Modbus TCP/IP-Ethernet-Treiber

© 2025 PTC Inc. Alle Rechte vorbehalten.

Inhaltsverzeichnis

Modbus TCP/IP-Ethernet-Treiber 1
Inhaltsverzeichnis 2
Modbus TCP/IP-Ethernet-Treiber 5
Übersicht 5
Unterstützte Gerätemodelle 6
Setup 7
Kanaleigenschaften – Allgemein 9
Tag-Zähler 10
Kanaleigenschaften – Ethernet-Kommunikation 11
Kanaleigenschaften – Schreiboptimierungen 11
Kanaleigenschaften – Erweitert 12
Kanaleigenschaften – Kommunikationsserialisierung 13
Kanaleigenschaften – Ethernet 14
Geräteeigenschaften – Allgemein 16
Geräteeigenschaften – Scan-Modus 17
Geräteeigenschaften – Zeitvorgabe 18
Geräteeigenschaften – Automatische Herabstufung 19
Geräteeigenschaften – Tag-Generierung 20
Geräteeigenschaften – Variablenimporteinstellungen 22
Geräteeigenschaften – Unangefordert 23
Überlegungen zu Modbus-Clients und Modbus-Servern 25
Geräteeigenschaften – Fehlerbehandlung 25
Geräteeigenschaften – Ethernet 26
Geräteeigenschaften – Blockgrößen 27
Geräteeigenschaften – Redundanz 28
Konfigurations-API Modbus Ethernet-Beispiel 29
Aufzählungen 30
Gerätemodellaufzählungen 31
Automatische Tag-Datenbankgenerierung 32
Aus benutzerdefinierten Anwendungen importieren 32
Kommunikation optimieren 34
Datentypbeschreibung 36
Adressbeschreibungen 38
Treibersystem-Tag-Adressierung 38
Beschreibung von Funktionscodes 39
Untergeordnetes Applicom-Modell und Adressierung 39

Generische Modbus-Adressierung 40
TSX Quantum 43
TSX Premium 47
CEG-Adressierung 50
Fluenta-Adressierung 50
Instromet-Adressierung 50
Postfach-Adressierung 50
Modbus-Adressierung 51
Roxar-Adressierung 55
Statistikelemente 55
Ereignisprotokollmeldungen 58
Fehler beim Starten der Winsock-Kommunikation. 58
Unangeforderte Kommunikation konnte nicht gestartet werden. 58
Unangeforderter Postfach-Zugriff für nicht definiertes Gerät. Socket wird geschlossen. | IP-Adresse = '<Adresse>'. 58
Unangeforderte, nicht unterstützte Anfrage erhalten. | IP-Adresse = '<Adresse>'. 59
Fehler bei unangeforderter Postfach-Speicherzuordnung. | IP-Adresse = '<Adresse>'. 59
Socket-Verbindung konnte nicht erstellt werden. 59
Fehler beim Öffnen der Datei für Tag-Datenbank-Import. | Betriebssystemfehler = '<Fehler>'. 60
Ungültiges Array. | Array-Bereich = <Start> bis <Ende>. 60
Ungültige Adresse im Block. | Blockbereich = <Adresse> bis <Adresse>. 60
Auflösen des Hosts fehlgeschlagen. | Host-Name = '<Name>'. 60
Angegebene Blockgröße für Ausgangs-Coils überschreitet maximale Blockgröße. | Angegebene Blockgröße = <Anzahl> (Coils), Maximale Blockgröße = <Anzahl> (Coils). 61
Angegebene Blockgröße für Eingangs-Coils überschreitet maximale Blockgröße. | Angegebene Blockgröße = <Anzahl> (Coils), Maximale Blockgröße = <Anzahl> (Coils). 61
Angegebene Blockgröße für interne Register überschreitet maximale Blockgröße. | Angegebene Blockgröße = <Anzahl> (Register), Maximale Blockgröße = <Anzahl> (Register). 61
Angegebene Blockgröße für Halteregister überschreitet maximale Blockgröße. | Angegebene Blockgröße = <Anzahl> (Register), Maximale Blockgröße = <Anzahl> (Register). 61
Blockanfrage hat mit Ausnahme geantwortet. | Blockbereich = <Adresse> bis <Adresse>, Ausnahme = <Code>. 61
Blockanfrage hat mit Ausnahme geantwortet. | Blockbereich = <Adresse> bis <Adresse>, Funktionscode = <Code>, Ausnahme = <Code>. 62
Ungültige Blocklänge erhalten. | Blockbereich = <Start> bis <Ende>. 62
Tag-Import wegen zu wenig Speicherressourcen fehlgeschlagen. 62
Beim Tag-Import ist eine Dateiausnahme aufgetreten. 62
Fehler beim Parsen von Datensatz in Importdatei. | Datensatznummer = <Nummer>, Feld = <Feld>. 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>'. 63
In Adresse kann nicht geschrieben werden, Gerät hat mit Ausnahme geantwortet. | Adresse = '<Adresse>', Ausnahme = <Code>. 64
Ethernet-Manager wurde gestartet. 64
Ethernet-Manager wurde angehalten. 64
Tag-Datenbank wird importiert. | Quelldatei = '<Dateiname>'. 64
Eine Client-Anwendung hat die CEG-Erweiterung über das System-Tag _CEGExtension geändert. | Erweiterung = '<Erweiterung>'. 64
Unangeforderte Kommunikation wird gestartet. | Protokoll = '<Name>', Port = <Nummer>. 65
Speicher für Modbus-Server-Gerät erstellt. | Modbus-Server-Geräte-ID = <Gerät>. 65
Alle Kanäle abonnieren ein virtuelles Netzwerk oder alle Geräte hören auf Remote-Adressen, wodurch unangeforderte Kommunikation beendet wird. 65
Kanal ist in einem virtuellen Netzwerk, alle Geräte verwenden jetzt wieder nur einen Socket pro Gerät. 65
Geräte-ID kann nicht von 'Modbus-Client-Modus' in 'Server-Modus' geändert werden, wenn ein Client verbunden ist. 65
Geräte-ID kann nicht von 'Modbus-Server-Modus' in 'Client-Modus' geändert werden, wenn ein Client verbunden ist. 65
Modbus-Server-Modus ist nicht zulässig, wenn sich der Kanal in einem virtuellen Netzwerk befindet. Die Geräte-ID darf keinen Loopback oder keine lokale IP-Adresse enthalten. 65
Postfach-Modell nicht zulässig, wenn sich der Kanal in einem virtuellen Netzwerk befindet. 65
Modbus-Ausnahmecodes 67
Modbus Ethernet-Kanaleigenschaften 69
Modbus Ethernet-Geräteeigenschaften 69
Modbus Ethernet-Tag-Eigenschaften 70
Index 71

Modbus TCP/IP-Ethernet-Treiber

Hilfeversion 1.158

INHALT

Übersicht
Was ist Modbus TCP/IP-Ethernet-Treiber?

Setup
Wie konfiguriere ich einen Kanal sowie ein Gerät für die Verwendung mit diesem Treiber?

Konfiguration via API
Wie konfiguriere ich einen Kanal und ein Gerät mithilfe der Konfigurations-API?

Automatische Tag-Datenbankgenerierung
Wie kann ich Tags für Modbus TCP/IP-Ethernet-Treiber konfigurieren?

Kommunikation optimieren
Wie erziele ich die beste Leistung mit Modbus TCP/IP-Ethernet-Treiber?

Datentypbeschreibung
Welche Datentypen unterstützt Modbus TCP/IP-Ethernet-Treiber?

Adressbeschreibungen
Wie referenziere ich eine Datenposition auf einem Ethernet-Modbus-Gerät?

Ereignisprotokollmeldungen
Welche Meldungen können bei Modbus TCP/IP-Ethernet-Treiber auftreten?

Übersicht

Modbus TCP/IP-Ethernet-Treiber bietet eine zuverlässige Möglichkeit, Modbus-Ethernet-Geräte mit Client-Anwendungen, u.a. HMI, SCADA, Historian, MES, ERP und zahlreichen benutzerdefinierten Anwendungen, zu verbinden. Benutzer müssen TCP/IP ordnungsgemäß installieren, um diesen Treiber zu verwenden. *Weitere Informationen zum Setup finden Sie in der Windows-Dokumentation.*

Hinweis: Der Treiber gibt Meldungen aus, wenn während der Operation ein Fehler auftritt.

Unterstützte Gerätemodelle

Applicom

Dieses Modell unterstützt die Applicom-Adressierungssyntax für generische Modbus-Geräte sowie TSX Premium und TSX Quantum.

Ethernet-zu-Modbus-Plus-Bridge

Der Treiber kann auch über eine Ethernet-zu-Modbus-Plus-Bridge mit Modbus-Plus-Geräten kommunizieren. Verwenden Sie als Geräte-ID die IP-Adresse der Bridge zusammen mit dem Modbus-Plus-Bridge-Index. Bridge-IP-Adresse 205.167.7.12 und Bridge-Index 5 ergeben beispielsweise die Geräte-ID 205.167.7.12.5. Informationen zu Erwerb und Setup einer MBE-zu-MBP-Bridge erhalten Sie beim Händler für Automatisierungsprodukte von Modicon/Schneider.

CEG

Dieses Modell unterstützt die erweiterte Blockgröße von CEG-Geräten.

Fluenta

Dieses Modell unterstützt die nicht standardmäßige entsprechende Modbus-Zuordnung des Mengenumwerters Fluenta FGM 100/130.

Instromet

Dieses Modell unterstützt die nicht standardmäßige entsprechende Modbus-Zuordnung von Instromet-Geräten.

Postfach

Dieses Modell betrifft die Art, wie unaufgeforderte Anfragen behandelt werden. Indem ein Postfach-Gerät definiert wird, verhält sich der Treiber im Netzwerk nicht wie eine SPS. Stattdessen fungiert er als Speicherbereich für jedes definierte Postfach-Gerät. Wenn der Treiber einen unangeforderten Befehl empfängt, erkennt er die IP-Adresse, von der die Nachricht stammt, und platziert die Daten in dem Speicherbereich, der dem betreffenden Gerät zugeordnet ist. Wenn die Nachricht von einem Gerät mit einer IP-Adresse stammt, die nicht als Postfach-Gerät definiert wurde, wird die Nachricht nicht verarbeitet. Jede Client-Anwendung, die auf einem Gerät dieses Typs liest oder schreibt, liest bzw. schreibt in dem im Treiber enthaltenen Speicherbereich und nicht auf dem physischen Gerät.

Informationen zum Senden von unaufgeforderten Anfragen an Modbus TCP/IP-Ethernet-Treiber finden Sie in der englischen Modicon-Dokumentation zur MSTR-Anweisung.

Hinweis: Modbus-Postfach unterstützt nicht Funktionscode 22 (0x16). Nur 0x10 (Holding Reg Write Multiple) und 0x6 (Holding Reg Write Single) werden unterstützt. Benutzer können in ein einzelnes Bit schreiben, indem Sie die Option Halteregister-Bit-Schreibvorgänge in den Geräteeigenschaften deaktivieren. Dadurch wird die Verwendung der Sequenz Lesen/Bearbeiten/Schreiben erzwungen, anstatt direkt in das Bit zu schreiben. Damit dies funktioniert, muss nur die Einstellung des Client-Modbus-Geräts (nicht des Postfachs) geändert werden.

Postfach-Client-Berechtigungen für das Postfach-Gerätemodell

Modbus-Client

Die meisten Projekte werden konfiguriert, um als ein Modbus-Client zu funktionieren. In diesem Modus greift der Treiber auf ein physisches Gerät zu (z.B. das TSX Quantum oder jedes andere mit Modbus Open Ethernet kompatible Gerät).

Modbus – Unangefordert oder Server-Modus

Modbus TCP/IP-Ethernet-Treiber fungiert als Gerät im Netzwerk, wenn Modbus das ausgewählte Modell ist und mit einer Geräte-ID konfiguriert wurde, die der IP-Adresse des Hostrechners entspricht. Der Treiber akzeptiert alle empfangenen unaufgeforderten Befehle und versucht, sie zu verarbeiten, als sei er eine weitere SPS. Jeder Modbus-Client im Netzwerk kann mithilfe dieser IP-Adresse mit diesem simulierten Gerät kommunizieren.

Die Geräte-ID für ein Modbus-Servergerät wird folgendermaßen angegeben: *YYY.YYY.YYY.YYY.XXX*. *YYY* kann entweder für die Loopback-Adresse oder für die lokale IP-Adresse des PCs stehen, der den Treiber ausführt. *XXX* weist die Stations-ID für den Modbus-Server aus und kann im Bereich 0 bis 255 liegen.

