenn der Kommunikationstreiber ein neues E/A-Modul erkannt hat. Wenn die alten Tags nicht entfernt wurden, könnten sich viele unbenutzte Tags im Tag-Raum des Servers ansammeln. Die Optionen sind:

- **Bei Erstellen löschen**: Mit dieser Option werden sämtliche Tags gelöscht, die zuvor dem Tag-Raum hinzugefügt wurden, bevor sämtliche neuen Tags hinzugefügt werden. Dies ist die Standardeinstellung.
- **Nach Bedarf überschreiben**: Mit dieser Option wird der Server angewiesen, nur die Tags zu entfernen, die der Kommunikationstreiber durch neue Tags ersetzt. Sämtliche Tags, die nicht überschrieben werden, bleiben im Tag-Raum des Servers.
- **Nicht überschreiben**: Mit dieser Option wird verhindert, dass der Server sämtliche Tags entfernt, die zuvor generiert wurden oder bereits auf dem Server vorhanden waren. Der Kommunikationstreiber kann nur Tags hinzufügen, die völlig neu sind.
- **Nicht überschreiben, Fehler protokollieren**: Diese Option hat denselben Effekt wie die vorherige Option und sendet auch eine Fehlermeldung an das Ereignisprotokoll des Servers, wenn eine Tag-Überschreibung stattgefunden hätte.

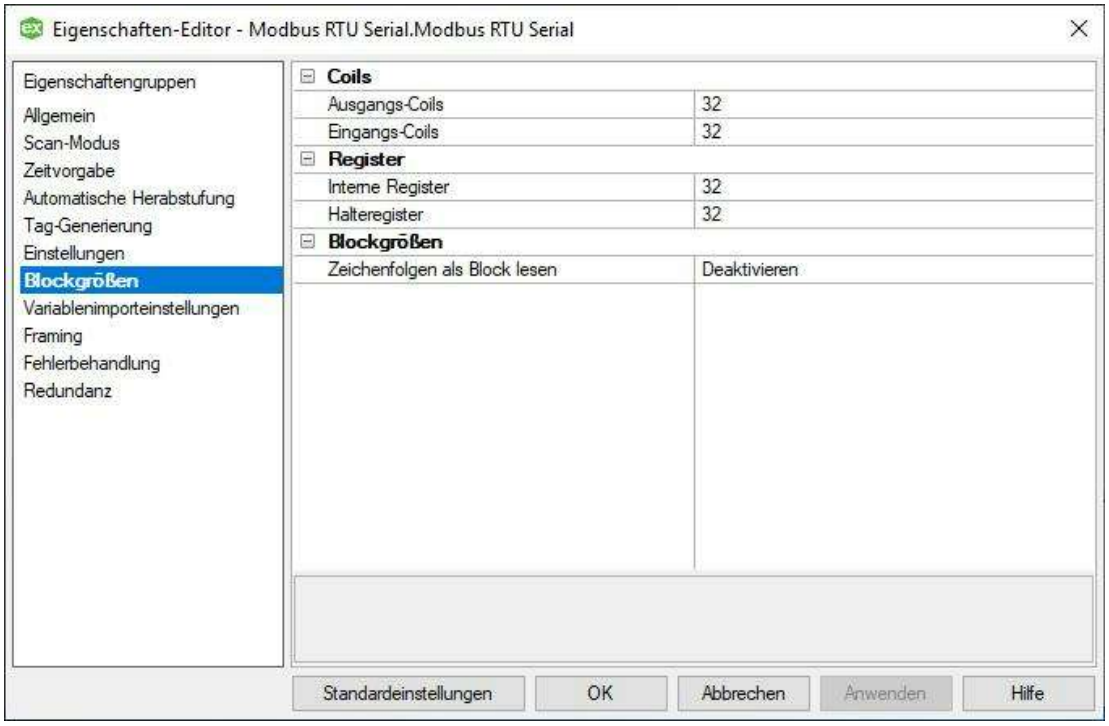
**Hinweis:** Das Entfernen von OPC-Tags wirkt sich auf Tags, die automatisch vom Kommunikationstreiber generiert wurden, sowie auf sämtliche Tags aus, die unter Verwendung von Namen, die generierten Tags entsprechen, hinzugefügt wurden. Benutzer sollten es vermeiden, Tags dem Server unter Verwendung von Namen hinzuzufügen, die möglicherweise den Tags entsprechen, die automatisch vom Treiber generiert werden.

**Elternteilgruppe**: Mit dieser Eigenschaft wird verhindert, dass sich automatisch generierte Tags mit Tags vermischen, die manuell eingegeben wurden, indem eine Gruppe festgelegt wurde, die für automatisch generierte Tags verwendet werden soll. Der Name der Gruppe kann bis zu 256 Zeichen lang sein. Diese Elternteilgruppe stellt einen Stammzweig bereit, dem alle automatisch generierten Tags hinzugefügt werden.

**Automatisch generierte Untergruppen zulassen**: Mit dieser Eigenschaft wird gesteuert, ob der Server automatisch Untergruppen für die automatisch generierten Tags erstellt. Dies ist die Standardeinstellung. Wenn diese Option deaktiviert ist, generiert der Server die Tags des Geräts in einer unstrukturierten Liste ohne jede Gruppierung. Im Serverprojekt werden die resultierenden Tags mit dem Adresswert benannt. Beispielsweise werden die Tag-Namen während des Generierungsprozesses nicht beibehalten.
**Hinweis**: Wenn beim Generieren von Tags durch den Server einem Tag derselbe Name wie einem bestehenden Tag zugewiesen wird, erhöht das System automatisch auf die nächste höchste Nummer, sodass der Tag-Name nicht dupliziert wird. Beispiel: Wenn der Generierungsprozess das Tag "AI22" erstellt, das bereits existiert, wird stattdessen das Tag als "AI23" erstellt.

**Erstellen**: Initiiert die Erstellung automatisch generierter OPC-Tags. Wenn die Konfiguration des Geräts geändert wurde, wird der Treiber durch die Option **Tags erstellen** gezwungen, das Gerät erneut auf mögliche Tag-Änderungen zu bewerten. Ihre Fähigkeit, über die System-Tags aufgerufen zu werden, ermöglicht einer Client-Anwendung das Initiieren der Tag-Datenbankerstellung.
**Hinweis**: **Tags erstellen** ist deaktiviert, wenn die Konfiguration ein Projekt offline bearbeitet.

## Geräteeigenschaften – Blockgrößen




### Coils

**Ausgangs-Coils**: Gibt die Ausgangsblockgröße in Bit an. Coils können von 8 bis 2000 Punkten (Bit) auf einmal gelesen werden. Eine höhere Blockgröße bedeutet, dass mehr Punkte vom Gerät bei einer einzelnen Anfrage gelesen werden. Die Blockgröße kann verringert werden, um Daten aus nicht zusammenhängenden Positionen innerhalb des Geräts zu lesen. Die Standardeinstellung ist 32.

**Eingangs-Coils**: Gibt die Eingangsblockgröße in Bit an. Coils können von 8 bis 2000 Punkten (Bit) auf einmal gelesen werden. Eine höhere Blockgröße bedeutet, dass mehr Punkte vom Gerät bei einer einzelnen Anfrage gelesen werden. Die Blockgröße kann verringert werden, um Daten aus nicht zusammenhängenden Positionen innerhalb des Geräts zu lesen. Die Standardeinstellung ist 32.

### Register

**Interne Register**: Gibt die Blockgröße für interne Register in Bit an. Bis zu 125 Modbus-Standardregister (16 Bit) können auf einmal gelesen werden. Eine höhere Blockgröße bedeutet, dass mehr Registerwerte vom Gerät bei einer einzelnen Anfrage gelesen werden. Die Blockgröße kann verringert werden, um Daten aus nicht zusammenhängenden Positionen innerhalb des Geräts zu lesen. Die Standardeinstellung ist 32.

**Halteregister**: Gibt die Blockgröße für Halteregister in Bit an. Bis zu 125 Modbus-Standardregister (16 Bit) können auf einmal gelesen werden. Eine höhere Blockgröße bedeutet, dass mehr Registerwerte vom Gerät bei einer einzelnen Anfrage gelesen werden. Die Blockgröße kann verringert werden, um Daten aus nicht zusammenhängenden Positionen innerhalb des Geräts zu lesen. Die Standardeinstellung ist 32.
**Achtung:** Wenn die Blockgrößen auf einen Wert über 120 festgelegt werden und ein 32- oder 64-Bit-Datentyp für Tags verwendet wird, kann der Fehler "Ungültige Adresse im Block" auftreten. Um dies zu verhindern, verringern Sie den Wert für die Blockgröße auf 120.

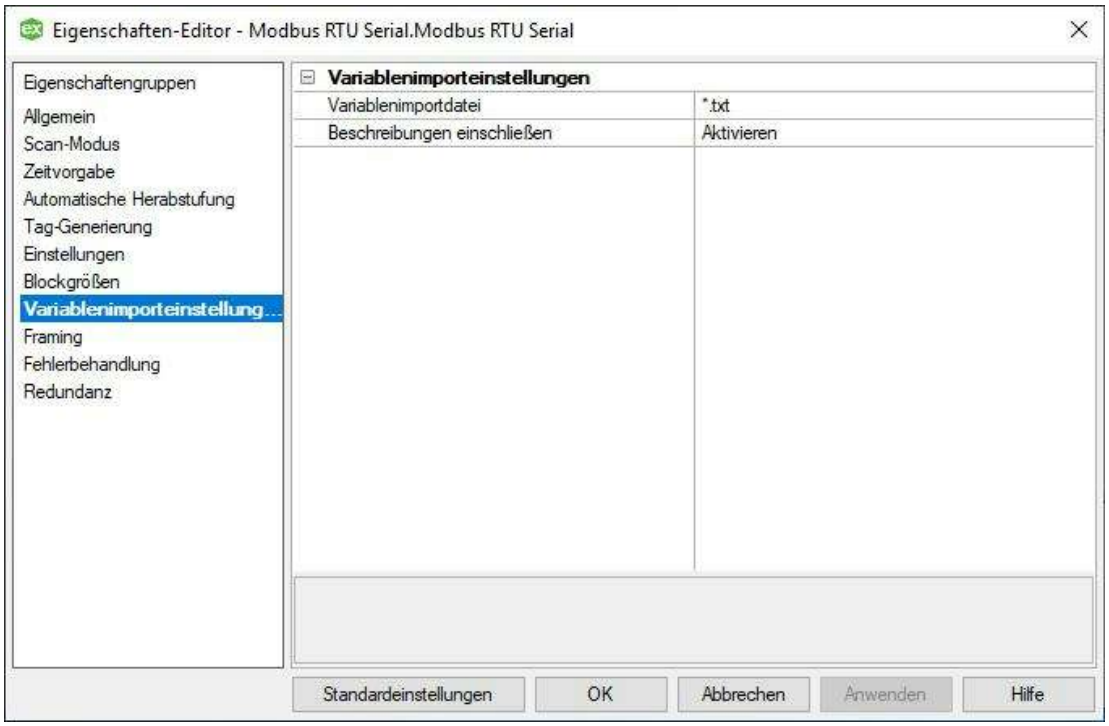
## Blockgrößen

**Zeichenfolgen als Block lesen**: Aktiviert Gruppen-/Block-Lesevorgänge von Zeichenfolgen-Tags, die normalerweise einzeln gelesen werden. Zeichenfolgen-Tags werden je nach ausgewählter Blockgröße gruppiert. Block-Lesevorgänge können nur für Zeichenfolgen-Tags des Modbus-Modells ausgeführt werden. Die Standardeinstellung ist deaktiviert.

## Geräteeigenschaften – Variablenimporteinstellungen

Die Parameter unter "Variablenimporteinstellungen" geben den Speicherort der Variablenimportdatei an, die für die automatische Tag-Datenbankgenerierung verwendet werden soll.

*Weitere Informationen zu CSV-Dateien für Modbus-Treiber finden Sie unter* ***CSV-Dateien für Modbus-Treiber erstellen****.*



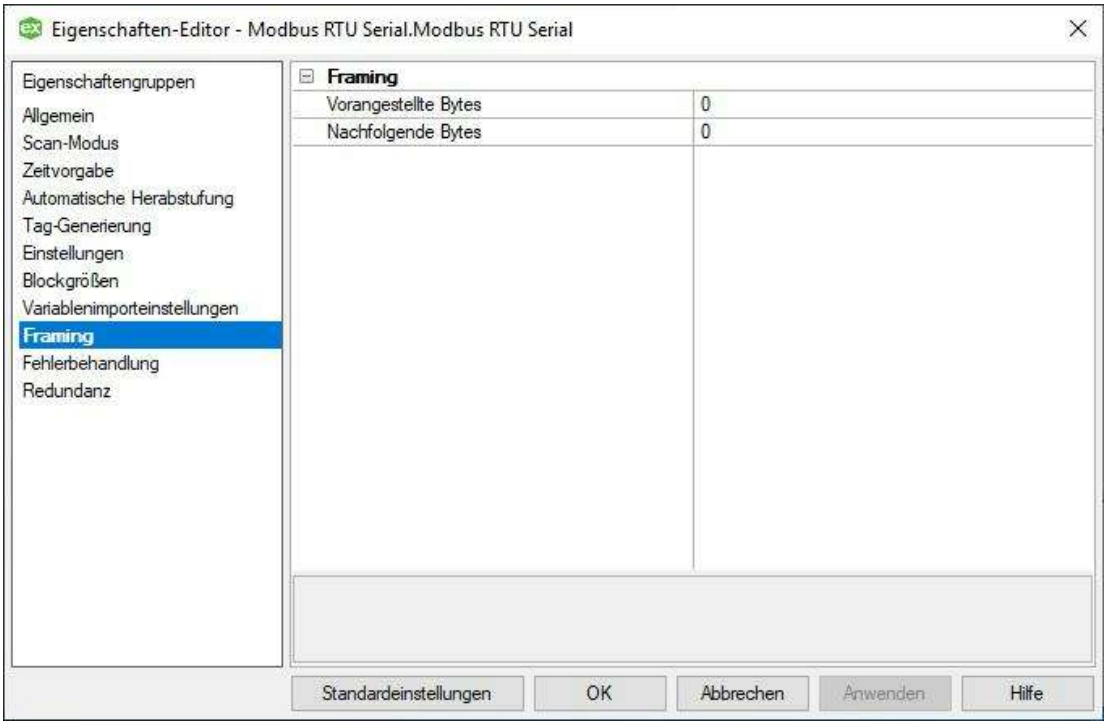

**Variablenimportdatei**: Dieser Parameter wird verwendet, um zum genauen Speicherort der Variablenimportdatei zu navigieren, die für die automatische Tag-Datenbankgenerierung verwendet werden soll.

**Beschreibungen einschließen**: Bei Aktivierung werden importierte Tag-Beschreibungen verwendet, sofern in der Datei vorhanden.

*Weitere Informationen zum Konfigurieren der Funktion zur automatischen Tag-Datenbankgenerierung (und zum Erstellen einer Variablenimportdatei) finden Sie unter* ***Automatische Tag-Datenbankgenerierung****.*

## Geräteeigenschaften – Framing

Einige Terminalserver-Geräte fügen zusätzliche Daten zu Modbus-Frames hinzu. Deshalb können die Framing-Parameter verwendet werden, um den Treiber so zu konfigurieren, dass er die zusätzlichen Bytes in Antwortnachrichten ignoriert.




### Framing

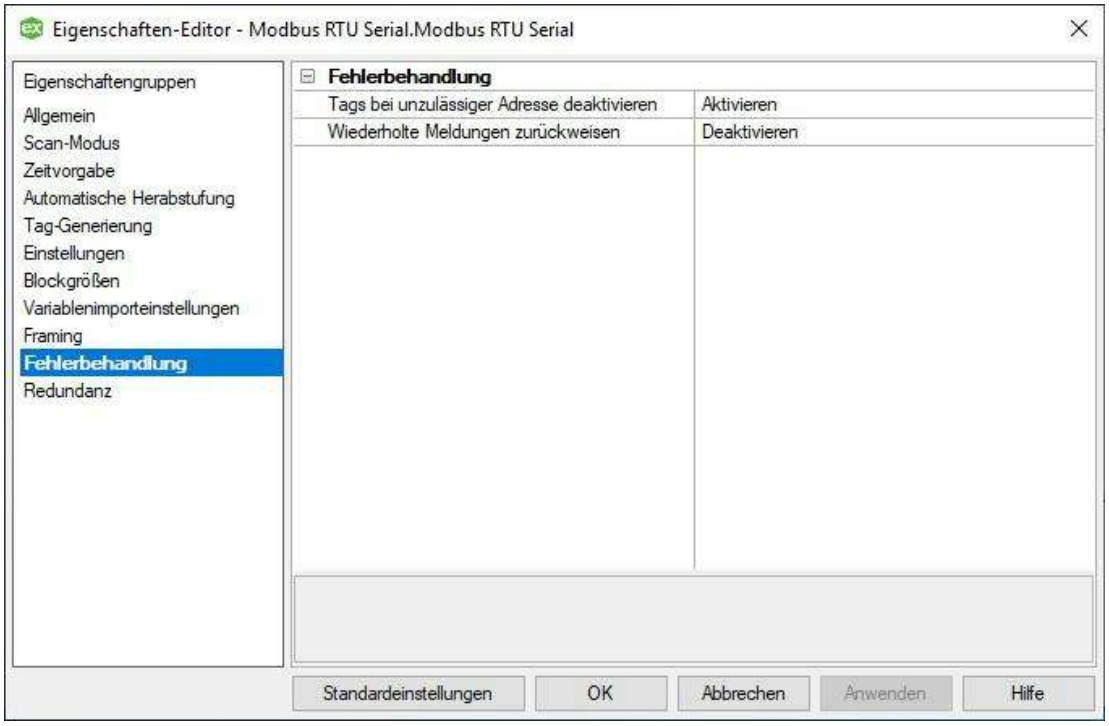
**Modbus TCP Framing**: Wählen Sie **Aktivieren** aus, wenn der Treiber Modbus TCP-Frames mit MBAP-Kopfzeilen verwenden soll. Die Standardeinstellung ist deaktiviert.
**Tipp**: Diese Einstellung sollte für die Kommunikation mit nativen Modbus-TCP-Geräten aktiviert sein.

**Vorangestellte Bytes**: Gibt die Anzahl der Bytes an, die Modbus-Antworten vorangestellt werden sollen. Der zulässige Wertebereich liegt zwischen 0 und 8.

**Nachfolgende Bytes**: Gibt die Anzahl der Bytes an, die Modbus-Antworten nachgestellt werden sollen. Der zulässige Wertebereich liegt zwischen 0 und 8.

## Geräteeigenschaften – Fehlerbehandlung

Die Fehlerbehandlungsparameter bestimmen, wie mit Fehlern vom Gerät umgegangen werden soll.

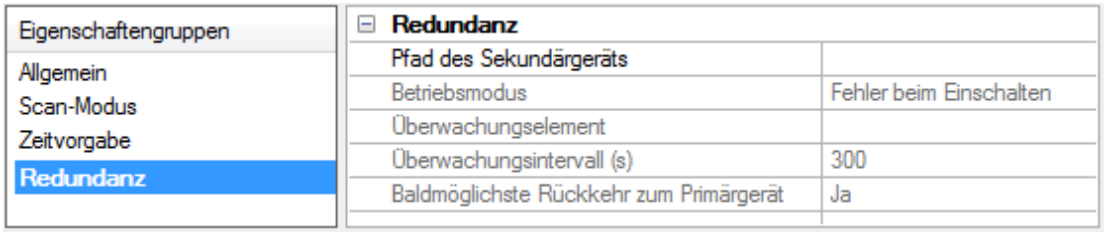



**Tags in unzulässigen Adressen deaktivieren**: Bei Aktivierung hält der Treiber das Abrufen eines Datenblocks an, wenn das Gerät Modbus-Ausnahmecode 2 (unzulässige Adresse) oder 3 (unzulässige Daten, z.B. Anzahl von Punkten) als Antwort auf einen Lesevorgang dieses Blocks zurückgibt. Bei Aktivierung setzt der Treiber das Abrufen des betreffenden Datenblocks fort. Die Standardeinstellung ist aktiviert.

**Wiederholte Meldungen zurückweisen**: Bei Aktivierung interpretiert der Treiber eine wiederholte Meldung als ungültige Antwort und versucht die Anforderung erneut. Die Standardeinstellung ist aktiviert. Bei Deaktivierung erwartet der Treiber wiederholte Meldungen.

**Hinweis:** Einige Meldungsrelais-Ausrüstungen senden Modbus-Anfragen als Echo an den Treiber zurück.

## Geräteeigenschaften – Redundanz




Redundanz steht mit dem Plugin für Redundanz auf Medienebene zur Verfügung.
*Weitere Informationen dazu erhalten Sie auf der Website, von einem Vertriebsrepräsentanten oder im **Benutzerhandbuch**.*

## Automatische Tag-Datenbankgenerierung

Modbus RTU Serial-Treiber nutzt die automatische Tag-Datenbankgenerierung, die es Treibern ermöglicht, automatisch Tags zu erstellen, die auf vom Kontaktplan des Geräts verwendete Datenpunkte zugreifen. Obwohl es manchmal möglich ist, ein Gerät nach Informationen abzufragen, die zur Erstellung einer Tag-Datenbank benötigt werden, muss dieser Treiber stattdessen eine **Variablenimportdatei** verwenden. Variablenimportdateien können mit den Gerätprogrammieranwendungen Concept und ProWORX generiert werden.

### Variablenimportdatei erstellen

Die Importdatei muss im durch Semikola getrennten Textformat (TXT) vorliegen, dem Standard-Exportdateiformat vieler Geräteprogrammieranwendungen.

*Spezifische Informationen zum Erstellen der Variablenimportdatei finden Sie in der englischen Dokumentation "Technical Note 'Creating CSV Files for Modbus Drivers'" (Technical Note zum Erstellen von CSV-Dateien für Modbus-Treiber).*

Dieser Treiber erfordert spezielle Einstellungen zusätzlich zu den grundlegenden Einstellungen, die für alle Treiber gelten, die die automatische Tag-Datenbankgenerierung unterstützen. Zu diesen speziellen Einstellungen gehören Name und Speicherort der Variablenimportdatei, die während des Schritts "Variablenimporteinstellungen" des Gerätassistenten angegeben werden können oder später durch Auswählen des Geräts und anschließendes Auswählen von **Geräteeigenschaften** | **Variablenimporteinstellungen**.

*Weitere Informationen finden Sie unter **Variablenimporteinstellungen**.*

### Serverkonfiguration

Die automatische Tag-Datenbankgenerierung kann so angepasst werden, dass sie den spezifischen Anforderungen der Anwendung entspricht. Die primären Steuerungsoptionen können während des Schritts "Database Creation" des Gerätassistenten angegeben werden oder später durch Auswählen des Geräts und anschließendes Auswählen von **Geräteeigenschaften** | **Tag-Generierung**.

### Operation

Je nach Konfiguration wird die Tag-Generierung automatisch gestartet, sobald das Serverprojekt gestartet wird, oder sie muss zu einem anderen Zeitpunkt manuell initiiert werden. Das Ereignisprotokoll zeigt an, wann der Tag-Generierungsprozess gestartet wurde, ob während der Verarbeitung der Variablenimportdatei Fehler aufgetreten sind und wann der Prozess abgeschlossen wurde.

## Statistikelemente

Statistische Elemente verwenden Daten, die durch zusätzliche, nicht standardmäßig gesammelte Diagnoseinformationen gesammelt wurden. Um statistische Elemente zu verwenden, müssen Kommunikationsdiagnosen aktiviert sein. Um Kommunikationsdiagnosen zu aktivieren, klicken Sie in der Projektansicht mit der rechten Maustaste auf den Kanal, und klicken Sie auf **Eigenschaften | Diagnose aktivieren**. Alternativ können Sie auf den Kanal doppelklicken und **Diagnose aktivieren** auswählen.

### Statistikelemente auf Kanalebene

Die Syntax für Statistikelemente auf Kanalebene lautet <*Kanal*>.*_Statistics*.

**Hinweis**: Statistiken auf Kanalebene sind die Summe der gleichen Elemente auf Geräteebene.

| Element | Datentyp | Zugriff | Beschreibung |
|---|---|---|---|
| _CommFailures | DWord | Lesen/Schreiben | Die Angabe, wie oft die Kommunikation insgesamt fehlgeschlagen ist (oder keine weiteren Wiederholungen möglich waren) |
| _ErrorResponses | DWord | Lesen/Schreiben | Die Gesamtzahl der empfangenem gültigen Fehlerantworten |
| _ExpectedResponses | DWord | Lesen/Schreiben | Die Gesamtzahl der empfangenem erwarteten Antworten |
| _LastResponseTime | String | Schreibgeschützt | Die Zeit, zu der die letzte gültige Antwort empfangen wurde |
| _LateData | DWord | Lesen/Schreiben | Wie oft ein Tag später gelesen wird als erwartet (basierend auf dem angegebenen Scan-Intervall). Aufgrund eines DNR-Fehlerstatus erhöht sich dieser Wert nicht. Ein Tag wird beim ersten Lesen nach einem Kommunikationsverlust nicht als spät gezählt, auch dann nicht, wenn es spät ist. Dieses Verhalten ist beabsichtigt. |
| _MsgResent | DWord | Lesen/Schreiben | Die Gesamtzahl der als Wiederholung gesendeten Nachrichten |
| _MsgSent | DWord | Lesen/Schreiben | Die Gesamtzahl der erstmals gesendeten Nachrichten |
| _MsgTotal | DWord | Schreibgeschützt | Die Gesamtzahl der gesendeten Nachrichten (_MsgSent + _MsgResent) |
| _PercentReturn | Float | Schreibgeschützt | Der prozentuale Anteil der erwarteten Antworten ("Empfangen") an den erstmals gesendeten Nachrichten ("Gesendet") |
| _PercentValid | Float | Schreibgeschützt | Der prozentuale Anteil der insgesamt empfangenen gültigen Antworten (_TotalResponses) an den insgesamt gesendeten Anforderungen (_MsgTotal) |
| _Reset | Bool | Lesen/Schreiben | Setzt alle Diagnosezähler zurück. Schreiben in das Tag _Reset bewirkt, dass alle Diagnosezähler auf dieser Ebene zurückgesetzt werden. |
| _RespBadChecksum | DWord | Lesen/Schreiben | Die Gesamtzahl der Antworten mit Prüfsummenfehlern |

| Element | Datentyp | Zugriff | Beschreibung |
|---|---|---|---|
| _RespTimeouts | DWord | Lesen/Schreiben | Die Gesamtzahl der Nachrichten, für die keinerlei Antwort empfangen wurde |
| _RespTruncated | DWord | Lesen/Schreiben | Die Gesamtzahl der Nachrichten, für die nur eine teilweise Antwort empfangen wurde |
| _TotalResponses | DWord | Schreibgeschützt | Die Gesamtzahl der empfangenen gültigen Antworten (_ErrorResponses + _ExpectedResponses) |

Statistische Elemente werden im Simulationsmodus nicht aktualisiert *(siehe allgemeine Eigenschaften des Geräts)*.

## Statistikelemente auf Geräteebene

Die Syntax für Statistikelemente auf Geräteebene lautet *<Kanal>.<Gerät>._Statistics*.

| Element | Datentyp | Zugriff | Beschreibung |
|---|---|---|---|
| _CommFailures | DWord | Lesen/Schreiben | Die Angabe, wie oft die Kommunikation insgesamt fehlgeschlagen ist (oder keine weiteren Wiederholungen möglich waren) |
| _ErrorResponses | DWord | Lesen/Schreiben | Die Gesamtzahl der empfangenem gültigen Fehlerantworten |
| _ExpectedResponses | DWord | Lesen/Schreiben | Die Gesamtzahl der empfangenem erwarteten Antworten |
| _LastResponseTime | String | Schreibgeschützt | Die Zeit, zu der die letzte gültige Antwort empfangen wurde |
| _LateData | DWord | Lesen/Schreiben | Wie oft ein Tag später gelesen wird als erwartet (basierend auf dem angegebenen Scan-Intervall). Aufgrund eines DNR-Fehlerstatus erhöht sich dieser Wert nicht. Ein Tag wird beim ersten Lesen nach einem Kommunikationsverlust nicht als spät gezählt, auch dann nicht, wenn es spät ist. Dieses Verhalten ist beabsichtigt. |
| _MsgResent | DWord | Lesen/Schreiben | Die Gesamtzahl der als Wiederholung gesendeten Nachrichten |
| _MsgSent | DWord | Lesen/Schreiben | Die Gesamtzahl der erstmals gesendeten Nachrichten |
| _MsgTotal | DWord | Schreibgeschützt | Die Gesamtzahl der gesendeten Nachrichten (_MsgSent + _MsgResent) |
| _PercentReturn | Float | Schreibgeschützt | Der prozentuale Anteil der erwarteten Antworten ("Empfangen") an den erstmals gesendeten Nachrichten ("Gesendet") |
| _PercentValid | Float | Schreibgeschützt | Der prozentuale Anteil der insgesamt empfangenen gültigen Antworten (_TotalResponses) an den insgesamt gesendeten Anforderungen (_MsgTotal) |
| _Reset | Bool | Lesen/Schreiben | Setzt alle Diagnosezähler zurück. Schreiben in das Tag _Reset bewirkt, dass alle Diagnosezähler auf dieser Ebene zurückgesetzt werden. |

| Element | Datentyp | Zugriff | Beschreibung |
|---|---|---|---|
| _RespBa-dChecksum | DWord | Lesen/Schreiben | Die Gesamtzahl der Antworten mit Prüf-summenfehlern |
| _RespTimeouts | DWord | Lesen/Schreiben | Die Gesamtzahl der Nachrichten, für die keinerlei Antwort empfangen wurde |
| _RespTruncated | DWord | Lesen/Schreiben | Die Gesamtzahl der Nachrichten, für die nur eine teilweise Antwort empfangen wurde |
| _TotalResponses | DWord | Schreibgeschützt | Die Gesamtzahl der empfangenen gültigen Ant-worten (_ErrorResponses + _ExpectedResponses) |

Statistische Elemente werden im Simulationsmodus nicht aktualisiert *(siehe allgemeine Eigenschaften des Geräts)*.

## Datentypbeschreibung

| Datentyp | Beschreibung |
|---|---|
| Boolean | Einzelnes Bit |
| Word | 16-Bit-Wert ohne Vorzeichen<br>Bit 0 ist das Low-Bit<br>Bit 15 ist das High-Bit |
| Short | 16-Bit-Wert mit Vorzeichen<br>Bit 0 ist das Low-Bit<br>Bit 14 ist das High-Bit<br>Bit 15 ist das Vorzeichen-Bit |
| DWord | 32-Bit-Wert ohne Vorzeichen<br>Bit 0 ist das Low-Bit<br>Bit 31 ist das High-Bit |
| Long | 32-Bit-Wert mit Vorzeichen<br>Bit 0 ist das Low-Bit<br>Bit 30 ist das High-Bit<br>Bit 31 ist das Vorzeichen-Bit |
| BCD | Gepacktes 2-Byte-BCD<br>Der Wertebereich liegt zwischen 0 und 9999. Für Werte außerhalb dieses Bereichs ist das Verhalten nicht definiert. |
| LBCD | Gepacktes 4-Byte-BCD<br>Der Wertebereich liegt zwischen 0 und 99999999. Für Werte außerhalb dieses Bereichs ist das Verhalten nicht definiert. |
| String | Mit Null beendete ASCII-Zeichenfolge<br>Wird im Modbus-Modell unterstützt, schließt die Byte-Reihenfolgen Hi-Lo und Lo-Hi ein sowie 8-Byte- und 16-Byte-Zeichenfolgendaten für Omni-Mengenumwerter. |
| Double* | 64-Bit-Gleitkommawert<br>Der Treiber interpretiert vier aufeinanderfolgende Register als Wert mit doppelter Genauigkeit, indem die letzten zwei Register als High-DWord und die ersten zwei Register als Low-DWord bewertet werden. |
| Double-Beispiel | Wenn Register 40001 als Double-Wert angegeben wird, ist Bit 0 des Registers 40001 Bit 0 des 64-Bit-Datentyps und Bit 15 des Registers 40004 ist Bit 63 des 64-Bit-Datentyps. |
| Float* | 32-Bit-Gleitkommawert<br>Der Treiber interpretiert zwei aufeinanderfolgende Register als Wert mit einfacher Genauigkeit, indem das erste Register als Low-Wort und das zweite Register als High-Wort bewertet wird. |
| Float-Beispiel | Wenn Register 40001 als Float-Wert angegeben wird, ist Bit 0 des Registers 40001 Bit 0 des 32-Bit-Datentyps und Bit 15 des Registers 40002 ist Bit 31 des 32-Bit-Datentyps. |

*Bei den Beschreibungen wird angenommen, dass standardmäßig für 64-Bit-Datentypen die Datenbehandlung "Erstes DWord 'Low'" verwendet wird und für 32-Bit-Datentypen die Datenbehandlung "Erstes Wort 'Low'".

## Adressbeschreibungen

Adressspezifikationen sind je nach verwendetem Modell unterschiedlich. Wählen Sie einen Link von der folgenden Liste aus, um bestimmte Adressinformationen für das entsprechende Modell zu erhalten.