Mehrere Modbus-Servergeräte können dieselbe Stations-ID haben. In diesem Szenario verweisen alle Geräte, die die Stations-ID gemeinsam nutzen, auf ein gemeinsames simuliertes Gerät. Wenn der Remote-Modbus-Client Daten von einem Modbus-Servergerät (Stations-ID) anfordert, das nicht vorhanden ist, enthält die Antwort Daten von der Station 0. Sobald ein Modbus-Servergerät im Projekt erstellt wird, wird dieser Modbus-Server aktiviert und bleibt aktiviert, bis der Server heruntergefahren wird. Durch Ändern der Stations-ID wird ein neues Modbus-Servergerät aktiviert, das aktiviert bleibt, bis der Server heruntergefahren wird.

Die Adressen 1 bis 65536 sind für Ausgangs-Coils, Eingangs-Coils, interne Register und Halteregister implementiert. Im unaufgeforderten Modus antwortet der Treiber auf jede gültige Anfrage zum Lesen oder Schreiben dieser Werte von externen Geräten (Funktionscodes [dezimal] 01, 02, 03, 04, 05, 06, 15 und 16). Außerdem wurde in diesem Treiber Loopback (auch als Funktionscode 08, Untercode 00 bezeichnet) implementiert. Auf diese Positionen kann lokal durch den Host-PC zugegriffen werden, und zwar als dem Modbus-Servergerät zugewiesene Tags.
Hinweis: Lesegeschützter Zugriff ist für unaufgeforderte Geräte nicht zulässig.

Roxar

Dieses Modell unterstützt die nicht standardmäßige entsprechende Modbus-Zuordnung des Roxar-RFM-Verwässerungsmessgeräts.
***Siehe auch**: Gerätemodellaufzählungen und Geräteeigenschaften.*

Setup

Kanal- und Gerätegrenzwerte

Die von diesem Treiber unterstützte maximale Anzahl von Kanälen liegt bei 1024. Die maximale Anzahl von Geräten, die von diesem Treiber unterstützt werden, liegt bei 8192 pro Kanal.

Tipp: Einstellungen auf Kanalebene gelten für alle Geräte, die auf diesem Kanal konfiguriert wurden.

Hinweis: Modbus TCP/IP-Ethernet-Treiber erfordert Winsock V1.1 oder höher.

Kommunikationsserialisierung

Modbus TCP/IP-Ethernet-Treiber unterstützt Kommunikationsserialisierung, die angibt, ob Datenübertragungen auf jeweils nur einen Kanal begrenzt werden sollten.
*Weitere Informationen finden Sie unter **Kommunikationsserialisierung**.*

Hinweise:

- Wenn die Kanalserialisierung aktiviert ist, sind die unaufgeforderte Kommunikation und die Eigenschaft **Max. Sockets pro Gerät** deaktiviert. Das Postfach-Modell ist für die Kanalserialisierung nicht

verfügbar.

- Nicht alle Eigenschaften sind für alle Modelle verfügbar bzw. auf diese anwendbar.

***Siehe auch**: Configuring a device with Configuration API commands* (Konfiguration eines Geräts mit Konfigurations-API-Befehlen), *API Example with Modbus* (API-Beispiel mit Modbus)

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 Mehr-

aufwandsverarbeitung 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.

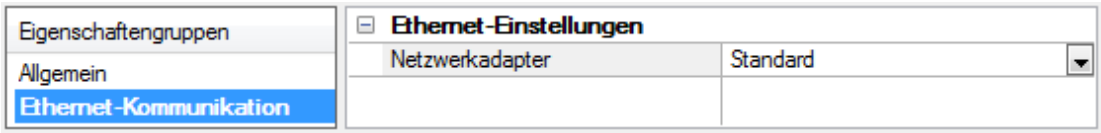
Weitere Informationen dazu finden Sie in der Serverhilfe unter "Kommunikationsdiagnosen".

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 – Ethernet-Kommunikation

Ethernet-Kommunikation kann für die Kommunikation mit Geräten verwendet werden.

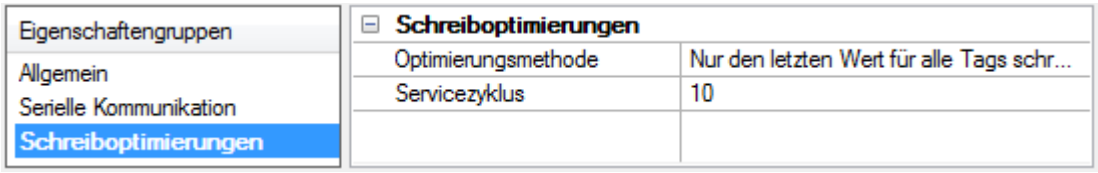


Ethernet-Einstellungen

Netzwerkadapter: Geben Sie den zu bindenden Netzwerkadapter an. Wird dieses Feld leer gelassen oder es wird "Standard" ausgewählt, so wählt das Betriebssystem den Standardadapter aus.

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 opti-

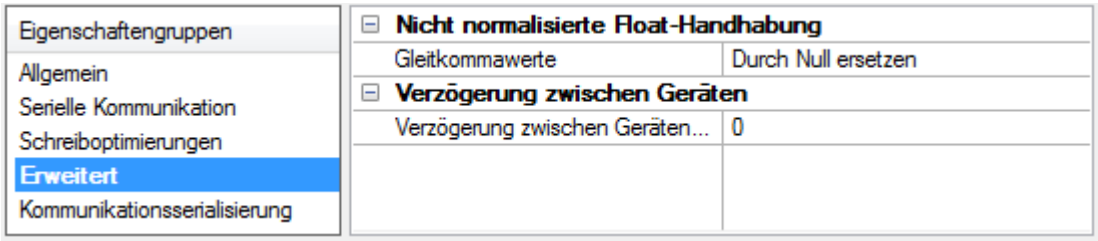
mieren. Dadurch können Benutzer den HMI-Datenvorgang optimieren, ohne Probleme mit Booleschen Operationen (z.B. eine vorübergehende Schaltfläche) zu verursachen.

- **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 – 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.

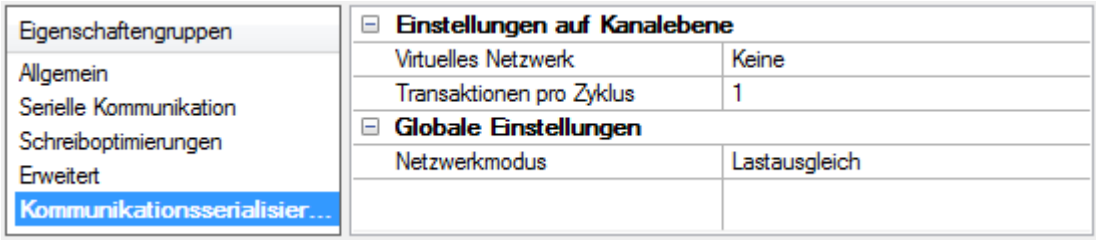
Verzögerung zwischen Geräten: Geben Sie die Zeitdauer an, in der der Kommunikationskanal das Senden einer Anforderung an das nächste Gerät verzögert, nachdem Daten vom aktuellen Gerät in demselben Kanal empfangen wurden. Null (0) deaktiviert die Verzögerung.

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

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:

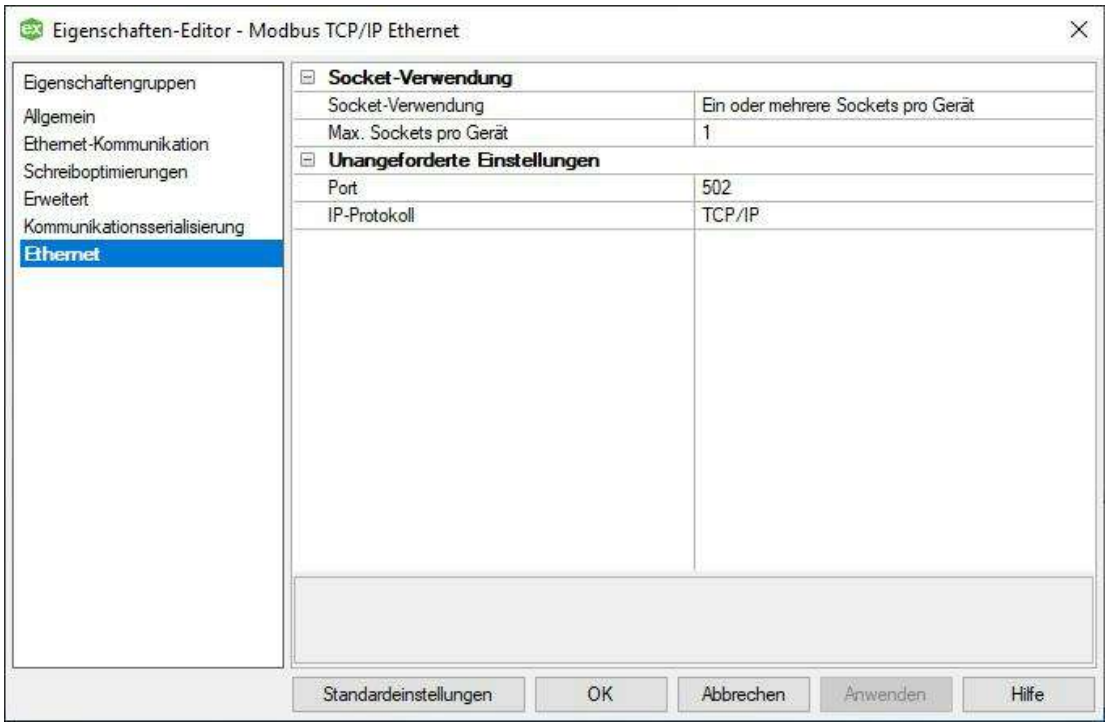
1. Kanäle mit ausstehenden Schreibvorgängen haben den höchsten Vorrang.
2. Kanäle mit ausstehenden expliziten Lesevorgängen (durch interne Plugins oder externe Client-Schnittstellen) werden je nach Priorität des Lesevorgangs priorisiert.
3. Gescannte Lesevorgänge und andere periodische Ereignisse (treiberspezifisch).

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 – Ethernet



Socket-Verwendung

Socket-Verwendung: Geben Sie an, ob der Treiber einen einzelnen Socket für alle Geräte auf diesem Kanal verwenden soll oder mehrere Sockets für die Kommunikation mit Geräten.

- **Ein oder mehrere Sockets pro Gerät**: Gibt an, dass der Treiber für jedes Gerät im Netzwerk mindestens einen Socket verwendet und diesen Socket als aktive Verbindungen aufrechterhält. Dies ist die Standardeinstellung und das Standardverhalten.
- **Ein Socket pro Kanal**: Gibt an, dass der Treiber nur einen einzelnen Socket verwendet, um die Kommunikation für jedes Gerät im Kanal bereitzustellen.

Max. Sockets pro Gerät: Gibt die maximale Anzahl von für das Gerät verfügbaren Sockets an. Die Standardeinstellung ist 1.

Weitere Informationen dazu, wie sich diese Einstellungen auf die Leistung auswirken, finden Sie unter ***Kommunikation optimieren****.*

Unangeforderte Einstellungen

Unangeforderte Einstellungen gelten sowohl für den Modbus-Client- als auch für den Modbus-Server-Modus.

Port: Gibt die Port-Nummer an, die der Treiber für die Überwachung unangeforderter Anfragen verwendet. Der gültige Bereich liegt zwischen 0 und 65535. Der Standardwert ist 502.

IP-Protokoll: Gibt das Protokoll an, das der Treiber für die Überwachung unangeforderter Anfragen verwendet. Die Optionen sind User Datagram Protocol (UDP) oder Transmission Control Protocol (TCP/IP). Die Standardeinstellung ist TCP/IP.

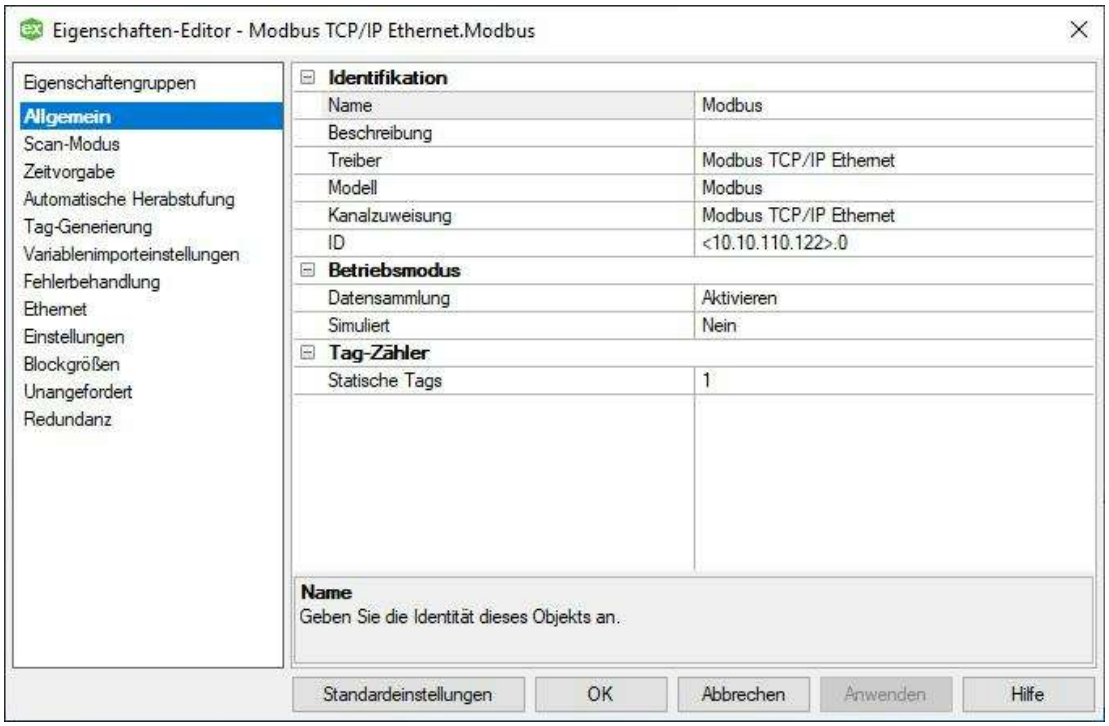
Hinweise:

1. Der Modbus-Client-Modus wird aktiviert, wenn eine externe IP als Geräte-ID verwendet wird. Wenn sich der Treiber im Client-Modus befindet, kann er unangeforderte Anfragen annehmen. Nachdem der Treiber vom OPC-Server geladen wurde, startet der Treiber einen Listening-Thread für nicht angeforderte Daten. Dieser Thread ist global für alle Kanäle, die im OPC-Server konfiguriert sind. Wenn beispielsweise in einem OPC-Serverprojekt drei Kanäle definiert sind und die Einstellung in einem der Kanäle geändert wird, wird dieselbe Änderung an den anderen beiden Kanälen vorgenommen. Der Listening-Thread wird neu gestartet, sobald die Änderung angewendet wurde. Das Ereignisprotokoll veröffentlicht ein Ereignis für den Neustart.

2. Der Modbus-Server-Modus ist aktiviert, wenn das Modell auf Modbus und die Geräte-ID auf IP_Address.yyy festgelegt ist, wobei IP_Address die lokale IP-Adresse des im Kanal zugewiesenen Netzwerkadapters oder die lokale Loopback-IP-Adresse des PCs sein kann, auf dem der Treiber ausgeführt wird. z.B. 127.0.0.1.1 or 127.xxx.xxx.xxx, wobei xxx = 0–255 ist, und yyy (Stations-ID) = 0–255. Wenn sich der Treiber im Modbus-Server-Modus befindet, startet der Treiber einen Listening-Thread für unangeforderte Daten. Dieser Thread überwacht nur einen Port und ist global für alle im OPC-Server konfigurierten Kanäle. Mehrere Geräte in einem Serverprojekt können so eingerichtet werden, dass sie als Modbus-Server fungieren. Während jedoch auf alle Modbus-Servergeräte im Projekt einzeln über eindeutige Geräte-IDs zugegriffen werden kann, überwachen alle Modbus-Server im selben Projekt denselben Port oder werden über diesen erreicht.

Weitere Informationen zum unangeforderten Modus finden Sie unter ***Modbus – Unangefordert oder Server-Modus****.*

Geräteeigenschaften – Allgemein



Identifikation

Name: Benutzerdefinierte ID dieses Geräts.

Beschreibung: Benutzerdefinierte Informationen über dieses Gerät.

Kanalzuweisung: Benutzerdefinierter Name des Kanals, zu dem dieses Gerät derzeit gehört.

Treiber: Ausgewählter Protokolltreiber für dieses Gerät.
Weitere Informationen zu einem bestimmten Gerätemodell finden Sie unter ***Unterstützte Gerätemodelle***.

Modell: Die jeweilige Version des Geräts.

ID: Geben Sie die Geräte-IP-Adresse zusammen mit einem Modbus-Bridge-Index im Ethernet-Netzwerk an. Geräte-IDs werden als <*HOST*>.*XXX* angegeben, wobei *HOST* ein UNC/DNS-Standardname oder eine IP-Adresse ist. *XXX* bezeichnet die Modbus-Server-ID des Geräts mit einem Wert im Bereich zwischen 0 und 255. Wenn keine Bridge verwendet wird, sollte der Index auf 0 gesetzt werden. Je nach Modell und Geräte-ID kann ein Gerät so konfiguriert werden, dass es als Client oder Server fungiert.
Weitere Informationen zum unangeforderten Modus finden Sie unter ***Modbus – Unangefordert oder Server-Modus***

Beispiele

1. Beim Anfordern von Daten von einem Modicon-TSX-Quantum-Gerät mit der IP-Adresse 205.167.7.19 muss die Geräte-ID als 205.167.7.19.0 eingegeben werden.

2. Beim Anfordern von Daten von einem Modbus-Plus-Gerät mit der IP-Adresse 205.167.7.50, das mit Bridge-Index 5 einer Modbus-Ethernet-Bridge verbunden ist, muss die Geräte-ID als 205.167.7.50.5 eingegeben werden.

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: Diese Option versetzt das Gerät in den Simulationsmodus. 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 Speicherzuordnung des Elements basiert auf der Gruppenaktualisierungsrate. Die Daten werden nicht gespeichert, wenn der Server das Element entfernt (z.B., wenn der Server neu initialisiert wird). Die Standardeinstellung ist "Nein".

Hinweise:

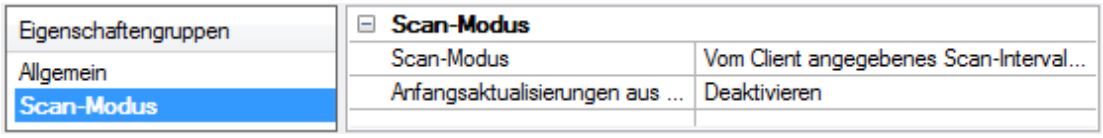
1. Dieses System-Tag (_Simulated) ist schreibgeschützt und kann für den Laufzeitschutz nicht geschrieben werden. Das System-Tag ermöglicht es, dass diese Eigenschaft vom Client überwacht wird.
2. Im Simulationsmodus basiert die Speicherzuordnung des Elements auf Client-Aktualisierungsraten (Gruppenaktualisierungsrate 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.

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

Siehe auch: ***Configuring a device with Configuration API commands*** (Konfiguration eines Geräts mit Konfigurations-API-Befehlen), ***API Example with Modbus*** (API-Beispiel mit Modbus)

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.



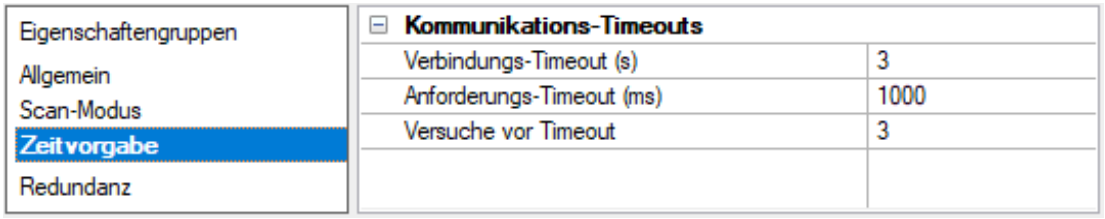
Scan-Modus: Geben Sie an, wie Tags im Gerät für an abonnierende Clients gesendete Aktualisierungen gescannt werden. Es folgen Beschreibungen der Optionen:

- **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 – 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

Verbindungs-Timeout: Mit dieser Eigenschaft (die in erster Linie von Ethernet-basierten Treibern verwendet wird) wird die Zeitdauer gesteuert, die zum Herstellen einer Socket-Verbindung mit einem Remote-Gerät erforderlich ist. Die Verbindungszeit des Gerät ist häufig länger als normale

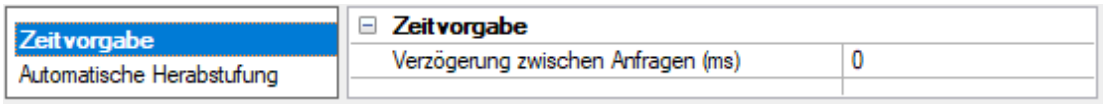
Kommunikationsanforderungen mit demselben Gerät. Der gültige Bereich liegt zwischen 1 und 30 Sekunden. Die Standardeinstellung ist normalerweise 3 Sekunden, kann jedoch abhängig vom jeweiligen Treiber unterschiedlich sein. Wenn diese Einstellung nicht vom Treiber unterstützt wird, ist sie deaktiviert.
Hinweis: Aufgrund der Art der UDP-Verbindungen ist die Einstellung für Verbindungs-Timeout nicht anwendbar, wenn die Kommunikation über UDP erfolgt.

Anforderungs-Timeout: Geben Sie ein von allen Treibern verwendetes Intervall an, um zu bestimmen, wie lange der Treiber abschließend auf eine Antwort vom Zielgerät wartet. Der gültige Bereich liegt zwischen 50 und 9999999 Millisekunden (167 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: Geben Sie an, 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: Geben Sie an, 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.

Eigenschaftengruppen	Automatische Herabstufung	
Allgemein	Herabstufen bei Fehler	Aktivieren
Scan-Modus	Timeout bis zum Herabstufen	3
Zeitvorgabe	Herabstufungszeitraum (ms)	10000
Automatische Herabstufung	Anfragen verwerfen, wenn herabgestuft	Deaktivieren

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.*

Eigenschaftengruppen	Tag-Generierung	
Allgemein	Bei Gerätestart	Nicht beim Start erstellen
Scan-Modus	Bei doppeltem Tag	Bei Erstellen löschen
Zeitvorgabe	Elternteilgruppe	
Automatische Herabstufung	Automatisch generierte Untergruppen zulassen	Aktivieren
Tag-Generierung		

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.

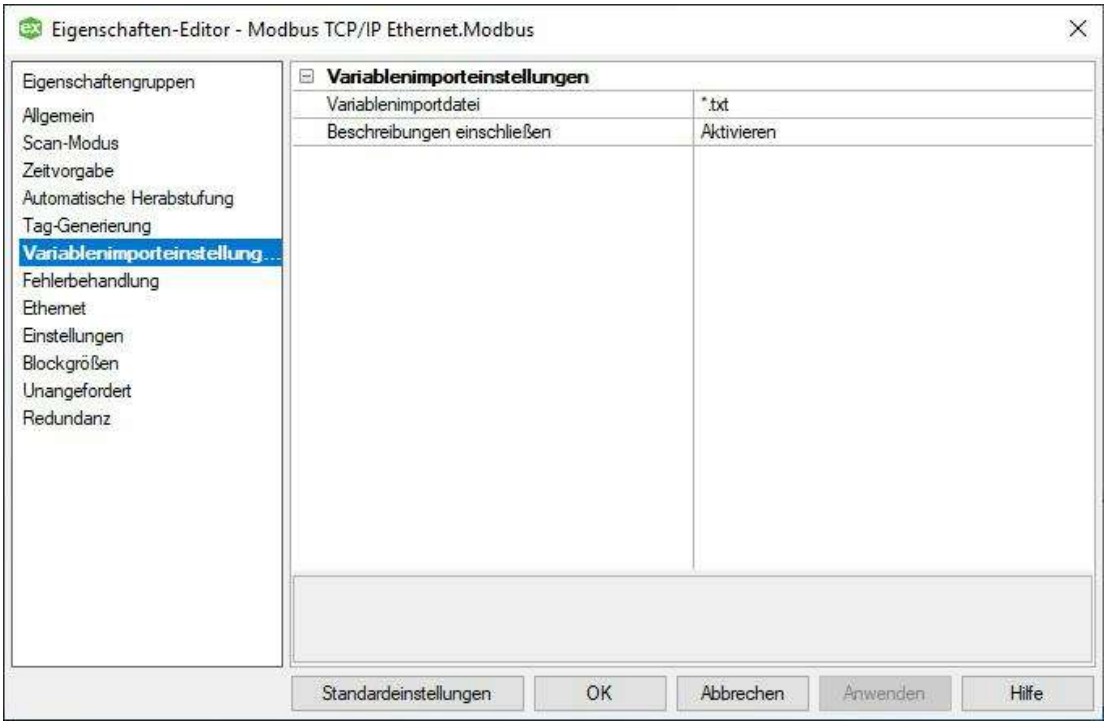
Elternteilgruppe: Mit dieser Eigenschaft wird verhindert, dass sich automatisch generierte Tags mit Tags vermischen, die manuell eingegeben wurden, indem eine Gruppe festgelegt wurde, die für automatisch generierte Tags verwendet werden soll. Der Name der Gruppe kann bis zu 256 Zeichen lang sein. Diese 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 – Variablenimporteinstellungen