**Modbus-Adressierung**
**Magnetek-GPD-515-Laufwerksadressierung**
**Elliott-Mengenumwerter-Adressierung**
**Daniels-S500-Mengenumwerter-Adressierung**
**Adressierung dynamischer Flüssigkeitsmessgeräte**
**Omni-Mengenumwerter-Adressierung**
**Statistiken**

**Siehe auch:** **Beschreibung von Funktionscodes**

## Modbus-Adressierung

Die Standard-Datentypen für dynamisch definierte Tags werden **fett** dargestellt. Die Funktionscodes werden als Dezimalzahlen angezeigt.

*Weitere Informationen finden Sie unter* ***Beschreibung von Funktionscodes***.

### Vergleich von 5-stelliger Adressierung mit 6-stelliger Adressierung

In der Modbus-Adressierung gibt die erste Stelle der Adresse die primäre Tabelle an. Die verbleibenden Stellen stellen das Datenelement des Geräts dar. Der Höchstwert des Datenelements ist eine 2-Byte-Ganzzahl ohne Vorzeichen (65.535). Intern sind für diesen Treiber sechs Stellen erforderlich, um die gesamte Adresstabelle und das Element darzustellen. Es ist wichtig zu beachten, dass viele Modbus-Geräte möglicherweise nicht den vollständigen Bereich des Datenelements unterstützen. Um beim Eingeben einer Adresse für ein solches Gerät Verwechslungen zu vermeiden, füllt dieser Treiber die Adresse entsprechend der Eingabe im Adressfeld auf (fügt eine Stelle hinzu). Wenn auf einen primären Tabellentyp bis zu 4 Stellen folgen (z.B. 4x, 4xx, 4xxx oder 4xxxx), wird die Adresse mit zusätzlichen Nullen auf fünf (5) Stellen aufgefüllt. Wenn auf einen primären Tabellentyp fünf (5) Stellen folgen (z.B. 4xxxxx), ändert sich die Adresse nicht. Intern sind Adressen, die als 41, 401, 4001, 40001 oder 400001 eingegeben werden, alle äquivalente Darstellungen einer Adresse, die den primären Tabellentyp 4 und das Datenelement 1 angibt.

| Primäre Tabelle | Beschreibung |
|---|---|
| 0 | Ausgangs-Coils |
| 1 | Eingangs-Coils |
| 3 | Interne Register |
| 4 | Halteregister |

### Modbus-Adressierung im Dezimalformat

| Adresse | Bereich | Datentyp | Zugriff* | Funktionscode |
|---|---|---|---|---|
| Ausgangs-Coils | 000001–065536 | **Boolean** | Lesen/Schreiben | 01, 05, 15 |
| Eingangs-Coils | 100001–165536 | **Boolean** | Schreibgeschützt | 02 |
| Interne Regis- | 300001–365536 | **Word**, Short, | Schreibgeschützt | 04 |

| Adresse | Bereich | Datentyp | Zugriff* | Funktionscode |
|---|---|---|---|---|
| ter | 300001–365535<br>300001–365533<br>3xxxxx.0/1–3xxxxx.15/16*** | BCD<br>Float, DWord, Long, LBCD<br>Double<br>Boolean | | |
| Interne Register als Zeichenfolge mit Byte-Reihenfolge Hi-Lo | 300001.2H–365536.240H<br>.Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte. | **String**** | Schreibgeschützt | 04 |
| Interne Register als Zeichenfolge mit Byte-Reihenfolge Lo-Hi | 300001.2L–365536.240L<br>.Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte. | **String**** | Schreibgeschützt | 04 |
| Halteregister | 400001–465536<br>400001–465535<br>400001–465533<br>4xxxxx.0/1–4xxxxx.15/16 *** | **Word**, Short, BCD<br>Float, DWord, Long, LBCD<br>Double<br>Boolean | Lesen/Schreiben | 03, 06, 16<br>03, 06, 16, 22 |
| Halteregister als Zeichenfolge mit Byte-Reihenfolge Hi-Lo | 400001.2H–465536.240H<br>.Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte. | **String**** | Lesen/Schreiben | 03,·16 |
| Halteregister als Zeichenfolge mit Byte-Reihenfolge Lo-Hi | 400001.2L–465536.240L<br>.Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte. | **String**** | Lesen/Schreiben | 03,·16 |

*Alle Lesen/Schreiben-Adressen können als lesegeschützt festgelegt werden, indem das Präfix "W" vor die Adresse gesetzt wird, z.B. "W40001." Dadurch wird verhindert, dass der Treiber das Register an der angegebenen Adresse liest. Versuche durch den Client, ein lesegeschütztes Tag zu lesen, führen dazu, dass der letzte erfolgreich in die angegebene Adresse geschriebene Wert abgerufen wird. Wenn keine erfolgreichen Schreibvorgänge stattgefunden haben, empfängt der Client 0 bzw. NULL für numerische bzw. Zeichenfolgenwerte als Anfangswert.

**Achtung:** Wenn die Client-Zugriffsrechte für lesegeschützte Tags auf "Lesegeschützt" festgelegt werden, führt dies dazu, dass Schreibvorgänge in diesen Tags fehlschlagen und der Client immer 0 bzw. NULL für numerische bzw. Zeichenfolgenwerte empfängt.

- ***Weitere Informationen finden Sie unter **Zeichenfolgenunterstützung**.*
- ****Weitere Informationen finden Sie unter Nullbasierte Bit-Adressierung unter **Einstellungen**.*

## Modbus-Adressierung im Hexadezimalformat

| Adresse | Bereich | Datentyp | Zugriff | Funktionscode |
|---|---|---|---|---|
| Ausgangs-Coils | H000001–H0FFFF | **Boolean** | Lesen/Schreiben | 01, 05, 15 |
| Eingangs-Coils | H100001–H1FFFF | **Boolean** | Schreibgeschützt | 02 |
| Interne Register | H300001–H310000<br>H300001–H3FFFF<br>H300001–H3FFFD<br>H3xxxxx.0/1–H3xxxxx.F/10* | **Word**, Short, BCD<br>Float, DWord, Long, LBCD<br>Double<br>Boolean | Schreibgeschützt | 04 |
| Interne Register als Zeichenfolge mit Byte-Reihenfolge Hi-Lo | H300001.2H–H3FFFF.240H.<br>Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte. | **String**** | Schreibgeschützt | 04 |
| Interne Register als Zeichenfolge mit Byte-Reihenfolge Lo-Hi | H300001.2L–H3FFFF.240L.<br>Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte. | **String**** | Schreibgeschützt | 04 |
| Halteregister | H400001–H410000<br>H400001–H4FFFF<br>H400001–H4FFFD<br>H4xxxxx.0/1–H4xxxxx.F/10* | **Word**, Short, BCD, Float, DWord, Long, LBCD, Double, Boolean | Lesen/Schreiben | 03, 06, 16<br>03, 06, 16, 22 |
| Halteregister als Zeichenfolge mit Byte-Reihenfolge Hi-Lo | H400001.2H–H4FFFF.240H.<br>Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte. | **String**** | Lesen/Schreiben | 03,·16 |
| Halteregister als Zeichenfolge mit Byte-Reihenfolge Lo-Hi | H400001.2L–H4FFFF.240L.<br>Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte. | **String**** | Lesen/Schreiben | 03,·16 |

- **Weitere Informationen finden Sie unter Nullbasierte Bit-Adressierung unter **Einstellungen**.*
- ***Weitere Informationen finden Sie unter **Zeichenfolgenunterstützung**.*

## Zeichenfolgenunterstützung

Das Modbus-Modell unterstützt das Lesen und Schreiben im Halteregisterspeicher als ASCII-Zeichenfolge. Bei Verwendung von Halteregistern für Zeichenfolgendaten enthält jedes Register zwei Byte ASCII-Daten. Die Reihenfolge der ASCII-Daten innerhalb eines gegebenen Registers kann beim Definieren der Zeichenfolge ausgewählt werden. Die Länge der Zeichenfolge kann zwischen 2 und 240 Byte liegen und wird statt einer Bit-Nummer eingegeben. Die Länge muss als gerade Zahl eingegeben werden. Geben Sie die Byte-Reihenfolge an, indem Sie entweder ein "H" oder ein "L" an die Adresse anhängen.

## Zeichenfolgenbeispiele

1. Um eine Zeichenfolge zu adressieren, die bei 40200 beginnt sowie eine Länge von 100 Byte und die Byte-Reihenfolge Hi-Lo aufweist, geben Sie "40200.100H" ein.
2. Um eine Zeichenfolge zu adressieren, die bei 40500 beginnt sowie eine Länge von 78 Byte und die Byte-Reihenfolge Lo-Hi aufweist, geben Sie "40500.78L" ein.

**Hinweis:** Die Länge der Zeichenfolge kann durch die maximale Größe der Schreibanforderung begrenzt werden, die das Gerät zulässt. Wenn bei Verwendung eines Zeichenfolgen-Tags die Fehlermeldung "In Adresse <Adresse> auf Gerät <Gerät> kann nicht geschrieben werden: Gerät hat mit Ausnahmecode 3 geantwortet." im Server-Ereignisfenster empfangen wird, bedeutet das, dass die Länge der Zeichenfolge vom Gerät abgelehnt wurde. Kürzen Sie nach Möglichkeit die Zeichenfolge.

## Normale Adressbeispiele

1. Das 255. Ausgangs-Coil würde bei dezimaler Adressierung als "0255" oder bei hexadezimaler Adressierung als "H0FF" adressiert.
2. In der Dokumentation werden Modbus-Adressen manchmal mit Funktionscode und Position angegeben. Funktionscode 3, Position 2000 wird beispielsweise als "42000" oder "H47D0" adressiert. Die vorangestellte "4" stellt Halteregister oder den Funktionscode 3 dar.
3. In der Dokumentation werden Modbus-Adressen manchmal mit Funktionscode und Position angegeben. Funktionscode 5, Position 100 wird beispielsweise als "0100" oder "H064" adressiert. Die vorangestellte "0" stellt Ausgangs-Coils oder den Funktionscode 5 dar. Durch Schreiben von 1 oder 0 an dieser Adresse wird die Coil eingestellt bzw. zurückgesetzt.

## Array-Unterstützung

Arrays werden für interne und Halteregister-Positionen für alle Datentypen außer Boolean und String unterstützt. Arrays werden auch für Eingangs- und Ausgangs-Coils unterstützt (Boolean-Datentypen). Es gibt zwei Methoden, um ein Array zu adressieren. In den folgenden Beispielen werden Halteregister-Positionen verwendet:

4xxxx [Zeilen] [Spalten]
4xxxx [Spalten] Bei dieser Methode wird angenommen, dass "Zeilen" gleich 1 ist.

Bei Arrays darf das Produkt aus "Zeilen" und "Spalten" nicht die Blockgröße überschreiten, die dem Gerät für den Register- bzw. Coil-Typ zugewiesen wurde. Bei Register-Arrays von 32-Bit-Datentypen darf das Produkt aus "Zeilen" und "Spalten", multipliziert mit 2, nicht die Blockgröße überschreiten.

## Gepackter Coil-Adresstyp

Der gepackte Coil-Adresstyp ermöglicht den Zugriff auf mehrere aufeinanderfolgende Coils als analogen Wert. Diese Funktion ist sowohl für Eingangs-Coils als auch für Ausgangs-Coils verfügbar, und zwar nur im Abfragemodus. Der einzige gültige Datentyp ist Word. Die Syntax lautet:

Ausgangs-Coils: 0*xxxxx#nn* Word Lesen/Schreiben
Eingangs-Coils: 1*xxxxx#nn* Word schreibgeschützt

Wobei *xxxxx* für die Adresse der ersten Coil steht (dezimale und hexadezimale Werte zulässig) und *nn* für die Anzahl der Coils, die in einen analogen Wert gepackt werden sollen (1–16, nur dezimal).

Die Bit-Reihenfolge ist so, dass die Startadresse das niedrigstwertige Bit (LSB, Least Significant Bit) des analogen Werts ist.

## Magnetek-GPD-515-Laufwerksadressierung

Diese Tabelle enthält die allgemeinen Bereiche der Daten, die vom Magnetek-GPD-515-Laufwerk verfügbar sind. Informationen dazu, wie mit der Modbus-RTU-Adressierung auf bestimmte Laufwerksparameter zugegriffen werden kann, finden Sie im englischen Handbuch Magnetek Modbus RTU Technical Manual (technisches Handbuch für Magnetek Modbus RTU), Teilenummer TM4025. Stellen Sie der gewünschten Adresse in jedem Fall den Buchstaben H (der für hexadezimale Adressierung steht) voran. Die Standard-Datentypen für dynamisch definierte Tags werden ggf. **fett** dargestellt.

### Magnetek-GPD-515-Adressierung im Hexadezimalformat

| Adresse | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| Befehlsregister-Zugriff auf Bit-Ebene | H40001–H4000F<br>H4xxxx.0/1–H4xxxx.F/10* | **Word**, Short<br>Boolean | Lesen/Schreiben |
| Monitor-Registerzugriff auf Bit-Ebene | H40010–H4001A<br>H4xxxx.0/1–H4xxxx.F/10* | **Word**, Short<br>Boolean | Schreibgeschützt |
| Laufwerksparameter-Registerzugriff (nur Monitor) auf Bit-Ebene | H40020–H40097<br>H4xxxx.0/1–H4xxxx.F/10* | **Word**, Short<br>Boolean | Schreibgeschützt |
| Laufwerksparameter-Registerzugriff auf Bit-Ebene | H40100–H4050D<br>H4xxxx.0/1–H4xxxx.F/10* | **Word**, Short<br>Boolean | Lesen/Schreiben |
| Spezielle Register | H4FFDD ACCEPT H4FFFD ENTER | Word, Short | Lesegeschützt |

*Weitere Informationen finden Sie unter Nullbasierte Bit-Adressierung unter* ***Einstellungen****.*

### Beispiel

Um auf den Operationsstatus des Treibers, Adresse 02BH, zuzugreifen, geben Sie die folgende Adresse ein: H4002B.

**Hinweis:** Beim Hinzufügen eines Magnetek-Geräts zum OPC-Server-Projekt müssen Benutzer darauf achten, dass die Einstellung Nullbasierte Adressierung deaktiviert ist. Wenn dieser Parameter nicht richtig eingestellt ist, versetzt der Modbus-RTU-Treiber alle Magnetek-Adressen um 1.

### Array-Unterstützung

Arrays werden für Halteregister-Positionen unterstützt, und zwar für alle Datentypen außer Boolean. Es gibt zwei Methoden, um ein Array zu adressieren. In den folgenden Beispielen werden Halteregister-Positionen verwendet:

4xxxx [Zeilen] [Spalten]
4xxxx [Spalten] Bei dieser Methode wird angenommen, dass "Zeilen" gleich 1 ist.

Das Produkt aus "rows" und "cols" darf nicht die Blockgröße überschreiten, die dem Gerät für den Registertyp zugewiesen wurde.

## Elliott-Mengenumwerter-Adressierung

Die Standard-Datentypen für dynamisch definierte Tags werden ggf. **fett** dargestellt.

| Adresse | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| Ausgangs-Coils | 000001–065536 | **Boolean** | Lesen/Schreiben |
| Eingangs-Coils | 100001–165536 | **Boolean** | Schreibgeschützt |
| Interne Register | 300001–365536<br>300001–365535<br>3xxxxx.0/1–3xxxxx.15/16* | **Word**, Short, BCD<br>Float, DWord, Long, LBCD<br>Boolean | Schreibgeschützt |
| Halteregister | 400001–465536<br>400001–465535<br>4xxxxx.0/1–4xxxxx.15/16* | **Word**, Short, BCD**<br>Float, DWord, Long, LBCD<br>Boolean | Lesen/Schreiben |

*Weitere Informationen finden Sie unter Nullbasierte Bit-Adressierung unter **Einstellungen**.*

**Die Adressbereiche 405001 bis 405315 und 407001 bis 407315 sind 32-Bit-Register. Adressen im Bereich 405001 bis 405315 verwenden den Standard-Datentyp Long.

Adressen im Bereich 407001 bis 407315 verwenden den Standard-Datentyp Float. Da diese Adressregister 32-Bit-Register sind, sind nur die Datentypen Float, DWord, Long und LBCD zulässig. Arrays sind nicht zulässig.

### Array-Unterstützung

Arrays werden für interne und Halteregister-Positionen unterstützt, und zwar für alle Datentypen außer Boolean. Es gibt zwei Methoden, um ein Array zu adressieren. In den folgenden Beispielen werden Halteregister-Positionen verwendet:

4xxxx [Zeilen] [Spalten]
4xxxx [cols]: Bei dieser Methode wird angenommen, dass "rows" gleich 1 ist.

Das Produkt aus "rows" und "cols" darf nicht die Blockgröße überschreiten, die dem Gerät für den Registertyp zugewiesen wurde. Bei Arrays von 32-Bit-Datentypen darf das Produkt aus "rows" und "cols", multipliziert mit 2, nicht die Blockgröße überschreiten.

## Daniels-S500-Mengenumwerter-Adressierung

Die Standard-Datentypen für dynamisch definierte Tags werden ggf. **fett** dargestellt. Die Funktionscodes werden als Dezimalzahlen angezeigt.