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

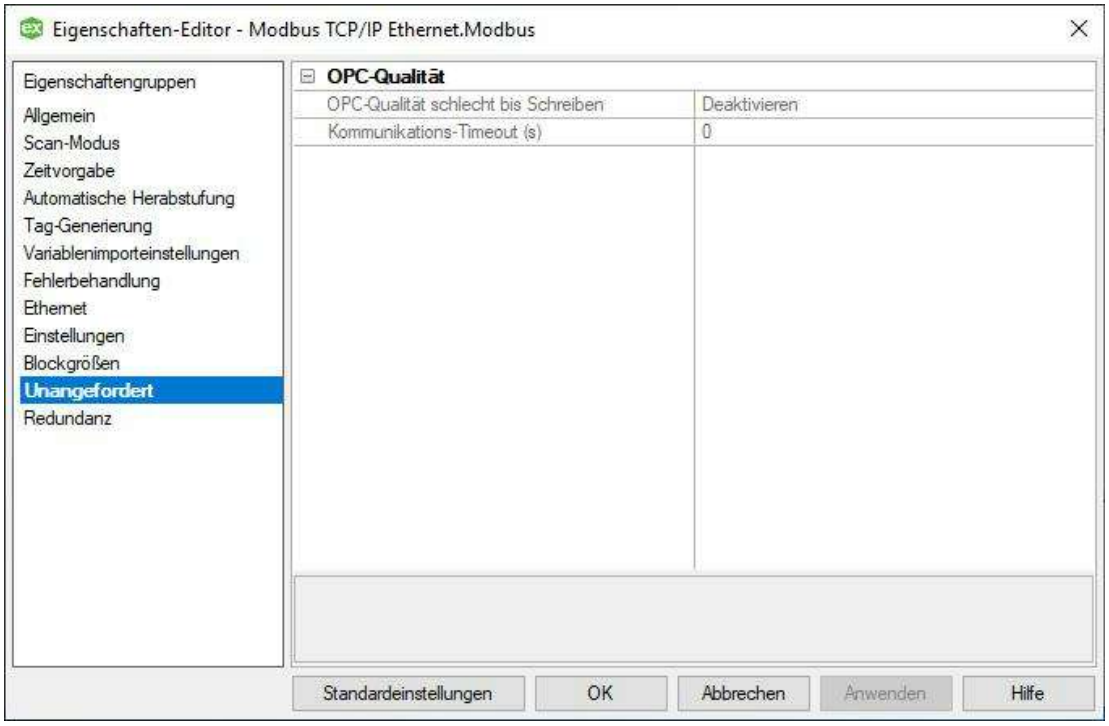


Variablenimportdatei: Dieser Parameter gibt den genauen Speicherort der Variablenimportdatei an, die der Treiber verwenden soll, wenn die Funktion zur automatischen Tag-Datenbankgenerierung aktiviert ist.

Beschreibungen einschließen: Bei Aktivierung importiert diese Option Tag-Beschreibungen (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 – Unangefordert



OPC-Qualität

OPC-Qualität schlecht bis Schreiben: Steuert die anfängliche OPC-Qualität von an diesen Treiber angehängten Tags. Bei Deaktivierung haben alle Tags den Anfangswert 0 und gute OPC-Qualität. Dies ist der Standardzustand. Bei Aktivierung haben alle Tags den Anfangswert 0 und schlechte OPC-Qualität. Die Qualität des Tags bleibt schlecht, bis ein Modbus-Client oder eine Client-Anwendung in alle vom Tag referenzierten Coils oder Register geschrieben hat. Zum Beispiel referenziert ein Tag mit Adresse 400001 und Datentyp DWord zwei Halteregister: 400001 und 400002. Dieses Tag zeigt erst gute Qualität an, wenn in beide Halteregister geschrieben wurde.

Hinweis: Wenn sich das Gerät nicht im Server-Modus befindet, wird diese Option abgeblendet.

Kommunikations-Timeout: Legt fest, wie lange (in Sekunden) der Treiber auf eine eingehende Anforderung wartet, bevor er die Tag-Qualität des Geräts als schlecht festlegt. Nachdem das Timeout eingetreten ist, besteht die einzige Möglichkeit, das Timeout zurückzusetzen und eine normale Verarbeitung aller Tags zu ermöglichen, darin, dass Sie die Kommunikation mit dem Remote-Client wiederherstellen oder das Kommunikations-Timeout deaktivieren, indem Sie es auf 0 (Null) setzen. Bei Aktivierung liegt der gültige Bereich zwischen 1 und 64.800 Sekunden (18 Stunden).

Hinweise:

1. Wenn eine eingehende Anforderung für ein nicht vorhandenes Severgerät (bzw. die eine nicht vorhandene Stations-ID) ankommt, wird die Anforderung zu Station 0 umgeleitet. In diesem Fall tritt das Timeout für ein Servergerät mit Stations-ID 0 nicht ein, selbst wenn es für den Timeout-Zeitraum keine Remote-Kommunikation explizit empfängt.

2. Server-Geräte erfordern das Modell "Modbus" und die Geräte-ID *IP-Adresse.yyy*, wobei *IP-Adresse* die IP-Adresse des im Kanal zugewiesenen Netzwerkadapters oder die lokale Loopback-IP-Adresse des PCs sein kann, auf dem der Treiber ausgeführt wird. Z.B. 127.0.0.1.1 oder 127.xxx.xxx.xxx, wobei *xxx* = 0–255 ist, und *yyy* (Stations-ID) = 0–255.

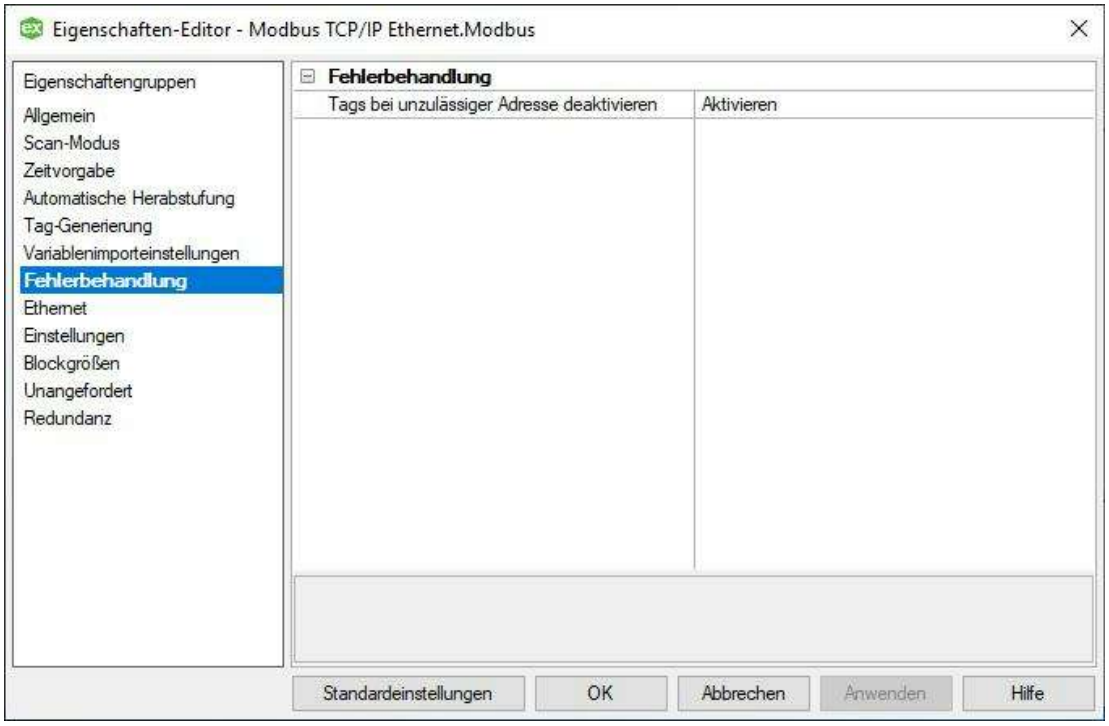
3. Wenn die erste unangeforderte Anforderung für ein Servergerät empfangen wird, zeigt das Ereignisprotokoll die folgende Informationsmeldung an: "<Datum>__<time>__<Ebene>__<source>__<Ereignis>". Beispiel: "4.2.2011__16:53:10__Information__Modbus-TCP/IP-Ethernet__Speicher für Servergerät <Servernummer> erstellt".

Überlegungen zu Modbus-Clients und Modbus-Servern

Folgende Hinweise beziehen sich sowohl auf Modbus-Clientgeräte als auch auf Modbus-Servergeräte.

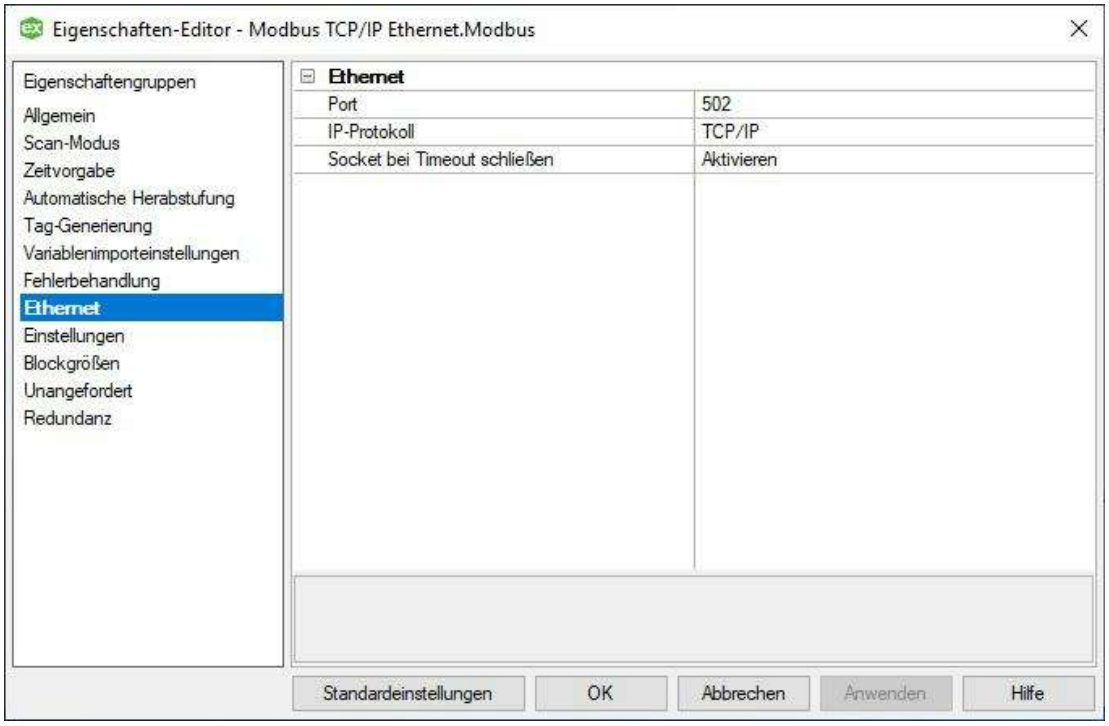
- Es empfiehlt es sich nicht, ein Postfach-Gerät und ein Modbus-Gerät auf demselben Rechner unterzubringen. Da ein Client nur Daten von jeweils einem dieser Geräte auf einmal erhält, ist es unsicher, von welchem er Daten erhält.
- Es wird empfohlen, Client- und Servergeräte im Serverprojekt auf separaten Kanälen zu platzieren, damit die Tags der Servergeräte optimal verarbeitet werden.
- Wenn ein OPC-Client verbunden ist, kann die Geräte-ID nur geändert werden, sofern dabei der Modus des Geräts nicht geändert wird (Client in Server bzw. Server in Client). Der Modus wird durch das Ändern der Loopback-Adresse oder der lokalen IP-Adresse in eine anderen IP-Adresse und umgekehrt geändert. Die Loopback-Adresse und die lokale IP-Adresse (des PCs, auf dem der Treiber ausgeführt wird) geben den Server-Modus (unangeforderten Modus) an, und alle anderen IP-Adressen geben den Client-Modus des Geräts an. Wenn kein OPC-Client verbunden ist, kann der Modus auf beliebige Weise geändert werden (z.B. Client in Client, Client in Server, Server in Server oder Server in Client).
  Hinweis: Jede Adresse im Format 127.xxx.xxx.xxx, wobei *xxx* im Bereich 0–255 liegt, ist die Loopback-Adresse.
- Die Datenkodierungs-Gruppeneinstellungen müssen in Client- und Servergeräten identisch sein, z.B. wenn ein als Modbus-Client konfiguriertes Gerät bei der Kommunikation das Geräte-Setup als ein Modbus-Server verwendet.
- Das gesamte Serverprojekt erlaubt maximal 255 Servergeräte, eines für jede eindeutige Server-ID. Dieselbe Server-ID kann nicht für mehrere Kanäle verwendet werden.
- Der Server betrachtet JEDE Loopbackadresse (127.x.x.x) oder Localhost-IP als eine Referenz auf sich selbst und erstellt einen gemeinsam genutzten Speicherbereich, der eindeutig für die Server-ID ist. Dieselbe ID in mehreren Kanälen ist dasselbe Servergerät, welches denselben Registerspeicher verwendet.
- Muss dieselbe Server-ID mehr als einmal in einem Projekt verwendet werden, so sollten Tag-Adressbereiche gewählt werden, die nicht mit anderen Instanzen derselben Servergeräte-IDs übereinstimmen. Verwenden mehrere Kanäle/Geräte denselben Tag-Adressbereich in derselben Server-ID so können Einstreuungen sowie Datenverluste auftreten.
- Für diesen Treiber werden die Begriffe "Server" und "unangefordert" synonym verwendet.

Geräteeigenschaften – Fehlerbehandlung



Tags in unzulässigen Adressen deaktivieren: Wählen Sie "Aktivieren", damit der Treiber das Abrufen eines Datenblocks anhält, 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. Wählen Sie "Deaktivieren", damit der Treiber das Abrufen des Datenblocks trotz Fehlern fortsetzt. Die Standardeinstellung ist "Aktiviert".

Geräteeigenschaften – Ethernet

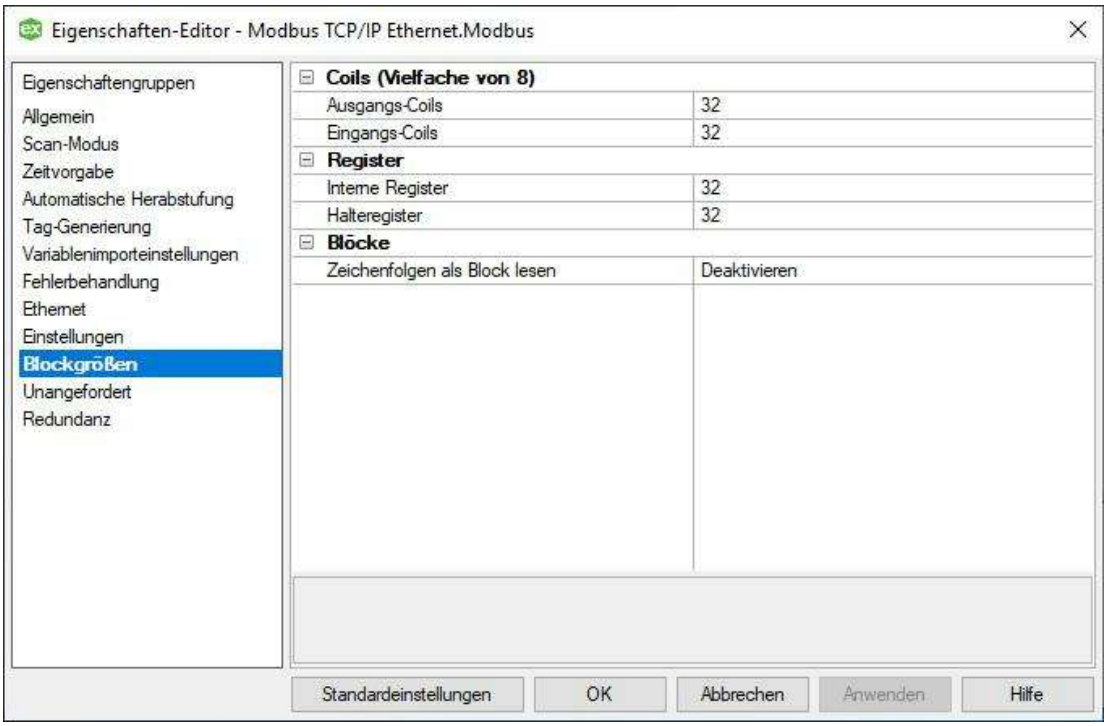


Port: Gibt die Port-Nummer an, für deren Verwendung das Remote-Gerät konfiguriert ist. Der gültige Bereich liegt zwischen 0 und 65535. Der Standardwert ist 502. Diese Port-Nummer wird verwendet, wenn sich das Geräteobjekt im Modbus-Client-Modus befindet und angeforderte Anfragen an ein Gerät sendet.
Wenn das Port-System-Tag verwendet wird, wird die Einstellung für die Port-Nummer geändert. Weitere Informationen finden Sie unter ***Treibersystem-Tag-Adressen****.*

IP-Protokoll: Gibt an, ob der Treiber eine Verbindung zum Remote-Gerät unter Verwendung des User Datagram Protocol (UDP) oder des Transmission Control Protocol (TCP/IP) herstellen soll. Die Client- und Servereinstellungen müssen übereinstimmen. Wenn beispielsweise die Server-IP-Protokolleinstellung TCP/IP ist, dann muss die Client-IP-Protokolleinstellung für dieses Gerät ebenfalls TCP/IP sein.
Hinweis: Dieser Treiber erfordert Winsock V1.1 oder höher.

Socket bei Timeout schließen: Gibt an, ob der Treiber eine TCP-Socket-Verbindung schließen soll, wenn das Gerät nicht bis zum Timeout antwortet. Bei Aktivierung, der Standardeinstellung, schließt der Treiber die Socket-Verbindung bei Timeout. Bei Deaktivierung verwendet der Treiber weiterhin denselben TCP-Socket, bis ein Fehler auftritt, das physische Gerät den Socket schließt oder der Treiber heruntergefahren wird.
Hinweis: Bei einem Socket-Fehler schließt Modbus TCP/IP-Ethernet-Treiber die Socket-Verbindung.

Geräteeigenschaften – Blockgrößen



Coils

Ausgangs-Coils: Gibt die Ausgangsblockgröße in Bit an. Coils können von 8 bis 2000 Punkten (Bit) auf einmal gelesen werden. Der Standardwert ist 32.

Eingangs-Coils: Gibt die Eingangsblockgröße in Bit an. Coils können von 8 bis 2000 Punkten (Bit) auf einmal gelesen werden. Der Standardwert ist 32.

Register

Interne Register: Gibt die Blockgröße für interne Register in Bit an. Bis zu 120 Modbus-Standardregister (16 Bit) können auf einmal gelesen werden. Der Standardwert ist 32.

Halteregister: Gibt die Blockgröße für Halteregister in Bit an. Bis zu 120 Modbus-Standardregister (16 Bit) können auf einmal gelesen werden. Der Standardwert ist 32.

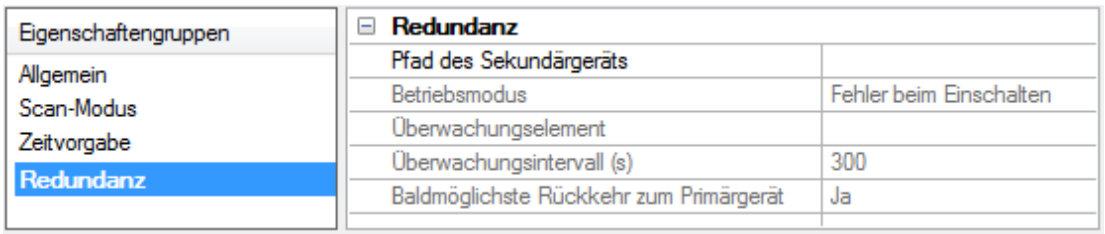
Blöcke

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.

Hinweise:

1. Auf die Modelle für Instromet, Roxar und Fluenta (die 32-Bit- und 64-Bit-Register unterstützen) muss besonderes Augenmerk gelegt werden. Das Modbus-Protokoll beschränkt die Blockgröße auf maximal 256 Byte. Dies entspricht einer maximalen Blockgröße von 64 32-Bit-Registern oder 32 64-Bit-Registern für diese Modelle.
2. Das CEG-Modell unterstützt Coil-Blockgrößen zwischen 8 und 8000 in Vielfachen von 8 sowie Registerblockgrößen zwischen 1 und 500. Dieses Modell darf nur mit CEG-Geräten verwendet werden.
3. 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.
4. Einige Geräte unterstützen möglicherweise bei Standardgröße keine Block-Lesevorgänge. Kleinere Modicon-SPSs und Nicht-Modicon-Geräte unterstützen möglicherweise nicht die vom Modbus-Ethernet-Netzwerk unterstützten maximalen Datenübertragungslängen.
5. Einige Geräte können nicht zusammenhängende Adressen enthalten. Wenn dies der Fall ist und der Treiber versucht, einen Datenblock zu lesen, der einen nicht definierten Arbeitsspeicher umfasst, wird die Anfrage möglicherweise zurückgewiesen.

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**.*

Konfigurations-API Modbus Ethernet-Beispiel

Für eine Liste von Kanal- und Gerätedefinitionen sowie Aufzählungen greifen Sie mit dem REST-Client auf die folgenden Endpunkte zu.

Kanaldefinitionen

Endpunkt (GET):

```
https://<Host-Name_oder_IP>:<Por-
t>/config/v1/doc/drivers/Modbus%20TCP%2FIP%20Ethernet/channels
```

Gerätedefinitionen

Endpunkt (GET):

```
https://<Host-Name_oder_IP>:<Por-
t>/config/v1/doc/drivers/Modbus%20TCP%2FIP%20Ethernet/devices
```

Modbus-Kanal erstellen

Endpunkt (POST):

```
https://<Host-Name_oder_IP>:<Port>/config/v1/project/channels
```

Textkörper:

```
{ "common.ALLTYPES_NAME": "MyChannel", "servermain.MULTIPLE_TYPES_DEVICE_DRIVER":
"Modbus TCP/IP Ethernet" }
```

Siehe auch: Anhang A *für eine Liste von Kanaleigenschaften.*

Modbus-Gerät erstellen

Endpunkt (POST):

```
https://<Host-Name_oder_IP>:<Port>/config/v1/project/channels/MyChannel/devices
```

Textkörper:

```
{ "common.ALLTYPES_NAME": "MyDevice", "servermain.DEVICE_ID_STRING":
"<192.160.0.1>.0", "servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Modbus TCP/IP Ether-
net" }
```

Siehe auch: Anhang B *für eine Liste von Geräteeigenschaften.*

Geräte-ID-Aktualisierung

Aktualisieren Sie die Geräte-ID mithilfe eines "PUT"-Befehls von einem REST-Client.
Das folgende Endpunkt-Beispiel verweist auf die Projektkonfiguration "demo-project.json" mit dem Kanalnamen "ModbusTCPIP" sowie dem Gerätenamen "ModbusDevice".

Geräte-ID-Beispiel

Endpunkt (PUT):

```
https://<Host-Name_oder_IP>:<Por-
t>/config/v1/project/channels/ModbusTCPIP/devices/ModbusDevice
```

Textkörper:

```
{ "project_id": <Projekt-ID_von_GET>, "servermain.DEVICE_ID_STRING": "<IP-Adresse>" }
```

Modbus-Tags erstellen

Endpunkt (POST):

```
https://<Host-Name_oder_IP>:<Por-
t>/config/v1/project/channels/MyChannel/devices/MyDevice/tags
```

Textkörper:

```
[ { "common.ALLTYPES_NAME": "MyTag1", "servermain.TAG_ADDRESS": "40001" } { "com-
mon.ALLTYPES_NAME": "MyTag2", "servermain.TAG_ADDRESS": "40002" } ]
```

***Siehe auch**: Anhang C für eine Liste von Tag-Eigenschaften.*

Weitere Informationen zum Konfigurieren von Projekten über die Konfigurations-API finden Sie in der Server-Hilfe.