*Weitere Informationen finden Sie unter **Beschreibung von Funktionscodes**.*

| Adresse | Hexadezimalbereich | Dezimalbereich | Datentyp | Funktionscodes | Zugriff |
|---|---|---|---|---|---|
| Summen | 000–0FF | 4096–4351 | Double | 03 | Schreibgeschützt |
| Berechnete/gemessene Variablen | 100–24F | 4352–4687 | Float | 03, 16 | Lesen/Schreiben |
| Berech- | 250–28F | 4688–4751 | Float | 03, 16 | Lesen/Schrei- |

| Adresse | Hexa-dezimalbereich | Dezi-malbereich | Daten-typ | Funk-tionscodes | Zugriff |
|---|---|---|---|---|---|
| nungskonstanten | | | | | ben |
| Tasta-turstandardwerte | 290–2AF | 4752–4783 | Float | 03, 16 | Lesen/Schrei-ben |
| Alarm- und Ska-lierungskonstanten | 2B0–5FF | 4784–5631 | Float | 03, 16 | Lesen/Schrei-ben |
| Status/Steuerung | 700–7FF | 5888–6143 | **Boolean** | 01, 05 | Lesen/Schrei-ben |
| Alarme | 800–FFF | 6144–8191 | **Boolean** | 02 | Schreib-geschützt |

## Adressierung dynamischer Flüssigkeitsmessgeräte

Die Standard-Datentypen für dynamisch definierte Tags werden ggf. **fett** dargestellt.

### Adressierung dynamischer Flüssigkeitsmessgeräte im Dezimalformat

| Adresse | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| Halteregister (16 Bit) | 400000–407000<br>400000–406999<br>408001–465535<br>408001–465534<br>4xxxxx.0/1–4xxxxx.15/16* | **Word**, Short, BCD<br>Float, DWord, Long, LBCD<br>**Word**, Short, BCD<br>Float, DWord, Long, LBCD<br>Boolean | Lesen/Schreiben |
| Halteregister (32 Bit) | 407001–408000 | **Float** | Lesen/Schreiben |
| Halteregister als Zeichenfolge mit Byte-Reihenfolge Hi-Lo | 400000.2H–407000.240H<br>408001.2H–465535.240H<br>.Bit ist die Zei-chenfolgenlänge, Bereich 2 bis 240 Byte. | **String** | Lesen/Schreiben |
| Halteregister als Zeichenfolge mit Byte-Reihenfolge Lo-Hi | 400000.2L–407000.240L<br>408001.2L–465535.240L<br>.Bit ist die Zei-chenfolgenlänge, Bereich 2 bis 240 Byte. | **String** | Lesen/Schreiben |

*Weitere Informationen finden Sie unter Nullbasierte Bit-Adressierung unter* ***Einstellungen****.*

### Adressierung dynamischer Flüssigkeitsmessgeräte im Hexadezimalformat

| Adresse | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| Halteregister (16 Bit) | H400000–H401B58<br>H400000–H401B57 | **Word**, Short, BCD | Lesen/Schreiben |

| Adresse | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| | H401F41–H40FFFF<br>H401F41–H40FFFE<br>H4xxxxx.0/1–H4xxxxx.F/10* | Float, DWord, Long, LBCD<br>**Word**, Short, BCD<br>Float, DWord, Long, LBCD<br>Boolean | |
| Halteregister (32 Bit) | H401B59–H401F40 | **Float** | Lesen/Schreiben |
| Halteregister als Zeichenfolge mit Byte-Reihenfolge Hi-Lo | H400000.2H–H401B58.240H<br>H401F41.2H–H40FFFF.240H<br>.Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte. | **String** | Lesen/Schreiben |
| Halteregister als Zeichenfolge mit Byte-Reihenfolge Lo-Hi | H400000.2L–H401B58.240L<br>H401F41.2L–H0FFFF.240L<br>.Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte. | **String** | Lesen/Schreiben |

*Weitere Informationen finden Sie unter Nullbasierte Bit-Adressierung unter **Einstellungen**.*

**Hinweis**: Bei diesem Treiber müssen alle Adressen mit der Ziffer "4" für das Modell dynamischer Flüssigkeitsmessgeräte beginnen. Diese 4 wird in der Dokumentation dynamischer Flüssigkeitsmessgeräte nicht immer explizit ausgeschrieben. Der Benutzer kann dort beispielsweise einen Verweis auf die Einheiten-ID an Adresse 3001 vorfinden. Dieser Wert muss im Server als "403001" adressiert werden.

## Zeichenfolgenunterstützung

Das Modell dynamischer Flüssigkeitsmessgeräte unterstützt das Lesen und Schreiben im Halteregisterspeicher als ASCII-Zeichenfolge. Bei Verwendung von Halteregistern für Zeichenfolgendaten enthält jedes Register zwei Byte ASCII-Daten. Die Reihenfolge der ASCII-Daten innerhalb eines gegebenen Registers kann beim Definieren der Zeichenfolge ausgewählt werden. Die Länge der Zeichenfolge kann zwischen 2 und 240 Byte liegen und wird statt einer Bit-Nummer eingegeben. Die Länge muss als gerade Zahl eingegeben werden. Geben Sie die Byte-Reihenfolge an, indem Sie entweder ein "H" oder ein "L" an die Adresse anhängen.

## Zeichenfolgenbeispiele

1. Um eine Zeichenfolge zu adressieren, die bei 40200 beginnt sowie eine Länge von 100 Byte und die Byte-Reihenfolge Hi-Lo aufweist, geben Sie "40200.100H" ein.
2. Um eine Zeichenfolge zu adressieren, die bei 40500 beginnt sowie eine Länge von 78 Byte und die Byte-Reihenfolge Lo-Hi aufweist, geben Sie "40500.78L" ein.

**Hinweis**: Die Länge der Zeichenfolge kann durch die maximale Größe der Schreibanforderung begrenzt werden, die das Gerät zulässt. Wenn bei Verwendung eines Zeichenfolgen-Tags die Fehlermeldung "In Adresse <Adresse> auf dem Gerät <Gerät> kann nicht geschrieben werden: Gerät hat mit Ausnahmecode 3 geantwortet." im Server-Ereignisfenster empfangen wird, bedeutet das, dass die Länge der Zeichenfolge vom Gerät abgelehnt wurde. Kürzen Sie nach Möglichkeit die Zeichenfolge.

## Omni-Mengenumwerter-Adressierung

Die Standard-Datentypen für dynamisch definierte Tags werden **fett** dargestellt.

| Adresse | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| Digitaler E/A-Punkt | 1001–1024 | **Boolean** | Lesen/Schreiben |
| Programmierbarer Boolean-Punkt | 1025–1088 | **Boolean** | Lesen/Schreiben |
| Status- und Alarmpunkte für Messläufe | 1n01–001n59<br>1n76–1n99<br>n = Nummer des Messlaufs | **Boolean** | Lesen/Schreiben |
| Alarm-Statuspunkte für Mikro-bewegungen | 1n60–1n75<br>n = Nummer des Messlaufs | **Boolean** | Lesen/Schreiben |
| Boolean-Punkte für Benutzer-Notiz-blöcke | 1501–1599<br>1601–1649 | **Boolean** | Lesen/Schreiben |
| Einmalige Punkte für Benutzer-Notiz-blöcke | 1650–1699 | **Boolean** | Lesen/Schreiben |
| Boolean-Punkte bzw. -Variablen für Befehle | 1700–1798 | **Boolean** | Lesen/Schreiben |
| Alarm- und Statuspunkte für Mess-gerät-Stationen | 1801–1899 | **Boolean** | Lesen/Schreiben |
| Alarm- und Statuspunkte für Prüfgeräte | 1901–1967 | **Boolean** | Lesen/Schreiben |
| Messgerätzähler-Rollover-Flags | 2n01–2n37<br>n = Nummer des Messlaufs | **Boolean** | Lesen/Schreiben |
| Versch. Alarm und Status für Mess-gerät-Stationen | 2601–2623 | **Boolean** | Lesen/Schreiben |
| Stationszähler-Rollover-Flags | 2801–2851 | **Boolean** | Lesen/Schreiben |
| Stationszähler-Dezimalauflösung | 2852–2862<br>2865–2999 | **Boolean** | Lesen/Schreiben |

| Adressen für 16-Bit-Ganz-zahldaten | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| Benutzerdefiniertes Datenpaket Nr. 1 | 3001–3040 | **Short**, Word, BCD | Lesen/Schreiben |
| Benutzerdefiniertes Datenpaket Nr. 2 | 3041–3056 | **Short**, Word, BCD | Lesen/Schreiben |
| Benutzerdefiniertes Datenpaket Nr. 3 | 3057–3096 | **Short**, Word, BCD | Lesen/Schreiben |
| Versch. 16-Bit-Ganzzahldaten | 3097–3099<br>3737–3799<br>3875–3899 | **Short**, Word, BCD | Lesen/Schreiben |
| 16-Bit-Ganzzahldaten für Messläufe | 3n01–3n52<br>n = Nummer des Messlaufs | **Short**, Word, BCD | Lesen/Schreiben |
| 16-Bit-Ganzzahldaten für Notizblöcke | 3501–3599 | **Short**, Word, BCD | Lesen/Schreiben |

| Adressen für 16-Bit-Ganzzahldaten | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| Benutzeranzeige Nr. 1 | 3601–3608 | **Short**, Word, BCD | Lesen/Schreiben |
| Benutzeranzeige Nr. 2 | 3609–3616 | **Short**, Word, BCD | Lesen/Schreiben |
| Benutzeranzeige Nr. 3 | 3617–3624 | **Short**, Word, BCD | Lesen/Schreiben |
| Benutzeranzeige Nr. 4 | 3625–3632 | **Short**, Word, BCD | Lesen/Schreiben |
| Benutzeranzeige Nr. 5 | 3633–3640 | **Short**, Word, BCD | Lesen/Schreiben |
| Benutzeranzeige Nr. 6 | 3641–3648 | **Short**, Word, BCD | Lesen/Schreiben |
| Benutzeranzeige Nr. 7 | 3649–3656 | **Short**, Word, BCD | Lesen/Schreiben |
| Benutzeranzeige Nr. 8 | 3657–3664 | **Short**, Word, BCD | Lesen/Schreiben |
| Zugriff auf Rohdaten-Archivdatensätze | 3701–3736 | **Short**, Word, BCD | Lesen/Schreiben |
| 16-Bit-Ganzzahldaten für Messgerät-Stationen | 3800–3842 | **Short**, Word, BCD | Lesen/Schreiben |
| Batch-Sequenz für Messgerät Nr. 1 | 3843–3848 | **Short**, Word, BCD | Lesen/Schreiben |
| Batch-Sequenz für Messgerät Nr. 2 | 3849–3854 | **Short**, Word, BCD | Lesen/Schreiben |
| Batch-Sequenz für Messgerät Nr. 3 | 3855–3860 | **Short**, Word, BCD | Lesen/Schreiben |
| Batch-Sequenz für Messgerät Nr. 4 | 3861–3866 | **Short**, Word, BCD | Lesen/Schreiben |
| Zeit/Datum für Mengenumwerter | 3867–3874 | **Short**, Word, BCD | Lesen/Schreiben |
| 16-Bit-Ganzzahldaten für Prüfgeräte | 3901–3999 | **Short**, Word, BCD | Lesen/Schreiben |

| ASCII-Zeichenfolgendaten (8 Zeichen) | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| ASCII-Daten für Messläufe | 4n01–4n39<br>n = Nummer des Messlaufs | **String** | Lesen/Schreiben |
| ASCII-Daten für Notizblöcke | 4501–4599 | **String** | Lesen/Schreiben |
| Benutzeranzeige-Definitionsvariablen | 4601–4640 | **String** | Lesen/Schreiben |
| Eingabe-Hilfsvariablen für Stationen | 4707–4710 | **String** | Lesen/Schreiben |
| ASCII-Daten für Messgerät-Stationen | 4801–4851 | **String** | Lesen/Schreiben |

| ASCII-Zeichenfolgendaten (8 Zeichen) | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| Batch-ID für Messgerät Nr. 1 | 4852–4863 | **String** | Lesen/Schreiben |
| Batch-ID für Messgerät Nr. 2 | 4864–4875 | **String** | Lesen/Schreiben |
| Batch-ID für Messgerät Nr. 3 | 4876–4887 | **String** | Lesen/Schreiben |
| Batch-ID für Messgerät Nr. 4 | 4888–4899 | **String** | Lesen/Schreiben |
| ASCII-Zeichenfolgendaten für Prüfgeräte | 4901–4942 | **String** | Lesen/Schreiben |

| 32-Bit-Ganzzahldaten | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| 32-Bit-Ganzzahldaten für Messläufe | 5n01–5n99<br>n = Nummer des Messlaufs | **Long**, DWord, LBCD, Float | Lesen/Schreiben |
| 32-Bit-Ganzzahldaten für Notizblöcke | 5501–5599 | **Long**, DWord, LBCD, Float | Lesen/Schreiben |
| 32-Bit-Ganzzahldaten für Stationen | 5801–5818 | **Long**, DWord, LBCD, Float | Lesen/Schreiben |
| Batch-Größe für Messgerät Nr. 1 | 5819–5824 | **Long**, DWord, LBCD, Float | Lesen/Schreiben |
| Batch-Größe für Messgerät Nr. 2 | 5825–5830 | **Long**, DWord, LBCD, Float | Lesen/Schreiben |
| Batch-Größe für Messgerät Nr. 3 | 5831–5836 | **Long**, DWord, LBCD, Float | Lesen/Schreiben |
| Batch-Größe für Messgerät Nr. 4 | 5837–5842 | **Long**, DWord, LBCD, Float | Lesen/Schreiben |
| Zusätzliche 32-Bit-Daten für Messläufe | 5843–5899 | **Long**, DWord, LBCD, Float | Lesen/Schreiben |
| 32-Bit-Ganzzahldaten für Prüfgeräte | 5901–5973 | **Long**, DWord, LBCD, Float | Lesen/Schreiben |
| TDVOL/TDFMP-Impulse für kompakte Prüfgeräte | 5974–5999 | **Long**, DWord, LBCD, Float | Lesen/Schreiben |

| 32-Bit-IEEE-Gleitkommadaten | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| Reservierte Daten | 6001–7000 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Digital-Analog-Ausgaben | 7001–7024 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Benutzervariablen | 7025–7088 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Programmierbarer Akkumulator | 7089-7099 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Daten für Messläufe | 7n01–7n99<br>n = Nummer des Messlaufs | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Notizblock-Daten | 7501–7599 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |

| 32-Bit-IEEE-Gleitkommadaten | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| PID-Steuerdaten | 7601–7623 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Verschiedene Daten für Messläufe | 7624–7699 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Verschiedene Variablen | 7701–7799 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Messgerät-Stationsdaten | 7801–7899 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Prüfgerätdaten | 7901–7918 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Konfigurationsdaten für Prüfgeräte | 7919–7958 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Letzte Prüfdaten | 7959–7966 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Während der Prüfung zurückgewiesene Daten | 7967–7990 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Daten für Prüfgerätläufe | 7991–8050 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Durchschnittsdaten für Prüfläufe | 8051–8079 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Client-Messgerätdaten für Prüfläufe | 8080–8199 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Prüfreihen-Daten | 8200–8223 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Daten des geprüften Messgeräts | 8224–8230 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Massen-Prüfdaten | 8231–8500 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Verschiedene Daten für Messlauf Nr. 1 | 8501–8599 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Verschiedene Daten für Messlauf Nr. 2 | 8601–8699 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Verschiedene Daten für Messlauf Nr. 3 | 8701–8799 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Verschiedene Daten für Messlauf Nr. 4 | 8801–8899 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |
| Durchschnittsdaten der Station für vorherige Batches | 8901–8999 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |

| 16-Bit-Ganzzahldaten für Konfigurationen | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| Messlauf Nr. 1 | 13001–13013 | **Short**, Word, BCD | Lesen/Schreiben |
| Messlauf Nr. 2 | 13014–13026 | **Short**, Word, | Lesen/Schreiben |

| 16-Bit-Ganzzahldaten für Konfigurationen | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| | | BCD | |
| Messlauf Nr. 3 | 13027–13039 | **Short**, Word, BCD | Lesen/Schreiben |
| Messlauf Nr. 4 | 13040–13052 | **Short**, Word, BCD | Lesen/Schreiben |
| Prüfgerät-Konfiguration | 13053–13073 | **Short**, Word, BCD | Lesen/Schreiben |
| Allgemeine Flusskonfiguration | 13074–13084 | **Short**, Word, BCD | Lesen/Schreiben |
| Konfiguration des seriellen Anschlusses | 13085–13128 | **Short**, Word, BCD | Lesen/Schreiben |
| PID-Konfiguration | 13129–13160 | **Short**, Word, BCD | Lesen/Schreiben |
| SPS-Daten | 13161–13299 | **Short**, Word, BCD | Lesen/Schreiben |
| Peer-to-Peer-Setup | 13300–13499 | **Short**, Word, BCD | Lesen/Schreiben |
| Rohdatenarchiv | 13500–13999 | **Short**, Word, BCD | Lesen/Schreiben |

| ASCII-Zeichenfolgendaten (16 Zeichen) | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| Mengenumwerter-Konfiguration | 14001–14499 | **String** | Lesen/Schreiben |

| 32-Bit-Ganzzahldaten | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| Mengenumwerter-Konfiguration | 15001–16999 | **Long**, DWord, LBCD, Float | Lesen/Schreiben |

| 32-Bit-IEEE-Gleitkommadaten | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| Mengenumwerter-Konfiguration | 17001–18999 | **Float**, Long, DWord, LBCD | Lesen/Schreiben |

### Unterstützte erweiterte Omni-Typen

**Benutzerdefinierte Pakete**
**Rohdatenarchiv**
**Textberichte**
**Textarchiv**

## Benutzerdefinierte Omni-Datenpakete

Der Omni-Mengenumwerter ermöglicht es Benutzern, verschiedene Speicherbereiche einer einzelnen Datenstruktur zuzuordnen, die mit einem einzigen, hoch effizienten Lesebefehl gelesen werden kann. Diese Datenstrukturen werden als benutzerdefinierte Datenpakete bezeichnet.

## Datenpaket-Konfiguration

Jedes benutzerdefinierte Datenpaket kann bis zu zwanzig Gruppen von Datenpunkten enthalten. Jede Gruppe wird durch ihren Startindex und die Anzahl der Datenpunkte definiert. Die gesamte Größe des benutzerdefinierten Datenpakets darf 250 Byte nicht überschreiten. Die zum Definieren der benutzerdefinierten Datenpakete verwendeten Adressen werden nachfolgend aufgeführt.

**Benutzerdefiniertes Datenpaket 1 (Adresse 1)**
3001 Startindex von Gruppe 1
3002 Anzahl der Punkte in Gruppe 1
bis
3039 Startindex von Gruppe 20
3040 Anzahl der Punkte in Gruppe 20

**Benutzerdefiniertes Datenpaket 2 (Adresse 201)**
3041 Startindex von Gruppe 1
3042 Anzahl der Punkte in Gruppe 1
bis
3055 Startindex von Gruppe 20
3056 Anzahl der Punkte in Gruppe 20

**Benutzerdefiniertes Datenpaket 3 (Adresse 401)**
3057 Startindex von Gruppe 1
3058 Anzahl der Punkte in Gruppe 1
bis
3095 Startindex von Gruppe 20
3096 Anzahl der Punkte in Gruppe 20

**Hinweis:** Daten werden vom Gerät als 16-Bit-Register zurückgegeben. Digital-E/A muss in Blöcken von 16 Bit zugeordnet werden

## Adresssyntax für benutzerdefinierte Datenpakete

Tags können erstellt werden, um auf Daten an einem gegebenen Offset innerhalb eines benutzerdefinierten Datenpakets zuzugreifen. Die Adresssyntax lautet folgendermaßen: Die Standard-Datentypen werden **fett** dargestellt.

| Adresse | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| CPn_o | n = Nummer des Datenpakets (1–3)<br>o = Wort-Offset (0–125) | Word, **Short**, BCD, DWord, Long, LBCD, Float, String | Schreibgeschützt |
| CPn_o.b | n = Nummer des Datenpakets (1–3)<br>o = Wort-Offset (0–125)<br>b = Bit-Nummer (0/1–15/16)* | **Boolean** | Schreibgeschützt |

*Weitere Informationen finden Sie unter "Nullbasierte Bit-Adressierung" unter* ***Einstellungen****.*

**Hinweise:**

1. Nur 8 Zeichen lange ASCII-Zeichenfolgendaten werden unterstützt.
2. Wenn in der Gruppenkonfiguration eine Adresse für 16 Zeichen lange ASCII-Zeichenfolgendaten enthalten ist, können Daten als zwei 8 Zeichen lange ASCII-Zeichenfolgen-Datenelemente gelesen werden.

## Beispiel

Definieren Sie das benutzerdefinierte Datenpaket Nr. 1 so, dass Folgendes zugeordnet wird:

- 16 Bit für Digital-E/A (1001–1016).
- Fünfzehn 32-Bit-Ganzzahlen für Batch-Daten von Messlauf 1 (5101–5115).
- Zwölf 32-Bit-Gleitkommazahlen für analoge Ausgaben (7001–7012).
- Vier 8 Zeichen lange ASCII-Zeichenfolgen für den Messlauf (4101–4104).
- Sechs 8 Zeichen lange ASCII-Zeichenfolgen für die Messgerät-Station (4808–4813).
- Zwei 16 Zeichen lange ASCII-Zeichenfolgen für Flusskonfigurationsdaten (14001–14002).

**Hinweis:** Das ergibt insgesamt 222 Byte. Die Konfigurationsregister für das benutzerdefinierte Datenpaket haben dann folgende Werte:

3001 = 1001
3002 = 16
3003 = 5101
3004 = 15
3005 = 7001
3006 = 12
3007 = 4101
3008 = 4
3009 = 4808
3010 = 6
3011 = 14001
3012 = 2

Tags für den Zugriff auf die Digital-E/A-Daten haben folgende Adressen (wobei alle 16 Werte in Wort 0 enthalten sind):
CP1_0.0 (Wort 0 des benutzerdefinierten Datenpakets 1, Bit 0 – zugeordnet zu 1009)
CP1_0.1 (Wort 0 des benutzerdefinierten Datenpakets 1, Bit 1 – zugeordnet zu 1010)
...
CP1_0.6 (Wort 0 des benutzerdefinierten Datenpakets 1, Bit 6 – zugeordnet zu 1015)
CP1_0.7 (Wort 0 des benutzerdefinierten Datenpakets 1, Bit 7 – zugeordnet zu 1016)
CP1_0.8 (Wort 0 des benutzerdefinierten Datenpakets 1, Bit 8 – zugeordnet zu 1001)
CP1_0.9 (Wort 0 des benutzerdefinierten Datenpakets 1, Bit 9 – zugeordnet zu 1002)
...
CP1_0.14 (Wort 0 des benutzerdefinierten Datenpakets 1, Bit 14 – zugeordnet zu 1007)
CP1_0.15 (Wort 0 des benutzerdefinierten Datenpakets 1, Bit 15 – zugeordnet zu 1008)

Tags für den Zugriff auf die Batch-Daten von Messlauf 1 haben folgende Adressen (wobei jeder 32-Bit-Wert 2 Wörter verwendet):
CP1_1 (Wort 1 des benutzerdefinierten Datenpakets 1 – zugeordnet zu 5101)
CP1_3 (Wort 3 des benutzerdefinierten Datenpakets 1 – zugeordnet zu 5102)
...
CP1_29 (Wort 29 des benutzerdefinierten Datenpakets 1 – zugeordnet zu 5115)

Tags für den Zugriff auf die Daten der analogen Ausgaben haben folgende Adressen (wobei jeder 32-Bit-Wert 2 Wörter verwendet):
CP1_31 (Wort 31 des benutzerdefinierten Datenpakets 1 – zugeordnet zu 7001)
CP1_33 (Wort 33 des benutzerdefinierten Datenpakets 1 – zugeordnet zu 7002)
...
CP1_53 (Wort 53 des benutzerdefinierten Datenpakets 1 – zugeordnet zu 7012)

Tags für den Zugriff auf die 8 Zeichen langen ASCII-Zeichenfolgendaten für den Messlauf haben folgende Adressen (wobei jeder Zeichenfolgenwert 4 Wörter verwendet):
CP1_55 (Wort 55 des benutzerdefinierten Datenpakets 1 – zugeordnet zu 4101)
...
CP1_67 (Wort 67 des benutzerdefinierten Datenpakets 1 – zugeordnet zu 4104)

Tags für den Zugriff auf die 8 Zeichen langen ASCII-Zeichenfolgendaten für die Messgerät-Station haben folgende Adressen (wobei jeder Zeichenfolgenwert 4 Wörter verwendet):
CP1_71 (Wort 71 des benutzerdefinierten Datenpakets 1 – zugeordnet zu 4808)
...
CP1_91 (Wort 91 des benutzerdefinierten Datenpakets 1 – zugeordnet zu 4813)

Tags für den Zugriff auf die 16 Zeichen langen ASCII-Zeichenfolgendaten für Flusskonfigurationsdaten haben folgende Adressen (wobei jeder Zeichenfolgenwert 4 Wörter verwendet):
CP1_95 (Wort 95 des benutzerdefinierten Datenpakets 1 – zugeordnet zu 14001, Zeichen 1–8)
CP1_99 (Wort 99 des benutzerdefinierten Datenpakets 1 – zugeordnet zu 14001, Zeichen 9–16)
CP1_103 (Wort 103 des benutzerdefinierten Datenpakets 1 – zugeordnet zu 14002, Zeichen 1–8)
CP1_107 (Wort 107 des benutzerdefinierten Datenpakets 1 – zugeordnet zu 14002, Zeichen 9–16)

## Omni-Rohdatenarchiv

Der Omni-Mengenumwerter kann so konfiguriert werden, dass er verschiedene Speicherbereiche einer einzelnen Datenstruktur zuordnet und diese Struktur bei Auslösung in einem Archiv speichert. Benutzer können bis zu zehn Archive konfigurieren. Es gibt zwei zusätzliche Archive mit festem Format für Alarm- und Prüfdaten. Jedes Archiv ist ein kreisförmiger Puffer, wobei jeder neue Datensatz den ältesten Datensatz ersetzt.

### Konfiguration und Abruf von Datensätzen

Benutzer können die Datensatzstruktur der Rohdatenarchive 1 bis 10 konfigurieren. Die Archive 11 und 12 haben ein festes Format und enthalten Alarm- bzw. Prüfdaten.
*Umfassende Informationen zu Rohdatenarchiven finden Sie im englischen "Omni Technical Bulletin" Nr. 96073.*

Jeder Datensatz kann bis zu sechzehn Gruppen von Datenpunkten enthalten. Jede Gruppe wird durch ihren Startindex und die Anzahl der Datenpunkte definiert. Die zum Definieren der Archivdatensätze verwendeten Adressen werden nachfolgend aufgeführt. Die gesamte Größe des Datensatzes darf 250 Byte nicht überschreiten. Das Gerät verwendet die ersten 6 Byte für Datums- und Zeitstempeldaten, sodass 244 Byte für Rohdaten bleiben. Jeder Datensatz hat seinen eigenen booleschen Trigger. Daten werden gespeichert, wenn der Trigger von "Low" zu "High" übergeht.

Bevor der Startindex einer Gruppe, die Anzahl der Punkte in einer Gruppe oder der Trigger für ein Rohdatenarchiv geändert werden kann, muss die Archivierung angehalten werden. Das Flag **Allow Archive Configuration** muss im Gerät festgelegt werden. Beachten Sie, dass dadurch wahrscheinlich das Datenarchiv im Gerät, einschließlich aller Rohdatenarchive und des Textarchivs, neu initialisiert wird.

13920 Archivlauf – 0 = Anhalten, 1 = Starten
13921 Archive neu konfigurieren – 0 = Keine Konfigurationsänderungen zulässig, 1 = Konfigurationsänderungen zulässig

Dieser Treiber kann verwendet werden, um ein Rohdatenarchiv datensatzweise zu lesen. Um einen Datensatz zu lesen, schreiben Sie zuerst den gewünschten Datensatzindex in das Register "Angeforderter Datensatz". Sobald dieser Wert festgelegt ist, können Benutzer den Datensatz mit einem RA-Tag lesen. Benutzer müssen darauf achten, dass der angegebene Datensatzindex nicht die für das betreffende Archiv zulässige maximale Anzahl von Datensätzen überschreitet. Wenn der Wert für "Letzter aktualisierter Datensatz" Null ist, wurden im Archiv seit der letzten Initialisierung keine Datensätze gespeichert.

**Rohdatenarchiv 1 (Adresse 701)**
13500 Startindex von Gruppe 1
13501 Anzahl der Punkte in Gruppe 1
bis
13530 Startindex von Gruppe 16
13531 Anzahl der Punkte in Gruppe 16

13900 Boolescher Trigger

3701 Maximale Anzahl von Datensätzen
3702 Letzter aktualisierter Datensatz
3703 Angeforderter Datensatz

**Rohdatenarchiv 2 (Adresse 702)**
13540 Startindex von Gruppe 1
13541 Anzahl der Punkte in Gruppe 1
bis
13570 Startindex von Gruppe 16
13571 Anzahl der Punkte in Gruppe 16

13901 Boolescher Trigger

3704 Maximale Anzahl von Datensätzen
3705 Letzter aktualisierter Datensatz
3706 Angeforderter Datensatz

**Rohdatenarchiv 3 (Adresse 703)**
13580 Startindex von Gruppe 1
13581 Anzahl der Punkte in Gruppe 1
bis
13610 Startindex von Gruppe 16
13611 Anzahl der Punkte in Gruppe 16

13902 Boolescher Trigger

3707 Maximale Anzahl von Datensätzen
3708 Letzter aktualisierter Datensatz
3709 Angeforderter Datensatz

**Rohdatenarchiv 4 (Adresse 704)**
13620 Startindex von Gruppe 1
13621 Anzahl der Punkte in Gruppe 1
bis
13650 Startindex von Gruppe 16
13651 Anzahl der Punkte in Gruppe 16

13903 Boolescher Trigger

3710 Maximale Anzahl von Datensätzen
3711 Letzter aktualisierter Datensatz
3712 Angeforderter Datensatz

**Rohdatenarchiv 5 (Adresse 705)**
13660 Startindex von Gruppe 1
13661 Anzahl der Punkte in Gruppe 1
bis
13690 Startindex von Gruppe 16
13691 Anzahl der Punkte in Gruppe 16

13904 Boolescher Trigger

3713 Maximale Anzahl von Datensätzen
3714 Letzter aktualisierter Datensatz
3715 Angeforderter Datensatz

**Rohdatenarchiv 6 (Adresse 706)**
13700 Startindex von Gruppe 1
13701 Anzahl der Punkte in Gruppe 1
bis
13730 Startindex von Gruppe 16
13731 Anzahl der Punkte in Gruppe 16

13905 Boolescher Trigger

3716 Maximale Anzahl von Datensätzen
3717 Letzter aktualisierter Datensatz
3718 Angeforderter Datensatz

**Rohdatenarchiv 7 (Adresse 707)**
13740 Startindex von Gruppe 1
13741 Anzahl der Punkte in Gruppe 1
bis
13770 Startindex von Gruppe 16
13771 Anzahl der Punkte in Gruppe 16

13906 Boolescher Trigger

3719 Maximale Anzahl von Datensätzen
3720 Letzter aktualisierter Datensatz
3721 Angeforderter Datensatz

**Rohdatenarchiv 8 (Adresse 708)**
13780 Startindex von Gruppe 1
13781 Anzahl der Punkte in Gruppe 1
bis
13810 Startindex von Gruppe 16
13811 Startindex von Gruppe 16

13907 Boolescher Trigger

3722 Maximale Anzahl von Datensätzen
3723 Letzter aktualisierter Datensatz
3724 Angeforderter Datensatz

**Rohdatenarchiv 9 (Adresse 709)**
13820 Startindex von Gruppe 1
13821 Anzahl der Punkte in Gruppe 1
bis
13850 Startindex von Gruppe 16
13851 Anzahl der Punkte in Gruppe 16

13908 Boolescher Trigger

3725 Maximale Anzahl von Datensätzen
3726 Letzter aktualisierter Datensatz
3727 Angeforderter Datensatz

**Rohdatenarchiv 10 (Adresse 710)**
13860 Startindex von Gruppe 1
13861 Anzahl der Punkte in Gruppe 1
bis
13890 Startindex von Gruppe 16
13891 Anzahl der Punkte in Gruppe 16

13909 Boolescher Trigger

3728 Maximale Anzahl von Datensätzen
3729 Letzter aktualisierter Datensatz
3730 Angeforderter Datensatz

**Rohdatenarchiv 11: Alarm (Adresse 711)**
Nicht konfigurierbar

3731 Maximale Anzahl von Datensätzen
3732 Letzter aktualisierter Datensatz
3733 Angeforderter Datensatz

**Rohdatenarchiv 12: Archiv (Adresse 712)**
Nicht konfigurierbar

3734 Maximale Anzahl von Datensätzen
3735 Letzter aktualisierter Datensatz
3736 Angeforderter Datensatz

**Hinweis:** Daten werden vom Gerät als 16-Bit-Register zurückgegeben. Digital-E/A muss in Blöcken von 16 Bit zugeordnet werden

## Adresssyntax für Rohdatenarchive

Tags können erstellt werden, um auf Daten an einem gegebenen Offset innerhalb eines Rohdatenarchiv-Datensatzes zuzugreifen. Die Adresssyntax lautet folgendermaßen: Die Standard-Datentypen werden **fett** dargestellt.

| Adresse | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| RAn_o | n = Archivnummer (1–12)<br>o = Wort-Offset (0–125) | Word, **Short**, BCD, DWord, Long, LBCD, Float, String | Schreibgeschützt |
| RAn_o.b | n = Archivnummer (1–12)<br>o = Wort-Offset (0–125)<br>b = Bit-Nummer (0/1–15/16)* | **Boolean** | Schreibgeschützt |

*Weitere Informationen finden Sie unter "Nullbasierte Bit-Adressierung" unter* ***Einstellungen****.*

**Hinweise**:

1. Nur 8 Zeichen lange ASCII-Zeichenfolgendaten werden unterstützt.
2. Wenn in der Gruppenkonfiguration eine Adresse für 16 Zeichen lange ASCII-Zeichenfolgendaten enthalten ist, können Daten als zwei 8 Zeichen lange ASCII-Zeichenfolgen-Datenelemente gelesen werden.