Aufzählungen

Einige Eigenschaften, z.B. das Gerätemodell, haben Werte, die einer Aufzählung zugeordnet sind. Eine gültige Liste mit Aufzählungen und ihren Werten kann abgerufen werden, indem der Geräte-Endpunkt mit content=property_definitions abgefragt wird oder die Dokumentationsdefinitions-Endpunkte.

Beispiel: Sollen die Eigenschaftsdefinitionen für ein Gerät mit dem Namen "MyDevice" unter einem Kanal namens "MyChannel" angezeigt werden, so wird die GET-Anforderung an die folgende Adresse gesendet:

```
https://<Host-Name_oder_IP>:<Por-
t>/config/v1/project/channels/MyChannel/devices/MyDevice/?content=property defi-
nitions
```

Eigenschaftsdefinitionen sind auch für andere Objekte wie Kanäle oder Tags verfügbar.

Alternativ können die Definitionen der Kanal- und Geräteeigenschaften für den Treiber unter folgender Adresse angezeigt werden, falls dies in den Einstellungen für die Konfigurations-API aktiviert ist:

```
https://<Host-Name_oder_IP>:<Port>/config/v1/doc/drivers/<drivername>/Channels
```

```
https://<Host-Name_oder_IP>:<Port>/config/v1/doc/drivers/<drivername>/Devices
```

Beispiele für Datentyp-Aufzählungen

Das Abfragen des Dokumentationsendpunkts nach Tag-Datentypen bietet die folgenden Aufzählungen:

```
{
   "Standard": -1,
   "String": 0,
   "Boolean": 1,
   "Char": 2,
   "Byte": 3,
```

```
  "Short": 4,
  "Word": 5,
  "Long": 6,
  "DWord": 7,
  "Float": 8,
  "Double": 9,
  "BCD": 10,
  "LBCD": 11,
  "Date": 12,
  "LLong": 13,
  "QWord": 14,
  "String-Array": 20,
  "Boolean-Array": 21,
  "Char-Array": 22,
  "Byte-Array": 23,
  "Short-Array": 24,
  "Word-Array": 25,
  "Long-Array": 26,
  "DWord-Array": 27,
  "Float-Array": 28,
  "Double-Array": 29,
  "BCD-Array": 30,
  "LBCD-Array": 31,
  "Date-Array": 32,
  "LLong-Array": 33,
  "QWord-Array": 34
}
```

Hinweis: Unterstützte Datentypen variieren je nach Protokoll und Treiber.

Gerätemodellaufzählungen

Die Werte der Eigenschaft Gerätemodell sind den folgenden Aufzählungen zugeordnet. Die folgende Tabelle dient nur als Referenz. Die Informationen am Geräteendpunkt sind die vollständige und aktuelle Informationsquelle:

```
https://<Host-Name_oder_IP>:<Por-
t>/config/v1/doc/drivers/Modbus%20TCP%2FIP%20Ethernet/Channels
```

```
https://<Host-Name_oder_IP>:<Por-
t>/config/v1/doc/drivers/Modbus%20TCP%2FIP%20Ethernet/Devices
```

Aufzählung	Gerätemodell
0	Modbus
1	Postfach
2	Instromet
3	Roxar RFM
4	Fluenta FGM
5	Applicom
6	CEG

Automatische Tag-Datenbankgenerierung

Dieser Treiber unterstützt die automatische Tag-Datenbankgenerierung, die es Treibern ermöglicht, automatisch Tags zu erstellen, die auf vom Kontaktplan des Geräts verwendete Datenpunkte zugreifen. 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 die Tag-Generierung gestartet wurde, ob während der Verarbeitung der Variablenimportdatei Fehler aufgetreten sind und wann der Prozess abgeschlossen wurde.

Weitere Informationen dazu finden Sie in der Hilfedokumentation zum Server.

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 Geräteprogrammieranwendungen wie Concept und ProWORX generiert werden. Die Importdatei muss im durch Semikola getrennten TXT-Format vorliegen, dem Standard-Exportdateiformat der Geräteprogrammieranwendung Concept.

Siehe auch: **Aus benutzerdefinierten Anwendungen importieren**

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).

Aus benutzerdefinierten Anwendungen importieren

Benutzerdefinierte Tags können im folgenden CSV-Dateiformat importiert werden:

[Datensatztyp] ; [Variablenname] ; [Datentyp] ; [Adresse] ; [Wert definieren] ; [Kommentar], wobei Folgendes gilt:

- **Datensatztyp:** Dieses in der Concept-Software verwendete Flag stellt eine weitere Möglichkeit dar, Tags zu importieren. Er kann N oder E lauten: Beide Flags werden gleich behandelt.
- **Variablenname:** Dies ist der Name des statischen Tags im Server. Er kann bis zu 256 Zeichen lang sein.
- **Datentyp:** Dies ist der Datentyp des Tags. Folgende Datentypen werden unterstützt:
  - BOOL
  - DINT
  - INT
  - REAL (32-Bit-Float)
  - UDINT
  - UINT
  - WORD
  - BYTE
  - TIME (als DWord behandelt)
  - STRING
- **Adresse:** Dies ist die Modbus-Adresse des Tags. Sie kann bis zu 16 Zeichen lang sein.
- **Wert definieren**: Dies wird ignoriert und sollte leer gelassen werden.
- **Anmerkung**: Dies ist die Beschreibung des Tags im Server. Sie kann bis zu 255 Zeichen lang sein.

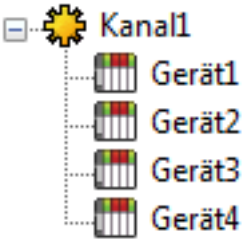
Beispiele

- N;Amps;WORD;40001;;Current in
- N;Volts;WORD;40003;;Volts in
- N;Temperature;REAL;40068;;Tank temp

Kommunikation optimieren

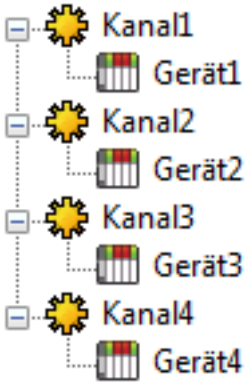
Modbus TCP/IP-Ethernet-Treiber wurde dafür konzipiert, eine optimale Leistung mit der geringsten Auswirkung auf die Gesamtleistung des Systems zu bieten. Zwar ist der Treiber schnell, doch gibt es eine Reihe von Richtlinien zur Optimierung der Anwendung und zum Erreichen der maximalen Leistung.

Der Server bezeichnet Kommunikationsprotokolle wie Modbus-Ethernet als Kanal. Jeder in der Anwendung definierte Kanal stellt einen separaten Ausführungspfad im Server dar. Sobald ein Kanal festgelegt wurde, muss eine Reihe von Geräten unter diesem Kanal definiert werden. Jedes dieser Geräte stellt einen einzelnen Modbus-Controller dar, von dem Daten gesammelt werden. Zwar ermöglicht diese Methode zum Definieren der Anwendung ein hohes Leistungsniveau, doch kann damit nicht vollständig Nutzen aus dem Treiber oder Netzwerk gezogen werden. Ein Beispiel dafür, wie die Anwendung bei Konfiguration mit einem Kanal möglicherweise aussehen kann, wird im Folgenden gezeigt.



Jedes Gerät wird unter einem einzelnen Modbus-Ethernet-Kanal definiert. In dieser Konfiguration muss sich der Treiber schnellstmöglich von einem Gerät zum nächsten bewegen, um Informationen in einem effektiven Intervall zu sammeln. Je mehr Geräte hinzugefügt oder je mehr Informationen von einem einzelnen Gerät angefordert werden, desto mehr leidet die Aktualisierungsrate insgesamt.

Wenn Modbus TCP/IP-Ethernet-Treiber nur einen Kanal definieren könnte, würde das Beispiel oben die einzige verfügbare Option darstellen. Der Treiber kann jedoch bis zu 1024 Kanäle definieren. Durch Verwenden mehrerer Kanäle wird die Arbeitsbelastung bei der Datensammlung verteilt, indem mehrere Anfragen gleichzeitig an das Netzwerk gestellt werden. Ein Beispiel dafür, wie dieselbe Anwendung aussehen kann, wenn sie mit mehreren Kanälen konfiguriert wird, um die Leistung zu verbessern, wird im Folgenden gezeigt.



Jedes Gerät kann unter seinem eigenen Kanal definiert werden. In dieser Konfiguration wird ein einziger Ausführungspfad dediziert für das Sammeln von Daten von jedem Gerät eingesetzt. Wenn die Anwendung über 1024 oder weniger Geräte verfügt, kann sie auf diese Weise optimiert werden.

Die Leistung wird verbessert, selbst wenn die Anwendung über mehr Geräte verfügt. Zwar sind weniger Geräte möglicherweise ideal, doch zieht die Anwendung weiterhin Nutzen aus zusätzlichen Kanälen. Obwohl das Verteilen der Gerätelast auf alle Kanäle zur Folge hat, dass sich der Server erneut von Gerät zu Gerät bewegt, kann dies mit weit weniger zu bearbeitenden Geräten auf einem einzigen Kanal erfolgen.

Blockgröße

Die Blockgröße kann sich auf die Leistung von Modbus TCP/IP-Ethernet-Treiber auswirken. Der Blockgröße-Parameter steht auf jedem Gerät zur Verfügung, das unter den **Blockgröße**-Einstellungen für Geräteeigenschaften definiert wird. Die Blockgröße bezieht sich auf die Anzahl von Registern oder Bits, die von einem Gerät auf einmal angefordert werden kann. Die Leistung des Treibers kann verfeinert werden, indem die Blockgröße auf 1 bis 120 Register und 8 bis 2000 Bit konfiguriert wird.

Tipps:

- Zusätzliche Leistungsgewinne können durch Aktivieren der Eigenschaft **Socket bei Timeout schließen** und durch Anpassen der Timeout- und Zeitvorgaben-Eigenschaften erzielt werden.

- Die Leistung wird auch durch die Socket-Verwendung beeinflusst (siehe weiter unten).

*Weitere Informationen finden Sie unter **Ethernet-Eigenschaften**, **Kommunikations-Timeouts** und **Zeitvorgabe**.*

Socket-Verwendung

Benutzer können angeben, ob der Treiber einen einzelnen Socket für alle Geräte auf einem Kanal verwenden soll oder mehrere Sockets für die Kommunikation mit Geräten auf einem einzelnen Kanal. Wenn der Treiber einen Port verwendet, kann kein anderes System auf das Zielgerät zugreifen. Daher muss in manchen Umgebungen möglicherweise eine Ausgleichung des Gerätezugriffs erfolgen. Wenn der Treiber keine Verbindung aufrechterhalten soll oder wenn das Gerät nur über eine begrenzte Anzahl an Verbindungen verfügt, kann der Benutzer die Anzahl der Sockets begrenzen, die der Treiber verwenden kann.

Die Möglichkeit, den Treiber im Modus mit einem Socket zu verwenden, ist wichtig bei Verwendung des Treibers für die Kommunikation mit einem Modbus-Ethernet-zu-Modbus-RTU-Bridge-Produkt. Die meisten dieser Produkte lassen es zu, mehrere serienbasierte RS-485-Geräte mit einer einzelnen Modbus-Ethernet-zu-Modbus-RTU-Bridge zu verbinden. Dies kann durch Festlegen von "Ein Socket pro Kanal" oder "Ein oder mehrere Sockets pro Gerät" erfolgen, wobei "Max. Sockets pro Gerät" auf "1" festgelegt werden muss (*siehe **Kanal-Ethernet**-Einstellungen*).

Wenn mehr als ein Socket für die Verwendung konfiguriert ist, verteilt der Treiber die Daten für die Lese- oder Schreibvorgänge auf alle für das Zielgerät verwendeten verfügbaren Sockets. Der Treiber verwendet **nur** die Anzahl der Sockets, die zur Verarbeitung der Daten benötigt werden, bis zur angegebenen Maximalzahl.

Wenn Geräteobjekte unter einem Kanal unterschiedliche IP-Adressen und/oder Ports als Ziel haben, wird der Socket nur bei Bedarf geschlossen und wieder geöffnet. Dies wird empfohlen, wenn ein Ethernet-zu-seriellem-Gateway-Gerät eine Reihe serieller Geräte verwaltet. Gateways (und Geräte) grenzen zum Schutz gegen Konflikte normalerweise die Anzahl der gleichzeitigen Verbindungen ein. Vermeiden Sie es, diese Grenzwerte zu überschreiten. Wenn diese Grenzwerte überschritten werden, gibt der Treiber Meldungen zu Verbindungsfehlern aus.

Geräteantwortmeldungen können vom Treiber zur gleichen Zeit empfangen werden. Die Antworten des Geräts werden sequenziell vom einzelnen Thread auf Kanalebene verarbeitet. Diese Verarbeitung von Daten auf Kanalebene kann jedoch sehr schnell (innerhalb von Millisekunden) ablaufen. Wenn die Verwendung mehrerer Sockets konfiguriert ist, kann daher eine deutliche Leistungsverbesserung erzielt werden.

Tipp: Die Leistung wird auch durch die oben erwähnte Blockgröße beeinflusst, die sich auf den Datentransfer aller Sockets auswirken kann.

Datentypbeschreibung

Datentyp	Beschreibung
Boolean	Einzelnes Bit
Word	16-Bit-Wert ohne Vorzeichen Bit 0 ist das Low-Bit Bit 15 ist das High-Bit
Short	16-Bit-Wert mit Vorzeichen Bit 0 ist das Low-Bit Bit 14 ist das High-Bit Bit 15 ist das Vorzeichen-Bit
DWord	32-Bit-Wert ohne Vorzeichen Bit 0 ist das Low-Bit Bit 31 ist das High-Bit
Long	32-Bit-Wert mit Vorzeichen Bit 0 ist das Low-Bit Bit 30 ist das High-Bit Bit 31 ist das Vorzeichen-Bit
BCD	Gepacktes 2-Byte-BCD Der Wertebereich liegt zwischen 0 und 9999. Für Werte außerhalb dieses Bereichs ist das Verhalten nicht definiert.
LBCD	Gepacktes 4-Byte-BCD Der Wertebereich liegt zwischen 0 und 99999999. Für Werte außerhalb dieses Bereichs ist das Verhalten nicht definiert.
String	Mit Null beendete ASCII-Zeichenfolge Wird im Modbus-Modell unterstützt, schließt eine Auswahl der Byte-Reihenfolgen Hi-Lo und Lo-Hi ein.
Double*	64-Bit-Gleitkommawert 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 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 werden die Standardeinstellungen angenommen, d.h., dass 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.

Applicom-Adressierung
CEG-Adressierung
Fluenta-Adressierung
Instromet-Adressierung
Postfach-Adressierung
Modbus-Adressierung
Roxar-Adressierung

Treibersystem-Tag-Adressierung

Interne Tags

Tag	Beschreibung	Datentyp	Zugriff
Port	Mithilfe des Port-System-Tags kann eine Client-Anwendung die Einstellung für die Port-Nummer lesen und schreiben. Schreibvorgänge in diesem Tag führen dazu, dass die Verbindung zwischen Treiber und Gerät getrennt wird, sowie zu einem Versuch, die Verbindung zum angegebenen Port wiederherzustellen.	Word, Short, DWord, Long	Lesen/Schreiben

Hinweise:

- Die Geräte-Port-Einstellung wird vom Treiber nicht für Serverkommunikation verwendet.
- Für diesen Treiber werden die Begriffe "Server" und "unangefordert" synonym verwendet.
- Durch Änderungen an diesem Tag wird das Projekt geändert, was dazu führt, dass Sie vom Server aufgefordert werden, das Projekt vor dem Herunterfahren zu speichern.

System-Tags

Tag	Beschreibung	Datentyp	Zugriff
_CEGExtension	Dieses Tag wird nur für Geräte des CEG-Modells verwendet. Mit seiner Hilfe kann die Geräteeigenschaft **CEG-Erweiterung** von einer Client-Anwendung geändert werden.	Boolean	Lesen/Schreiben
_InputCoilBlockSize	Mithilfe dieses Tags kann die Eigenschaft der Blockgröße für Eingangs-Coils von einer Client-Anwendung geändert werden.	DWord	Lesen/Schreiben
_OutputCoilBlockSize	Mithilfe dieses Tags kann die Eigenschaft der Blockgröße für Ausgangs-Coils von einer Client-Anwendung geändert werden.	DWord	Lesen/Schreiben
_InternalRegisterBlockSize	Mithilfe dieses Tags kann die Eigenschaft der Blockgröße für Interne Register von einer Client-Anwendung geändert werden.	DWord	Lesen/Schreiben

Tag	Beschreibung	Datentyp	Zugriff
_HoldingRegisterBlockSize	Mithilfe dieses Tags kann die Eigenschaft der Blockgröße für Halteregister von einer Client-Anwendung geändert werden.	DWord	Lesen/Schreiben

Hinweis: Durch Änderungen an diesem Tag wird das Projekt geändert, was dazu führt, dass Sie vom Server aufgefordert werden, das Projekt vor dem Herunterfahren zu speichern.

Siehe auch: *Ethernet*

Beschreibung von Funktionscodes

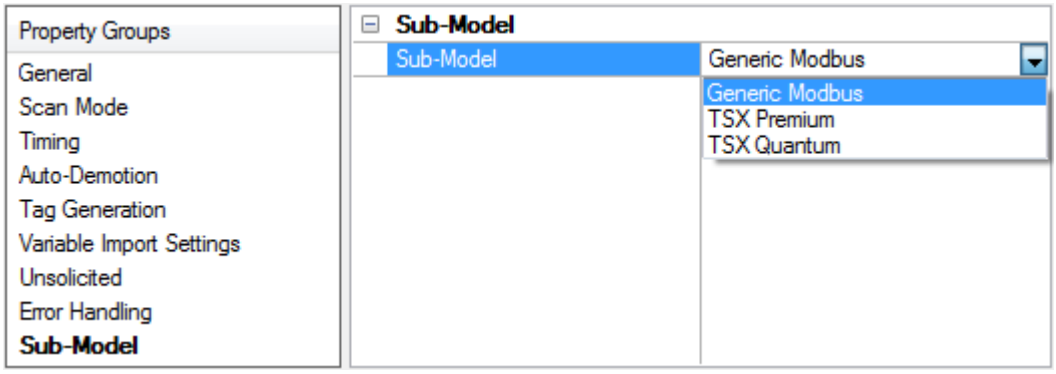
Die in der Tabelle unten angezeigten Funktionscodes werden von den Gerätemodellen Modbus und Applicom unterstützt.

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

Untergeordnetes Applicom-Modell und Adressierung

Applicom-Geräte unterstützen drei untergeordnete Applicom-Modelle. Wählen Sie das entsprechende untergeordnete Modell für das Gerät aus, das verbunden werden soll. Klicken Sie auf den nachstehenden Link des untergeordneten Modells, um Adressinformationen abzurufen.

Generischer Modbus
TSX Premium
TSX Quantum



Generische Modbus-Adressierung

Alle Funktionscodes werden als Dezimalzahlen angezeigt. Weitere Informationen finden Sie unter **Beschreibung von Funktionscodes**.

Ausgangs-Coils

Adresse	Bereich	Datentyp	Zugriff	Funktionscode
Bxxxxx	0–65535	Boolean	Lesen/Schreiben	01, 05, 15

Array-Unterstützung
Arrays werden für die Ausgangs-Coil-Adressen unterstützt. Die Syntax zum Deklarieren eines Arrays lautet wie folgt:
Bxxxxx_cols mit einer angenommenen Zeilenanzahl von 1.
Bxxxxx_rows_cols.

Die Basisadresse + (rows * cols) darf den Wert 65535 nicht überschreiten. Die Gesamtanzahl der angeforderten Coils darf die für dieses Gerät angegebene Blockgröße für Ausgangs-Coils nicht überschreiten.

Eingangs-Coils

Adresse	Bereich	Datentyp	Zugriff	Funktionscode
BIxxxxx	0–65535	Boolean	Schreibgeschützt	02

Array-Unterstützung
Arrays werden für die Eingangs-Coil-Adressen unterstützt. Die Syntax zum Deklarieren eines Arrays lautet wie folgt:
BIxxxxx_cols mit einer angenommenen Zeilenanzahl von 1.
BIxxxxx_rows_cols.

Die Basisadresse + (rows * cols) darf den Wert 65535 nicht überschreiten. Die Gesamtanzahl der angeforderten Coils darf die für dieses Gerät angegebene Blockgröße für Eingangs-Coils nicht überschreiten.

Interne Register

Die Standard-Datentypen werden **fett** dargestellt.

Arrays werden für interne Registerpositionen für alle Datentypen außer Boolean und String unterstützt.

Hinweis: Bei Servergeräten ist für schreibgeschützte Speicherorte das Lesen/Schreiben möglich.

Adresse	Bereich	Datentyp	Zugriff	Funktionscode
WIxxxxx	0–65535 0–65534 0–65532	**Word**, Short, BCD Float, DWord, Long, LBCD Double	Schreibgeschützt	04
WIxxxxx.bb	xxxxx=0–65535 bb=0/1–15/16*	**Boolean**	Schreibgeschützt	04
WIxxxxx:Xbb	xxxxx=0–65535 bb=0/1–15/16*	**Boolean**	Schreibgeschützt	04
DIxxxxx	0–65534	**DWord**	Schreibgeschützt	04

Adresse	Bereich	Datentyp	Zugriff	Funktionscode
Flxxxxx	0–65534	**Float**	Schreibgeschützt	04
Wlxxxxx_S	0–65535	**Short**	Schreibgeschützt	04
Wlxxxxx_B	0–65535	**BCD**	Schreibgeschützt	04
Wlxxxxx_A**	0–65535	**String**	Schreibgeschützt	04
Wlxxxxx_X<1, 2, 3>***	0–65535 0–65534	**Word**, Short, BCD Float, DWord, Long, LBCD	Schreibgeschützt	04
Dlxxxxx_S	0–65534	**Long**	Schreibgeschützt	04
Dlxxxxx_B	0–65534	**LBCD**	Schreibgeschützt	04
Dlxxxxx_X<1, 2, 3>***	0–65534	**DWord**	Schreibgeschützt	04
Flxxxxx_X<1, 2, 3>***	0–65534	**Float**	Schreibgeschützt	04
M_Wlxxxxx_n(H) Zeichenfolge mit Byte-Reihenfolge Hi-Lo (H optional)	xxxxx=0–65535 n ist die Zeichenfolgenlänge. Bereich liegt zwischen 1 und 120 Wörtern.	**String**	Schreibgeschützt	04
M_Wlxxxxx_nL Zeichenfolge mit Byte-Reihenfolge Lo-Hi	xxxxx=0–65535 n ist die Zeichenfolgenlänge. Bereich liegt zwischen 1 und 120 Wörtern.	**String**	Schreibgeschützt	04

*Weitere Informationen finden Sie unter Nullbasierte Bit-Adressierung unter **Einstellungen**.
**Die Länge der Zeichenfolge beträgt 2 Byte.
***Weitere Informationen finden Sie unter **Byte-Wechsel-Suffixe**.

Array-Unterstützung
Arrays werden für die internen Registeradressen unterstützt. Die Syntax zum Deklarieren eines Arrays lautet wie folgt:
Wlxxxxx_cols mit einer angenommenen Zeilenanzahl von 1.
Wlxxxxx_rows_cols.

Für Word-, Short- und BCD-Arrays darf die Basisadresse + (rows * cols) den Wert 65535 nicht überschreiten.
Für Float-, DWord-, Long- und Long BCD-Arrays darf die Basisadresse–+ (rows * cols * 2) den Wert 65534 nicht überschreiten.
Für alle Arrays darf die Gesamtanzahl der angeforderten Register die für dieses Gerät angegebene Blockgröße für interne Register nicht überschreiten.

Halteregister

Die Standard-Datentypen werden **fett** dargestellt.

Arrays werden für Halteregister-Speicherorte für alle Datentypen außer Boolean und String unterstützt.

Hinweis: Bei Servergeräten ist für schreibgeschützte Speicherorte das Lesen/Schreiben möglich.