## Zeitstempelformat

Die ersten 6 Byte jedes Datensatzes enthalten die Datums- und Uhrzeitangabe, zu der der Datensatz im Archiv platziert wurde.

| Byte | Beschreibung |
|---|---|
| 1 | Monat (1–12) *<br>Tag (1–31) |
| 2 | Tag (1–31)*<br>Monat (1–12) |
| 3 | Jahr (0–99) |
| 4 | Stunde (0–23) |
| 5 | Minute (0–59) |
| 6 | Sekunden (0–59) |

*Das Datumsformat wird mit Register 3842 festgelegt (0 = dd/mm/yy, 1 = mm/dd/yy).

## Datensatzstruktur des Alarm-/Ereignisprotokolls (Adresse 711)

| Feld | Datentyp | Beschreibung |
|---|---|---|
| 1 | 3-Byte-Datum | dd/mm/yy oder mm/dd/yy. |
| 2 | 3-Byte-Zeit | hh/mm/ss. |
| 3 | 16-Bit-Ganzzahl | Modbus-Indexnummer des Alarms oder Ereignisses. |
| 4 | 1 Byte | Alarmtyp. |
| 5 | 1 Byte | 0 = OK, 1 = Alarm. |
| 6 | IEEE-Gleitkommazahl | Wert der Messwertgeber-Variablen zum Zeitpunkt des Alarms bzw. Ereignisses. |
| 7 | 32-Bit-Ganzzahl | Volumenzähler zum Zeitpunkt des Alarms bzw. Ereignisses. |

| | | |
|---|---|---|
| 8 | 32-Bit-Ganzzahl | Massezähler zum Zeitpunkt des Alarms bzw. Ereignisses. |

### Alarmtypen

| Typ | Beschreibung |
|---|---|
| 0 | Ereignis protokollieren, Piepser auslösen und auf dem LCD jede Flankenänderung im durch Feld Nr. 3 identifizierten Bit anzeigen. |
| 1 | Ereignis protokollieren, Piepser auslösen und auf dem LCD jede Änderung der steigenden Flanke im durch Feld Nr. 3 identifizierten Bit anzeigen. |
| 2 | Jede Flankenänderung im durch Feld Nr. 3 identifizierten Bit im Ereignisprotokoll protokollieren. Keine Aktion von Piepser oder LCD-Anzeige. |
| 3 | Jede Änderung der steigenden Flanke im durch Feld Nr. 3 identifizierten Bit im Ereignisprotokoll protokollieren. Keine Aktion von Piepser oder LCD-Anzeige. |

### Datensatzstruktur des Prüfereignis-Protokolls (Adresse 712)

| Feld | Datentyp | Beschreibung |
|---|---|---|
| 1 | 3-Byte-Datum | dd/mm/yy oder mm/dd/yy. |
| 2 | 3-Byte-Zeit | hh/mm/ss. |
| 3 | 16-Bit-Ganzzahl | Ereignisnummer, wird bei jedem Ereignis inkrementiert, Rollover bei 65535. |
| 4 | 16-Bit-Ganzzahl | Modbus-Index der geänderten Variablen. |
| 5 | IEEE-Gleitkommazahl | Wert der numerischen Variablen vor der Änderung – alter Wert. |
| 6 | IEEE-Gleitkommazahl | Wert der numerischen Variablen nach der Änderung – neuer Wert. |
| 7 | 16 Zeichen lange ASCII-Zeichenfolge | Wert der Zeichenfolgenvariablen vor der Änderung – alter Wert. |
| 8 | 16 Zeichen lange ASCII-Zeichenfolge | Wert der Zeichenfolgenvariablen nach der Änderung – neuer Wert. |
| 9 | 32-Bit-Ganzzahl | Volumenzähler zum Zeitpunkt der Änderung. |
| 10 | 32-Bit-Ganzzahl | Massezähler zum Zeitpunkt der Änderung. |

**Hinweis:** Die Felder 5 und 6 werden auf 0.0 gesetzt, wenn die geänderte Variable eine Zeichenfolge ist. Die Felder 7 und 8 enthalten Nullzeichen, wenn die geänderte Variable keine Zeichenfolge ist. Wenn die Felder 7 und 8 acht Zeichen lange Zeichenfolgen enthalten, werden die restlichen acht Zeichen mit Nullen aufgefüllt.

## Omni-Textberichte

Der Omni-Mengenumwerter kann mehrere verschiedene Typen von Textberichten generieren. Jeder dieser Berichte kann von diesen Treiber gelesen und als Zeichenfolgenwert an den OPC-Client gesendet werden.

### Textberichtstypen

Es gibt eine Reihe von Berichtstypen, die aus dem Omni-Mengenumwerter abgerufen werden können. Sie können mit einem TR-Tag gelesen werden. Folgende Berichtstypen sind verfügbar:

**Benutzerdefinierte Berichtvorlagen**
9001 Berichtvorlage – Momentaufnahme/Intervall
9002 Berichtvorlage – Batch
9003 Berichtvorlage – Täglich
9004 Berichtvorlage – Prüfung

**Vorherige Batch-Berichte**
9101 Batch-Bericht – Letzter
9102 Batch-Bericht – Vorletzter
...
9108 Batch-Bericht – Achtletzter

**Vorherige Prüfberichte**
9201 Prüfbericht – Letzter
9202 Prüfbericht – Vorletzter
...
9208 Prüfbericht – Achtletzter

**Vorherige tägliche Berichte**
9301 Bericht vom Vortag – Letzter
9302 Bericht vom Vortag – Vorletzter
...
9308 Bericht vom Vortag – Achtletzter

**Letzter Momentaufnahme-Bericht**
9401 Letzter lokaler Momentaufnahme- bzw. Intervallbericht

**Puffer für verschiedene Berichte**
9402 Puffer für verschiedene Berichte

**Vorschau für monatliche Berichte**
9501 Bericht vom Vormonat – Letzter
9502 Bericht vom Vormonat – Vorletzter
...
9508 Bericht vom Vormonat – Achtletzter

## Adresssyntax für Textberichte

| Adresse | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| TRn<br>TRn T (ausgelöster Lese-vorgang) | n = Berichtsadresse (9001-9508) | String | Lesen/Schreiben |

## Beispiel

Um die Momentaufnahme-Berichtvorlage (Adresse 9001)zu lesen oder darin zu schreiben, erstellen Sie ein Tag mit der Adresse "TR9001".

**Hinweis:** Da das Lesen eines Textberichts mehrere Sekunden dauern kann, sollten die TR-Tags im OPC-Client inaktiv gehalten werden. Alternativ können stattdessen ausgelöste Lesevorgänge verwendet werden. Solange der Treiber einen Textbericht liest oder schreibt, sind für keine anderen Tags auf dem Kanal Lese- oder Schreibvorgänge möglich.

## Ausgelöste Textbericht-Lesevorgänge

Wie zuvor beschrieben, wird empfohlen, das Textbericht-Tag inaktiv zu halten, obwohl dies nicht immer möglich ist. Eine Funktion für ausgelöste Lesevorgänge wurde als Alternative hinzugefügt, sodass das Textbericht-Tag aktiv bleiben kann. Außerdem wird mit einem Hilfs-Trigger-Tag gesteuert, wann die tatsächlichen Lesevorgänge auf dem Gerät stattfinden.

Ein ausgelöster Lesevorgang beginnt möglicherweise nicht sofort, je nachdem, auf welchen Zeitpunkt im Aktualisierungszyklus des Textbericht-Tags der Trigger eingestellt ist. Nach Abschluss des Leseversuchs löscht der Treiber den Trigger-Status. Das Textbericht-Tag zeigt den Wert und die Datenqualität an, die sich aus dem letzten ausgelösten Leseversuch ergeben haben.

## Adresssyntax für Textbericht-Lesevorgang-Trigger

| Adresse | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| TRIG_TRn | n = Berichtsadresse (9001-9508) | Boolean | Lesen/Schreiben |

## Beispiel

Um den Bericht zum letzten Batch (Adresse 9101) nach Trigger zu lesen, erstellen Sie zwei Tags. Das erste ist ein Textbericht-Tag mit der Adresse "TR9101 T", und das zweite ist ein Textbericht-Lesevorgang-Trigger-Tag mit der Adresse "TRIG_TR9101".

**Hinweis:** Die Adresse des Textbericht-Tags sieht aus wie eine normale Textbericht-Adresse, gefolgt von einem Leerzeichen und dem Buchstaben "T" für "Triggered Read" (ausgelöster Lesevorgang). Dieses "T" muss in der Adresse vorhanden sein, damit ausgelöste Lesevorgänge funktionieren.

Um einen Lesevorgang auszulösen, setzen Sie den Trigger-Tag-Wert auf "Wahr" (ungleich null). Nach Abschluss des Leseversuchs setzt der Treiber den Trigger-Wert auf "Falsch" (0). Wenn der Lesevorgang erfolgreich war, ist die Datenqualität des Textbericht-Tags gut. Wenn der Lesevorgang fehlgeschlagen ist, ist die Datenqualität des Textbericht-Tags schlecht, und der Wert ist der letzte erfolgreich gelesene Wert.

## Textberichtsdaten auf dem Datenträger speichern

Der Treiber kann Textberichtsdaten auf dem Datenträger speichern. Diese Funktion wird mithilfe von Textbericht-Pfad-Tags aktiviert. Diese Tags werden verwendet, um Dateipfad-Zeichenfolgen in den Speicher des Treibers zu schreiben. Jeder Berichtstyp hat seinen eigenen Pfadpuffer. Nach einem erfolgreichen Textbericht-Lesevorgang prüft der Treiber den zugeordneten Pfadpuffer. Wenn dort ein gültiger Pfad gespeichert ist, speichert der Treiber die Berichtsdaten als ASCII-Text in dieser Datei. Die Datei wird bei Bedarf erstellt. Die Datei wird bei nachfolgenden Textbericht-Lesevorgängen überschrieben.

Die Pfadpuffer werden beim Start des Servers auf leere Zeichenfolgen initialisiert. Der Treiber schreibt Textberichtsdaten erst in die Datei, wenn ein gültiger Pfad im zugeordneten Pfadpuffer gespeichert wird. Pfaddaten sind nicht persistent. Die Pfadzeichenfolgen müssen jedes Mal neu geschrieben werden, wenn der Server neu gestartet wird. Die Pfadwerte können jederzeit geändert werden, sodass Benutzer Daten bei jedem Lesevorgang in anderen Dateien speichern können, sofern gewünscht.

Pfadzeichenfolgen können bis zu 255 Zeichen lang sein.

## Adresssyntax für Pfade von Textberichten

| Adresse | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| PATH_TRn | n = Berichtsadresse (9001-9508) | String | Lesen/Schreiben |

### Beispiel

Um den letzten Batch-Bericht (Adresse 9101) zu lesen und das Ergebnis auf dem Datenträger zu speichern, erstellen Sie zwei Tags. Das erste ist ein Textbericht-Tag mit der Adresse "TR9101", und das zweite ist ein Pfad-Tag mit der Adresse "PATH_TR9101".

Um die Berichtsdaten in einer Datei mit dem Namen "LastBatch.txt" zu speichern (die im Ordner "C:\OmniData\BatchReports" erstellt werden soll), richten Sie den Client so ein, dass er als Erstes den Pfad "C:\OmniData\BatchReports\LastBatch.txt" in das Pfad-Tag schreibt. Sobald dies abgeschlossen ist, lesen Sie das Textbericht-Tag. Wenn der Pfad nicht vor dem ersten Textbericht-Lesevorgang festgelegt wird, kann der Treiber die Daten nicht auf dem Datenträger speichern.

**Hinweis:** Um diese Funktion zu deaktivieren, schreiben Sie eine leere Zeichenfolge in das Pfad-Tag.

## Omni-Textarchiv

Der Omni-Mengenumwerter kann auch Berichte in einem Archiv speichern. Dieser Treiber kann einen Bereich von Berichten aus dem Archiv lesen und sie als Zeichenfolgenwert an den OPC-Client senden.

### Textarchive lesen

Bevor das Textarchiv gelesen werden kann, müssen zwei Einstellungen im Gerät festgelegt werden: das Startdatum des Archivs und die Anzahl der abzurufenden Tage. Diese 32-Bit-Ganzzahlwerte befinden sich an Adresse 15128 bzw. 15127. Das Datumsformat kann mit dem Wert an Adresse 3842 angegeben werden (0 = dd/mm/yy, 1 = mm/dd/yy). Kurz nachdem die Anzahl von Tagen festgelegt wurde, beginnt das Gerät, die Daten vorzubereiten. Wenn die Daten zum Lesen bereit sind, wird der Wert für die Anzahl von Tagen negativ. Das Textarchiv kann jederzeit gelesen werden, nachdem die Anzahl von Tagen festgelegt wurde. Der Treiber wartet darauf, dass der Wert negativ wird.

### Adresssyntax für Textarchive

| Adresse | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| TA<br>TA T (ausgelöster Lesevorgang) | k.A. | String | Schreibgeschützt |

**Hinweis**: Da das Lesen eines Textarchivs mehrere Minuten dauern kann, sollte das Tag "TA" im OPC-Client inaktiv gehalten werden. Alternativ können stattdessen ausgelöste Lesevorgänge verwendet werden. Dieses Tag darf nur mit asynchronen Lesevorgängen gelesen werden, da das maximale Timeout für synchrone Lesevorgänge im Server nicht hoch genug gesetzt werden kann, um eine typische Textarchiv-Anforderung zu lesen. Solange das Textarchiv gelesen wird, sind für keine anderen Tags auf dem Kanal Lese- oder Schreibvorgänge möglich.

Wenn ein laufender Textarchiv-Lesevorgang fehlschlägt, sollten Benutzer den Lesepuffer des Geräts zurücksetzen, indem sie den Wert 999 in das Register für die Anzahl der Tage (15127) schreiben und anschließend das normale Textarchiv-Leseverfahren wiederholen. Andernfalls ruft der Treiber möglicherweise den ersten Teil des angeforderten Archivbereichs nicht ab.

### Ausgelöste Textarchiv-Lesevorgänge

Es wird empfohlen, das Textarchiv-Tag inaktiv zu halten, obwohl dies nicht immer möglich ist. Eine Funktion für ausgelöste Lesevorgänge wurde als Alternative hinzugefügt, sodass das Textarchiv-Tag aktiv bleiben kann. Außerdem wird mit einem Hilfs-Trigger-Tag gesteuert, wann die tatsächlichen Lesevorgänge auf dem

Gerät stattfinden. Der Trigger-Wert wird im Speicher des Treibers gespeichert und kann mithilfe eines Tags mit der im Folgenden beschriebenen Syntax gelesen und festgelegt werden.

Ein ausgelöster Lesevorgang beginnt möglicherweise nicht sofort, je nachdem, auf welchen Zeitpunkt im Aktualisierungszyklus des Textarchivs der Trigger eingestellt ist. Nach Abschluss des Leseversuchs löscht der Treiber den Trigger-Status. Das Textarchiv-Tag zeigt den Wert und die Datenqualität an, die sich aus dem letzten ausgelösten Leseversuch ergeben haben.

## Adresssyntax für Textarchiv-Lesevorgang-Trigger

| Adresse | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| TRIG_TA | k.A. | Boolean | Lesen/Schreiben |

## Beispiel

Um das Textarchiv nach Trigger zu lesen, erstellen Sie zwei Tags. Das erste ist ein Textarchiv-Tag mit der Adresse "TA T", und das zweite ist ein Textarchiv-Lesevorgang-Trigger-Tag mit der Adresse "TRIG_TA". Benutzer müssen Tags für das Startdatum und die Anzahl der Tage erstellen.

**Hinweis**: Die Adresse des Textarchiv-Tags sieht aus wie eine normale Textarchiv-Adresse, gefolgt von einem Leerzeichen und dem Buchstaben "T" für "Triggered Read" (ausgelöster Lesevorgang). Dieses "T" muss in der Adresse vorhanden sein, damit ausgelöste Lesevorgänge funktionieren.

Um einen Lesevorgang auszulösen, setzen Sie den Trigger-Tag-Wert auf "Wahr" (ungleich null). Nach Abschluss des Leseversuchs setzt der Treiber den Trigger-Wert auf "Falsch" (0). Wenn der Lesevorgang erfolgreich war, ist die Datenqualität des Textarchiv-Tags gut. Wenn der Lesevorgang fehlgeschlagen ist, ist die Datenqualität des Textarchiv-Tags schlecht, und der Wert ist der letzte erfolgreich gelesene Wert.

## Textarchivdaten auf dem Datenträger speichern

Der Treiber kann Textarchivdaten auf dem Datenträger speichern. Diese Funktion wird mit einem Textarchiv-Pfad-Tag aktiviert. Dieses Tag wird verwendet, um eine Dateipfad-Zeichenfolge in den Speicher des Treibers zu schreiben. Nach einem erfolgreichen Textarchiv-Lesevorgang prüft der Treiber den zugeordneten Pfadpuffer. Wenn dort ein gültiger Pfad gespeichert ist, speichert der Treiber die Textarchivdaten als ASCII-Text in dieser Datei. Die Datei wird bei Bedarf erstellt. Die Datei wird bei nachfolgenden Textarchiv-Lesevorgängen überschrieben.

Der Pfadpuffer wird beim Start des Servers auf eine leere Zeichenfolge initialisiert. Der Treiber schreibt Textarchivdaten erst in die Datei, wenn ein gültiger Pfad im zugeordneten Pfadpuffer gespeichert wird. Pfaddaten sind nicht persistent. Benutzer müssen die Pfadzeichenfolge jedes Mal neu schreiben, wenn der Server neu gestartet wird. Der Pfadwert kann jederzeit geändert werden, sodass die Daten bei jedem Lesevorgang in einer anderen Datei gespeichert werden können (sofern gewünscht).

Die Pfadzeichenfolge kann bis zu 255 Zeichen lang sein. Das Verzeichnis kann nicht das Stammverzeichnis (C:\\TextArchive.txt) sein innerhalb des Windows-Verzeichnis oder innerhalb des Programmverzeichnisses, abgesehen vom Installationsverzeichnis des Servers. Die Dateierweiterung muss entweder .txt oder .log sein.

## Adresssyntax für Pfade von Textarchiven

| Adresse | Bereich | Datentyp | Zugriff |
|---|---|---|---|
| PATH_TA | k.A. | String | Lesen/Schreiben |

### Beispiel

Um das Textarchiv zu lesen und das Ergebnis auf dem Datenträger zu speichern, erstellen Sie zwei Tags. Das erste ist ein Textarchiv-Tag mit der Adresse "TA", und das zweite ist ein Pfad-Tag mit der Adresse "PATH_TA". Benutzer müssen Tags für das Startdatum und die Anzahl der Tage erstellen wie oben beschrieben.

Um die Textarchivdaten in einer Datei mit dem Namen "TextArchive.txt" zu speichern (die im Ordner "C:\OmniData\ArchiveData" erstellt werden soll), richten Sie den Client so ein, dass er als Erstes den Pfad "C:\OmniData\ArchiveData\TextArchive.txt" in das Pfad-Tag schreibt. Sobald dies abgeschlossen ist, lesen Sie das Textarchiv-Tag. Wenn der Pfad nicht vor dem ersten Textarchiv-Lesevorgang festgelegt wird, kann der Treiber die Daten nicht auf dem Datenträger speichern.

**Hinweis**: Um diese Funktion zu deaktivieren, schreiben Sie eine leere Zeichenfolge in das Pfad-Tag.

## Beschreibung von Funktionscodes

### Modbus-Adressierungsmodell

| Dezimalwert | Hexadezimalwert | Beschreibung |
|---|---|---|
| 01 | 0x01 | Read Coil Status |
| 02 | 0x02 | Read Input Status |
| 03 | 0x03 | Read Holding Registers |
| 04 | 0x04 | Read Internal Registers |
| 05 | 0x05 | Force Single Coil |
| 06 | 0x06 | Preset Single Register |
| 15 | 0x0F | Force Multiple Coils |
| 16 | 0x10 | Preset Multiple Registers |
| 22 | 0x16 | Masked Write Register |

### Daniels-S500-Mengenumwerter-Adressierungsmodell

| Dezimalwert | Hexadezimalwert | Beschreibung |
|---|---|---|
| 01 | 0x01 | Read Coil Status |
| 02 | 0x02 | Read Input Coil |
| 03 | 0x03 | Read Holding Registers |
| 05 | 0x05 | Force Single Coil |
| 16 | 0x10 | Preset Multiple Registers |

## Konfigurations-API-Dienst – Kanaleigenschaften

Die folgenden Befehle definieren einen Kanal, welcher den Konfigurations-API-Dienst verwendet.

### Allgemeine Eigenschaften

`common.ALLTYPES_NAME` * Erforderlicher Parameter.

**Hinweis**: Wird diese Eigenschaft geändert, so ändert sich die API-Endpunkt-URL.

`common.ALLTYPES_DESCRIPTION`

`servermain.MULTIPLE_TYPES_DEVICE_DRIVER` * Erforderlicher Parameter

`servermain.CHANNEL_DIAGNOSTICS_CAPTURE`

### Ethernet-Kommunikationseigenschaften

`servermain.CHANNEL_ETHERNET_COMMUNICATIONS_NETWORK_ADAPTER_STRING`

### Erweiterte Eigenschaften

`servermain.CHANNEL_NON_NORMALIZED_FLOATING_POINT_HANDLING`

### Schreiboptimierungen

`servermain.CHANNEL_WRITE_OPTIMIZATIONS_METHOD`

`servermain.CHANNEL_WRITE_OPTIMIZATIONS_DUTY_CYCLE`

***Siehe auch***: *Abschnitt "Konfigurations-API-Dienst " im Hilfesystem für den Server.*

## Konfigurations-API-Dienst – Geräteeigenschaften

Die folgenden Befehle definieren ein Gerät, welches den Konfigurations-API-Dienst verwendet.

### Allgemeine Eigenschaften

`common.ALLTYPES_NAME` * Erforderlicher Parameter

`common.ALLTYPES_DESCRIPTION`

`servermain.DEVICE_CHANNEL_ASSIGNMENT`

`servermain.MULTIPLE_TYPES_DEVICE_DRIVER` * Erforderlicher Parameter

`servermain.DEVICE_MODEL` * Nicht erforderlich, aber prüfen Sie, ob der Standardwert akzeptabel ist

`servermain.DEVICE_ID_STRING` * Erforderlicher Parameter

`servermain.DEVICE_DATA_COLLECTION`

`servermain.DEVICE_SIMULATED`

## Scan-Modus

`servermain.DEVICE_SCAN_MODE`

`servermain.DEVICE_SCAN_MODE_RATE_MS`

`servermain.DEVICE_SCAN_MODE_RATE_MS`

`servermain.DEVICE_SCAN_MODE_PROVIDE_INITIAL_UPDATES_FROM_CACHE`

## Automatische Herabstufung

`servermain.DEVICE_AUTO_DEMOTION_ENABLE_ON_COMMUNICATIONS_FAILURES`

`servermain.DEVICE_AUTO_DEMOTION_DEMOTE_AFTER_SUCCESSIVE_TIMEOUTS`

`servermain.DEVICE_AUTO_DEMOTION_PERIOD_MS`

`servermain.DEVICE_AUTO_DEMOTION_DISCARD_WRITES`

## Tag-Generierung

`servermain.DEVICE_TAG_GENERATION_ON_STARTUP`

`servermain.DEVICE_TAG_GENERATION_DUPLICATE_HANDLING`

`servermain.DEVICE_TAG_GENERATION_GROUP`

`servermain.DEVICE_TAG_GENERATION_ALLOW_SUB_GROUPS`

**TPip**: Senden Sie einen PUT-Befehl mit leerem Textkörper an den TagGeneration-Dienstendpunkt auf dem Gerät, um die automatische Tag-Generierung aufzurufen.
***Siehe auch***: *Weitere Informationen finden Sie in der Hilfe zu den Diensten.*

## Zeitvorgabe

`servermain.DEVICE_CONNECTION_TIMEOUT_SECONDS`

`servermain.DEVICE_REQUEST_TIMEOUT_MILLISECONDS`

`servermain.DEVICE_RETRY_ATTEMPTS`

`servermain.DEVICE_INTER_REQUEST_DELAY_MILLISECONDS`

***Siehe auch***: *Abschnitt "Konfigurations-API-Dienst" im Hilfesystem für den Server.*

# Ereignisprotokollmeldungen

Die folgenden Informationen betreffen Meldungen, die im Fensterbereich Ereignisprotokoll in der Hauptbenutzeroberfläche angezeigt werden. Informationen zum Filtern und Sortieren der Detailansicht Ereignisprotokoll finden Sie in der OPC-Serverhilfe. In der Serverhilfe sind viele allgemeine Meldungen enthalten, die also auch gesucht werden sollten. Im Allgemeinen werden die Art der Meldung (Information, Warnung) sowie Fehlerbehebungsinformationen bereitgestellt (sofern möglich).

**Tipp**: Meldungen, die aus einer Datenquelle stammen (z.B. Drittanbieter-Software, einschließlich Datenbanken), werden über das Ereignisprotokoll dargestellt. Die Schritte der Problembehandlung sollten eine Recherche zu diesen Meldungen im Internet und in der Händlerdokumentation beinhalten.

## Ungültige Adresse im Blockbereich. | Adressbereich = <Start> bis <Ende>.

### Fehlertyp:

Fehler

### Mögliche Ursache:

1. Es wurde versucht, einen nicht vorhandenen Speicherort im angegebenen Gerät zu referenzieren.
2. Es wurde versucht, mehr Register zu lesen als im Protokoll zugelassen.

### Mögliche Lösung:

1. Überprüfen Sie die Tags, die Adressen im angegebenen Bereich auf dem Gerät zugewiesen sind, und entfernen Sie solche, die ungültige Speicherorte referenzieren.
2. Verringern Sie den Wert für die Registerblockgröße auf 125.

### Siehe auch:

1. Fehlerbehandlung
2. Blockgrößen

## Ungültiges Array. | Array-Bereich = <Start> bis <Ende>.

### Fehlertyp:

Fehler

### Mögliche Ursache:

Das definierte Array von Adressen reicht über das Ende des Adressraums hinaus.

### Mögliche Lösung:

1. Überprüfen Sie die Größe des Speichers auf dem Gerät, und definieren Sie die Array-Länge entsprechend um.
2. Überprüfen Sie die Tags, die Adressen im angegebenen Bereich auf dem Gerät zugewiesen sind,

und entfernen Sie solche, die ungültige Speicherorte referenzieren.

3. Verringern Sie den Wert für die Array-Größe auf 125.

### Siehe auch:

1. Fehlerbehandlung
2. Blockgrößen

## Blockanfrage mit Ausnahmecode beantwortet. | Adressenbereich = <Start> bis <Ende>, Ausnahmecode = <Code>.

### Fehlertyp:

Warnung

### Mögliche Ursache:

Eine Beschreibung des Ausnahmecodes finden Sie unter 'Modbus-Ausnahmecodes'.

### Mögliche Lösung:

Siehe 'Modbus-Ausnahmecodes'.

## In Adresse kann nicht geschrieben werden, Gerät hat mit Ausnahmecode geantwortet. | Adresse = '<Adresse>', Ausnahmecode = <Code>.

### Fehlertyp:

Warnung

### Mögliche Ursache:

Eine Beschreibung des Ausnahmecodes finden Sie unter 'Modbus-Ausnahmecodes'.

### Mögliche Lösung:

Siehe 'Modbus-Ausnahmecodes'.

## Von Adresse kann nicht gelesen werden, Gerät hat mit Ausnahmecode geantwortet. | Adresse = '<Adresse>', Ausnahmecode = <Code>.

### Fehlertyp:

Warnung

### Mögliche Ursache:

Eine Beschreibung des Ausnahmecodes finden Sie unter 'Modbus-Ausnahmecodes'.

### Mögliche Lösung:

Siehe 'Modbus-Ausnahmecodes'.

## Tag-Import wegen zu wenig Speicherressourcen fehlgeschlagen.

### Fehlertyp:

Warnung

### Mögliche Ursache:
Der Treiber kann den für die Verarbeitung der Variablenimportdatei erforderlichen Speicherplatz nicht zuweisen.

### Mögliche Lösung:
Schließen Sie alle nicht benötigten Anwendungen, und versuchen Sie es erneut.

## Beim Tag-Import ist eine Dateiausnahme aufgetreten.

### Fehlertyp:
Warnung

### Mögliche Ursache:
Die Variablenimportdatei konnte nicht gelesen werden.

### Mögliche Lösung:
Regenerieren Sie die Variablenimportdatei.

## Fehler beim Parsen von Datensatz in Importdatei. | Datensatznummer = <Nummer>, Feld = <Name>.

### Fehlertyp:
Warnung

### Mögliche Ursache:
Das angegebene Feld in der Variablenimportdatei konnte nicht geparst werden, da es ungültig oder länger als erwartet ist.

### Mögliche Lösung:
Ändern Sie in der Variablenimportdatei das Feld, das den Fehler verursacht.

## Beschreibung für Datensatz in Importdatei abgeschnitten. | Datensatznummer = <Nummer>.

### Fehlertyp:
Warnung

### Mögliche Ursache:
Die Tag-Beschreibung im angegebenen Datensatz ist zu lang.

### Mögliche Lösung:
Der Treiber schneidet Beschreibungen nach Bedarf ab. Um diesen Fehler zu verhindern, kürzen Sie die Beschreibung in der Variablenimportdatei.

## Importierter Tag-Name ist ungültig und wurde geändert. | Tag-Name = '<Tag>', geänderter Tag-Name = '<Tag>'.

### Fehlertyp:
Warnung

### Mögliche Ursache:
Der in der Variablenimportdatei gefundene Tag-Name enthält ungültige Zeichen.

### Mögliche Lösung:
Der Treiber erstellt gültige Namen basierend auf der Variablenimportdatei. Um diesen Fehler zu vermeiden und die Konsistenz der Namen zu wahren, ändern Sie den Namen der exportierten Variablen.

## Tag konnte nicht importiert werden, da der Datentyp nicht unterstützt wird. | Tag-Name = '<Tag>', nicht unterstützter Datentyp = '<Typ>'.

### Fehlertyp:
Warnung

### Mögliche Ursache:
Der Treiber unterstützt nicht den in der Variablenimportdatei angegebenen Datentyp.

### Mögliche Lösung:
Ändern Sie den in der Variablenimportdatei angegebenen Datentyp in einen unterstützten Typ. Wenn die Variable für eine Struktur verwendet wird, bearbeiten Sie die Datei manuell, um jedes für die Struktur erforderliche Tag zu definieren, oder konfigurieren Sie die erforderlichen Tags im Server manuell.

### Siehe auch:
Variablen werden aus Concept exportiert

## Omni-Textpuffer konnte aufgrund eines Fehlers bei der Speicherzuordnung nicht gelesen werden.

### Fehlertyp:
Warnung

### Mögliche Ursache:
Der Treiber kann den für den Lesevorgang eines Omni-Textdatensatzes oder -Textarchivs erforderlichen Speicherplatz nicht zuweisen.

### Mögliche Lösung:
Schließen Sie alle nicht benötigten Anwendungen, und versuchen Sie es erneut.

## Keine Omni-Textarchivdaten im angegebenen Datumsbereich verfügbar.

### Fehlertyp:
Warnung

### Mögliche Ursache:
Es sind keine Daten im Textarchiv für den Datumsbereich vorhanden, der im Register für Startdatum (15128) und Register für Anzahl an Tagen (15127) angegeben wurde.

### Mögliche Lösung:
Dies ist nicht unbedingt ein Fehler. Vergewissern Sie sich, dass keine Daten für den angegebenen Datumsbereich verfügbar sind.

## Texteingabe im Omni-Textbericht abgeschnitten. | Berichtsnummer = <Nummer>.

### Fehlertyp:
Warnung

### Mögliche Ursache:
Es wurde versucht, mehr als 8192 Byte in einen Textbericht zu schreiben. Dieser Grenzwert ist vom Protokoll festgelegt.

### Mögliche Lösung:
Schreiben Sie keine Zeichenfolgen mit mehr als 8192 Byte. Bei längeren Zeichenfolgen werden nur die ersten 8192 Zeichen in das Gerät geschrieben.

## Omni-Textbericht konnte aufgrund von Grenzwert für Paketanzahl nicht gelesen werden. | Berichtsnummer = <Nummer>.

### Fehlertyp:
Warnung

### Mögliche Ursache:
Die Länge von Textberichten beträgt höchstens 8192 Byte. Dieser Grenzwert ist vom Protokoll festgelegt. Der Treiber liest 8192 Byte, bevor das entsprechende Zeichen für das Dateiende festgestellt wird.

### Mögliche Lösung:
Vergewissern Sie sich, dass die vom Gerät verwendete Berichtvorlage einen Bericht aus höchstens 8192 Byte generiert.

## Schreibvorgang fehlgeschlagen. Maximale Pfadlänge überschritten. | Tag-Adresse = '<Adresse>', Maximale Länge = <Anzahl>.

### Fehlertyp:
Warnung

### Mögliche Ursache:
Pfadlänge ist auf die angegebene Anzahl von Zeichen begrenzt.

### Mögliche Lösung:
Verwenden Sie einen kürzeren Pfad.

## Fehler beim Schreiben der Omni-Textdaten in Datei. | Tag-Adresse = '<Adresse>', Grund = '<Grund>'.

### Fehlertyp:
Warnung

### Mögliche Ursache:
Der Treiber konnte die Omni-Textdaten aus dem angegebenen Grund nicht auf den Datenträger schreiben.

### Mögliche Lösung:

Weitere Informationen zur Fehlerbehebung für den angegebenen Grund erhalten Sie in der Dokumentation zum Betriebssystem.

## Omni-Textausgabedatei konnte nicht geöffnet werden. | Tag-Adresse = '<Adresse>', Grund = '<Grund>'.

### Fehlertyp:

Warnung

### Mögliche Ursache:

Die Datei, die in einem Omni-Textpfad-Tag angegeben ist, konnte nicht erstellt oder geöffnet werden.

### Mögliche Lösung:

Weitere Informationen zur Fehlerbehebung für den angegebenen Grund erhalten Sie in der Dokumentation zum Betriebssystem. Wahrscheinlich liegt der Grund an einem ungültigen Pfad.

### Siehe auch:

1. Omni-Textberichte
2. Omni-Textarchiv

## In Adresse kann nicht geschrieben werden. Unerwartete Zeichen in Antwort. | Tag-Adresse = '<Adresse>'.

### Fehlertyp:

Warnung

## Von Adresse kann nicht gelesen werden. Unerwartete Zeichen in Antwort. | Tag-Adresse = '<Adresse>'.

### Fehlertyp:

Warnung

## Blockadresse konnte nicht gelesen werden. Unerwartete Zeichen in Antwort. | Adressbereich = <Start> bis <Ende>.

### Fehlertyp:

Warnung

## Die Omni-Textausgabedatei konnte nicht geändert werden. | Tag-Adresse = '<address>', Grund = Der angegeben Pfad ist nicht zulässig.

### Fehlertyp:

Warnung

### Mögliche Ursache:

Das in einem Omni-Textpfad angegebene Verzeichnis ist nicht zulässig.

### Mögliche Lösung:

Lesen Sie in der Hilfe für den Treiber nach und geben Sie einen Pfad zu einem sicheren Ort an.

### Siehe auch:

1. Omni-Textberichte
2. Omni-Textarchiv

## Die Omni-Textausgabedatei konnte nicht geändert werden. | Tag-Adresse = '<address>', Grund = Die angegebene Dateierweiterung muss '.txt' oder '.log' sein.

### Fehlertyp:

Warnung

### Mögliche Ursache:

Die im Omni-Textpfad-Tag angegeben Erweiterung ist nicht zulässig. Sie muss .txt oder .log sein.

### Mögliche Lösung:

Gültige Erweiterung angeben.

### Siehe auch:

1. Omni-Textberichte
2. Omni-Textarchiv

## Tag-Datenbank wird von Datei importiert. | Dateiname = '<Name>'.

### Fehlertyp:

Informationen

## Fehlermaskendefinitionen

**B** = Hardwareunterbrechung festgestellt
**F** = Framing-Fehler
**E** = E/A-Fehler
**O** = Zeichenpufferüberlauf
**R** = RX-Pufferüberlauf
**P** = Erhaltener Byte-Paritätsfehler
**T =** TX-Puffer voll

## Modbus-Ausnahmecodes

Folgende Daten stammen aus der englischen Dokumentation "Modbus Application Protocol Specifications" (Spezifikationen für das Modbus-Anwendungsprotokoll).

| Code dezimal/hexadezimal | Name | Bedeutung |
|---|---|---|
| 01/0x01 | ILLEGAL FUNCTION | Der in der Abfrage erhaltene Funktionscode ist keine zulässige Aktion für den Server. Das kann daran liegen, dass der Funktionscode nur auf neuere Geräte anwendbar ist und in der ausgewählten Einheit nicht implementiert wurde. Es könnte auch anzeigen, dass der Server sich im falschen Status befindet, um eine Anfrage dieses Typs zu verarbeiten, z.B. weil er nicht konfiguriert ist und aufgefordert wird, Registerwerte zurückzugeben. |
| 02/0x02 | ILLEGAL DATA ADDRESS | Die in der Abfrage erhaltene Datenadresse ist keine zulässige Adresse für den Server. Insbesondere ist die Kombination aus Referenznummer und Übertragungslänge ungültig. Für einen Controller mit 100 Registern ist eine Anfrage mit Offset 96 und Länge 4 erfolgreich. Eine Anfrage mit Offset 96 und Länge 5 generiert Ausnahme 02. |
| 03/0x03 | ILLEGAL DATA VALUE | Ein im Abfrage-Datenfeld enthaltener Wert ist kein zulässiger Wert für den Server. Dies deutet darauf hin, dass ein Fehler in der Struktur des Rests einer komplexen Anfrage vorliegt, z.B. eine falsche implizierte Länge. Es bedeutet insbesondere nicht, dass ein Datenelement, das zur Speicherung in einem Register eingereicht wurde, einen Wert außerhalb der Erwartung des Anwendungsprogrammes hat, da das Modbus-Protokoll die Bedeutung bestimmter Werte für bestimmte Register nicht kennt. |
| 04/0x04 | SERVER DEVICE FAILURE | Ein nicht wiederherstellbarer Fehler ist aufgetreten, während der Server versucht hat, die angeforderte Aktion auszuführen. |
| 05/0x05 | ACKNOWLEDGE | Der Server hat die Anfrage akzeptiert und verarbeitet sie, aber dies wird viel Zeit in Anspruch nehmen. Diese Antwort wird zurückgegeben, um einen Timeout-Fehler im Client zu verhindern. Der Client kann als Nächstes eine Meldung ausgeben, dass das Abrufprogramm abgeschlossen ist, um zu ermitteln, ob Verarbeitung abgeschlossen ist. |
| 06/0x06 | SERVER DEVICE BUSY | Der Server ist mit der Verarbeitung eines lang dauernden Programmbefehls beschäftigt. Der Client muss die Nachricht später erneut senden, wenn der Server frei ist. |
| 07/0x07 | NEGATIVE ACKNOWLEDGE | Der Server kann die in der Abfrage erhaltene Programmfunktion nicht ausführen. Dieser Code wird für eine erfolglose Programmieranfrage mit Funktionscode 13 oder 14 (dezimal) zurückgegeben. Der |

| Code dezimal/hexadezimal | Name | Bedeutung |
|---|---|---|
| | | Client muss Diagnose- oder Fehlerinformationen vom Server anfordern. |
| 08/0x08 | MEMORY PARITY ERROR | Der Server hat versucht, Erweiterungsspeicher zu lesen, aber dabei einen Paritätsfehler im Arbeitsspeicher gefunden. Der Client kann die Anfrage erneut versuchen, aber möglicherweise muss das Servergerät gewartet werden. |
| 10/0x0A | GATEWAY PATH UNAVAILABLE | Die spezielle Verwendung in Verbindung mit Gateways deutet darauf hin, dass das Gateway keinen internen Kommunikationspfad vom Eingangskanal zum Ausgangskanal zuordnen konnte, um die Anfrage zu verarbeiten. Das bedeutet normalerweise, dass das Gateway falsch konfiguriert oder überlastet ist. |
| 11/0x0B | GATEWAY TARGET DEVICE FAILED TO RESPOND | Die spezielle Verwendung in Verbindung mit Gateways deutet darauf hin, dass keine Antwort vom Zielgerät empfangen wurde. Das bedeutet normalerweise, dass das Gerät im Netzwerk nicht vorhanden ist. |

**Hinweis**: Für diesen Treiber werden die Begriffe "Server" und "unangefordert" synonym verwendet.

# Index

## A

Abfall 11
Abrufverzögerung 11
Adressbeschreibungen 33
Adressierung dynamischer Flüssigkeitsmessgeräte 39
Alle Werte für alle Tags schreiben 9
Allgemein 13
Anfangsaktualisierungen aus Cache 16
Anforderungs-Timeout 17
Anfragen verwerfen, wenn herabgestuft 19
Anstieg 11
Ausgangs-Coils 22
Automatische Herabstufung 18
Automatische Tag-Datenbankgenerierung 27
Automatisches Wählen 12

## B

Baudrate 5, 11
BCD 31
Bei doppeltem Tag 20
Bei Eigenschaftsänderung 19
Bei Gerätestart 19
Beim Tag-Import ist eine Dateiausnahme aufgetreten. 63
Benutzerdefinierte Omni-Datenpakete 45
Beschreibung für Datensatz in Importdatei abgeschnitten. | Datensatznummer = <Nummer>. 63
Beschreibung von Funktionscodes 58
Beschreibungen einschließen 24
Betriebsmodus 14
Betriebsverhalten 11
Blockadresse konnte nicht gelesen werden. Unerwartete Zeichen in Antwort. | Adressbereich = <Start> bis <Ende>. 66
Blockanfrage mit Ausnahmecode beantwortet. | Adressenbereich = <Start> bis <Ende>, Ausnahmecode = <Code>. 62
Blockgrößen 22
Boolean 31

## C

Coils 33
COM-ID 11
COM-Port 10

## D

Daniel-S500-Mengenumwerter 5
Daniels-S500-Mengenumwerter-Adressierung 38
Daten-Bits 5, 11
Datensammlung 14
Datentypbeschreibung 31
Diagnose 7
Die Omni-Textausgabedatei konnte nicht geändert werden. | Tag-Adresse = '<address>', Grund = Der angegeben Pfad ist nicht zulässig. 66
Die Omni-Textausgabedatei konnte nicht geändert werden. | Tag-Adresse = '<address>', Grund = Die angegebene Dateierweiterung muss '.txt' oder '.log' sein. 67
Double 31
DTR 11
Durch Null ersetzen 13
Durch Tag angegebenes Scan-Intervall berücksichtigen 16
DWord 31
Dynamisches Flüssigkeitsmessgerät (DFM) SFC3 6

## E

E/A-Fehler 67
Eingangs-Coils 22
Einstellungen auf Kanalebene 7
Elliott-Mengenumwerter 5
Elliott-Mengenumwerter-Adressierung 37
Elternteilgruppe 20
Ereignisprotokollmeldungen 61
Erstellen 21
Ethernet-Kapselung 16

## F

Fehler beim Parsen von Datensatz in Importdatei. | Datensatznummer = <Nummer>, Feld = <Name>. 63
Fehler beim Schreiben der Omni-Textdaten in Datei. | Tag-Adresse = '<Adresse>', Grund = '<Grund>'. 65
Fehlerbehandlung 25
Fehlermaskendefinitionen 67
Float 31
Flusssteuerung 11
Framing 24, 67

## G

Generieren 20
Geräteeigenschaften – Allgemein 13
Geräteeigenschaften – Automatische Herabstufung 18
Geräteeigenschaften – Ethernet-Kapselung 16
Geräteeigenschaften – Redundanz 26
Geräteeigenschaften – Tag-Generierung 19
Geteilt 10
Globale Einstellungen 8

## H

Halteregister 22
Hardwareunterbrechung 67
Herabstufen bei Fehler 18
Herabstufungszeitraum 19

## I

ID 14
Identifikation 6, 13
Importierter Tag-Name ist ungültig und wurde geändert. | Tag-Name = '<Tag>', geänderter Tag-Name = '<Tag>'. 63
In Adresse kann nicht geschrieben werden, Gerät hat mit Ausnahmecode geantwortet. | Adresse = '<Adresse>', Ausnahmecode = <Code>. 62
In Adresse kann nicht geschrieben werden. Unerwartete Zeichen in Antwort. | Tag-Adresse = '<Adresse>'. 66
Inaktive Verbindung schließen 12

Inaktivitätsdauer bis Schließen 12
Interne Register 22
IP-Adresse 16

## K

Kanaleigenschaften – Allgemein 6
Kanaleigenschaften – Erweitert 12
Kanaleigenschaften – Kommunikationsserialisierung 7
Kanaleigenschaften – Schreiboptimierungen 9
Kanaleigenschaften – Serielle Kommunikation 10
Kanalzuweisung 14
Keine 10
Keine Omni-Textarchivdaten im angegebenen Datumsbereich verfügbar. 64
Kommunikations-Timeouts 17-18
Kommunikationsfehler melden 11-12
Kommunikationsprotokoll 5

## L

Lastausgleich 8
LBCD 31
Leseverarbeitung 12
Long 31
Löschen 20

## M

Magnetek-GPD-515-Laufwerk 5
Magnetek-GPD-515-Laufwerksadressierung 37
Modbus-Adressierung 33
Modbus-Ausnahmecodes 68
Modbus-kompatible Geräte 5
Modell 14
Modem 10, 12
Modemeinstellungen 12

## N

Nachfolgend 25
Name 13
Netzwerk 1 – Netzwerk 500 8
Netzwerksmodus 8
Nicht geändert 13
Nicht normalisierte Float-Handhabung 13
Nicht scannen, nur Abruf anfordern 16
Nur den letzten Wert für alle Tags schreiben 9
Nur den letzten Wert für nicht boolesche Tags schreiben 9

## O

Omni-Mengenumwerter 5
Omni-Mengenumwerter-Adressierung 41
Omni-Rohdatenarchiv 48
Omni-Textarchiv 56
Omni-Textausgabedatei konnte nicht geöffnet werden. | Tag-Adresse = '<Adresse>', Grund = '<Grund>'. 66
Omni-Textbericht konnte aufgrund von Grenzwert für Paketanzahl nicht gelesen werden. | Berichts-nummer = <Nummer>. 65
Omni-Textberichte 53
Omni-Textpuffer konnte aufgrund eines Fehlers bei der Speicherzuordnung nicht gelesen werden. 64
Operation ohne Kommunikation 12
Optimierungsmethode 9

## P

Parität 5, 11, 67
Physisches Medium 10
Port 16
Priorität 8
Protokoll 17

## R

Redundanz 26

Register 33
RS-485 11
RTS 11
RX-Pufferüberlauf 67

## S

Scan-Modus 15
Schreibvorgang fehlgeschlagen. Maximale Pfadlänge überschritten. | Tag-Adresse = '<Adresse>', Maximale Länge = <Anzahl>. 65
Serielle Kommunikation 10
Serielle Port-Einstellungen 10
Servicezyklus 9
Setup 5
Short 31
Simuliert 15
Statistikelemente 28
Stopp-Bits 5, 11
String 31

## T

Tag-Datenbank wird von Datei importiert. | Dateiname = '<Name>'. 67
Tag-Generierung 19
Tag-Import wegen zu wenig Speicherressourcen fehlgeschlagen. 62
Tag-Zähler 7, 15
Tag konnte nicht importiert werden, da der Datentyp nicht unterstützt wird. | Tag-Name = '<Tag>', nicht unterstützter Datentyp = '<Typ>'. 64
Tags bei unzulässiger Adresse deaktivieren 26
Texteingabe im Omni-Textbericht abgeschnitten. | Berichtsnummer = <Nummer>. 65
Timeout bis zum Herabstufen 18
Transaktionen pro Zyklus 8
Treiber 14
TSXCUSBMBP-USB-Adapter 6
TX-Puffer voll 67

## U

Überlauf 67

Überschreiben 20
Übersicht 5
Ungültige Adresse im Blockbereich. | Adressbereich = <Start> bis <Ende>. 61
Ungültiges Array. | Array-Bereich = <Start> bis <Ende>. 61
Untergruppen zulassen 21
Unterstützte Geräte 5

## V

Variablenimportdatei 24
Variablenimporteinstellungen 24
Verbindungs-Timeout 12
Verbindungstyp 10
Versuche vor Timeout 18
Verzögerung zwischen Anfragen 18
Verzögerung zwischen Geräten 13
Virtuelles Netzwerk 7
Von Adresse kann nicht gelesen werden, Gerät hat mit Ausnahmecode geantwortet. | Adresse = '<Adresse>', Ausnahmecode = <Code>. 62
Von Adresse kann nicht gelesen werden. Unerwartete Zeichen in Antwort. | Tag-Adresse = '<Adresse>'. 66
Vorangestellt 25

## W

Wiederholte Meldungen zurückweisen 26
Word 31

## Z

Zeichenfolgen als Block lesen 23