Adresse	Bereich	Datentyp	Zugriff	Funktionscode
Wxxxxx	0–65535 0–65534 0–65532	**Word**, Short, BCD Float, DWord, Long, LBCD Double	Lesen/Schreiben	03, 06, 16
Wxxxxx.bb	xxxxx=0–65535 bb=0/1–15/16*	**Boolean**	Lesen/Schreiben	03, 06, 16, 22
Wxxxxx:Xbb	xxxxx=0–65535 bb=0/1–15/16*	**Boolean**	Lesen/Schreiben	03, 06, 16, 22
Dxxxxx	0–65534	**DWord**	Lesen/Schreiben	03, 06, 16
Fxxxxx	0–65534	**Float**	Lesen/Schreiben	03, 06, 16
Wxxxxx_S	0–65535	**Short**	Lesen/Schreiben	03, 06, 16
Wxxxxx_B	0–65535	**BCD**	Lesen/Schreiben	03, 06, 16
Wxxxxx_A**	0–65535	**String**	Schreibgeschützt	03,·16
Wxxxxx_X<1, 2, 3>***	0–65535 0–65534	**Word**, Short, BCD Float, DWord, Long, LBCD	Lesen/Schreiben	03, 06, 16
Dxxxxx_S	0–65534	**Long**	Lesen/Schreiben	03, 06, 16
Dxxxxx_B	0–65534	**LBCD**	Lesen/Schreiben	03, 06, 16
Dxxxxx_X<1, 2, 3>***	0–65534	**DWord**	Lesen/Schreiben	03, 06, 16
Fxxxxx_X<1, 2, 3>***	0–65534	**Float**	Lesen/Schreiben	03, 06, 16
M_Wxxxxx_n(H) Zeichenfolge mit Byte-Reihenfolge Hi-Lo (H optional)	xxxxx=0–65535 n ist die Zeichenfolgenlänge. Bereich liegt zwischen 1 und 120 Wörtern.	**String**	Lesen/Schreiben	03,·16
M_Wxxxxx_nL Zeichenfolge mit Byte-Reihenfolge Lo-Hi	xxxxx=0–65535 n ist die Zeichenfolgenlänge. Bereich liegt zwischen 1 und 120 Wörtern.	**String**	Lesen/Schreiben	03,·16

*Weitere Informationen finden Sie unter Nullbasierte Bit-Adressierung unter **Einstellungen**.
**Die Länge der Zeichenfolge beträgt 2 Byte.
***Weitere Informationen finden Sie unter **Byte-Wechsel-Suffixe**.

Array-Unterstützung
Arrays werden für die Halteregister-Adressen unterstützt. Die Syntax zum Deklarieren eines Arrays mit dezimaler Adressierung lautet wie folgt:
Wxxxxx_cols mit einer angenommenen Zeilenanzahl von 1.

Wxxxxx_rows_cols.

Für Word-, Short- und BCD-Arrays darf die Basisadresse + (rows * cols) den Wert 65535 nicht überschreiten. Für Float-, DWord-, Long- und Long BCD-Arrays darf die Basisadresse–+ (rows * cols * 2) den Wert 65534 nicht überschreiten.
Für alle Arrays darf die Gesamtanzahl der angeforderten Register die für dieses Gerät angegebene Blockgröße für Halteregister nicht überschreiten.

Zeichenfolgenunterstützung
Das Applicom-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 Zeichenfolgenlänge kann zwischen 1 und 120 Wörtern liegen. Weitere Informationen zum Ausführen eines Block-Lesevorgangs für Zeichenfolgen-Tags finden Sie unter **Blockgrößen**.

Hinweis: Die Zeichenfolgenlänge kann durch die maximale Größe der Schreibanforderung begrenzt werden, die das Gerät zulässt. Wenn im Server-Ereignisfenster die Fehlermeldung "In Adresse <Adresse> auf dem Gerät <Gerät> kann nicht geschrieben werden: Gerät hat mit Ausnahmecode 3 geantwortet." empfangen wird, unterstützt das Gerät die Zeichenfolgenlänge nicht. Um dies zu korrigieren, kürzen Sie die Zeichenfolge auf die unterstützte Länge.

Byte-Wechsel-Suffixe

Diese Suffixe werden verwendet, um die Byte zu wechseln, die Daten des Typs 16-Bit-Word, 32-Bit-DWord oder 32-Bit-Float formatieren. Der Byte-Wechsel wird angewendet, nachdem die Einstellungen für Modbus-Byte-Reihenfolge und Erstes Wort 'Low' auf Geräteebene angewendet wurden. Weitere Informationen finden Sie unter **Einstellungen**.

Byte-Wechsel-Suffixe können nur mit internen Registern und Halteregistern verwendet werden. Informationen zu den verschiedenen Wechseltypen, die vom Suffix und Datentyp des Elements abhängig sind, finden Sie in der folgenden Tabelle.

Suffix	16-Bit-Datentypen (Word, Short, BCD)	32-Bit-Datentypen (DWord, Long, LBCD, Float)
_X1	O1 O2 -> O2 O1 (Byte-Wechsel)	O1 O2 O3 O4 -> O4 O3 O2 O1 (Byte-Wechsel)
_X2	O1 O2 -> O2 O1 (Byte-Wechsel)	O1 O2 O3 O4 -> O3 O4 O1 O2 (Word-Wechsel)
_X3	O1 O2 -> O2 O1 (Byte-Wechsel)	O1 O2 O3 O4 -> O2 O1 O4 O3 (Byte in den Wörtern wechselnd)

TSX Quantum

Alle Funktionscodes werden als Dezimalzahlen angezeigt. *Weitere Informationen finden Sie unter **Beschreibung von Funktionscodes**.*

Ausgangs-Coils

Adresse	Bereich	Datentyp	Zugriff	Funktionscode
0xxxxx	1–65536	Boolean	Lesen/Schreiben	01, 05, 15

Array-Unterstützung
Arrays werden für die Ausgangs-Coil-Adressen unterstützt. Die Syntax zum Deklarieren eines Arrays lautet wie folgt:
0xxxxx_cols mit einer angenommenen Zeilenanzahl von 1.

0xxxxx_rows_cols.

Die Basisadresse + (rows * cols) darf den Wert 65536 nicht überschreiten. Die Gesamtanzahl der angeforderten Coils darf die für dieses Gerät angegebene Blockgröße für Ausgangs-Coils nicht überschreiten.

Eingangs-Coils

Adresse	Bereich	Datentyp	Zugriff	Funktionscode
1xxxxx	1–65536	Boolean	Schreibgeschützt	02

Array-Unterstützung
Arrays werden für die Eingangs-Coil-Adressen unterstützt. Die Syntax zum Deklarieren eines Arrays lautet wie folgt:
1xxxxx_cols mit einer angenommenen Zeilenanzahl von 1.
1xxxxx_rows_cols.

Die Basisadresse + (rows * cols) darf den Wert 65536 nicht überschreiten. Die Gesamtanzahl der angeforderten Coils darf die für dieses Gerät angegebene Blockgröße für Eingangs-Coils nicht überschreiten.

Interne Register

Die Standard-Datentypen werden **fett** dargestellt.

Arrays werden für interne Registerpositionen für alle Datentypen außer Boolean und String unterstützt.

Hinweis: Bei Servergeräten ist für schreibgeschützte Positionen das Lesen/Schreiben möglich.

Adresse	Bereich	Datentyp	Zugriff	Funktionscode
3xxxxx	1–65536 1–65535 1–65533	**Word**, Short, BCD Float, DWord, Long, LBCD Double	Schreibgeschützt	04
3xxxxx.bb	xxxxx=1–65536 bb=0/1–15/16*	**Boolean**	Schreibgeschützt	04
3xxxxx:Xbb	xxxxx=0–65535 bb=0/1–15/16*	**Boolean**	Schreibgeschützt	04
D3xxxxx	1–65535	**DWord**	Schreibgeschützt	04
F3xxxxx	1–65535	**Float**	Schreibgeschützt	04
3xxxxx_S	1–65536	**Short**	Schreibgeschützt	04
3xxxxx_B	1–65536	**BCD**	Schreibgeschützt	04
3xxxxx_A**	1–65536	**String**	Schreibgeschützt	04
3xxxxx_X<1, 2, 3>***	1–65536 1–65535	**Word**, Short, BCD Float, DWord, Long, LBCD	Schreibgeschützt	04
D3xxxxx_S	1–65535	**Long**	Schreibgeschützt	04
D3xxxxx_B	1–65535	**LBCD**	Schreibgeschützt	04
D3xxxxx_X<1, 2, 3>***	1–65535	**DWord**	Schreibgeschützt	04

Adresse	Bereich	Datentyp	Zugriff	Funktionscode
F3xxxxx_X<1, 2, 3>***	1–65535	**Float**	Schreibgeschützt	04
M_3xxxxx_n(H) Zeichenfolge mit Byte-Reihenfolge Hi-Lo (H optional)	xxxxx=1–65536 n ist die Zeichenfolgenlänge. Bereich liegt zwischen 1 und 120 Wörtern.	**String**	Schreibgeschützt	04
M_3xxxxx_nL Zeichenfolge mit Byte-Reihenfolge Lo-Hi	xxxxx=1–65536 n ist die Zeichenfolgenlänge. Bereich liegt zwischen 1 und 120 Wörtern.	**String**	Schreibgeschützt	04

*Weitere Informationen finden Sie unter Nullbasierte Bit-Adressierung unter **Einstellungen**.
**Die Länge der Zeichenfolge beträgt 2 Byte.
***Weitere Informationen finden Sie unter **Byte-Wechsel-Suffixe**.

Array-Unterstützung
Arrays werden für die internen Registeradressen unterstützt. Die Syntax zum Deklarieren eines Arrays lautet wie folgt:
3xxxxx_cols mit einer angenommenen Zeilenanzahl von 1.
3xxxxx_rows_cols.

Für Word-, Short- und BCD-Arrays darf die Basisadresse + (rows * cols) den Wert 65536 nicht überschreiten.
Für Float-, DWord-, Long- und Long BCD-Arrays darf die Basisadresse–+ (rows * cols * 2) den Wert 65535 nicht überschreiten.
Für alle Arrays darf die Gesamtanzahl der angeforderten Register die für dieses Gerät angegebene Blockgröße für interne Register nicht überschreiten.

Halteregister

Die Standard-Datentypen werden **fett** dargestellt.

Arrays werden für Halteregister-Speicherorte für alle Datentypen außer Boolean und String unterstützt.

Hinweis: Bei Servergeräten ist für schreibgeschützte Positionen das Lesen/Schreiben möglich.

Adresse	Bereich	Datentyp	Zugriff	Funktionscode
4xxxxx	1–65536 1–65535 1–65533	**Word**, Short, BCD Float, DWord, Long, LBCD Double	Lesen/Schreiben	03, 06, 16
4xxxxx.bb	xxxxx=1–65536 bb=0/1–15/16*	**Boolean**	Lesen/Schreiben	03, 06, 16, 22
4xxxxx:Xbb	xxxxx=0–65535 bb=0/1–15/16*	**Boolean**	Lesen/Schreiben	03, 06, 16, 22

Adresse	Bereich	Datentyp	Zugriff	Funktionscode
D4xxxxx	1–65535	**DWord**	Lesen/Schreiben	03, 06, 16
F4xxxxx	1–65535	**Float**	Lesen/Schreiben	03, 06, 16
4xxxxx_S	1–65536	**Short**	Lesen/Schreiben	03, 06, 16
4xxxxx_B	1–65536	**BCD**	Lesen/Schreiben	03, 06, 16
4xxxxx_A**	1–65536	**String**	Schreibgeschützt	03,·16
4xxxxx_X<1, 2, 3>***	1–65536 1–65535	**Word**, Short, BCD Float, DWord, Long, LBCD	Lesen/Schreiben	03, 06, 16
D4xxxxx_S	1–65535	**Long**	Lesen/Schreiben	03, 06, 16
D4xxxxx_B	1–65535	**LBCD**	Lesen/Schreiben	03, 06, 16
D4xxxxx_X<1, 2, 3>***	1–65535	**DWord**	Lesen/Schreiben	03, 06, 16
F4xxxxx_X<1, 2, 3>***	1–65535	**Float**	Lesen/Schreiben	03, 06, 16
M_4xxxxx_n(H) Zeichenfolge mit Byte-Reihenfolge Hi-Lo (H optional)	xxxxx=1–65536 n ist die Zeichenfolgenlänge. Bereich liegt zwischen 1 und 120 Wörtern.	**String**	Lesen/Schreiben	03,·16
M_4xxxxx_nL Zeichenfolge mit Byte-Reihenfolge Lo-Hi	xxxxx=1–65536 n ist die Zeichenfolgenlänge. Bereich liegt zwischen 1 und 120 Wörtern.	**String**	Lesen/Schreiben	03,·16

*Weitere Informationen finden Sie unter Nullbasierte Bit-Adressierung unter **Einstellungen**.
**Die Länge der Zeichenfolge beträgt 2 Byte.
***Weitere Informationen finden Sie unter **Byte-Wechsel-Suffixe**.

Array-Unterstützung
Arrays werden für die Halteregister-Adressen unterstützt. Die Syntax zum Deklarieren eines Arrays mit dezimaler Adressierung lautet wie folgt:

4xxxxx_cols mit einer angenommenen Zeilenanzahl von 1.
4xxxxx_rows_cols.

Für Word-, Short- und BCD-Arrays darf die Basisadresse + (rows * cols) den Wert 65536 nicht überschreiten.
Für Float-, DWord-, Long- und Long BCD-Arrays darf die Basisadresse–+ (rows * cols * 2) den Wert 65535 nicht überschreiten.
Für alle Arrays darf die Gesamtanzahl der angeforderten Register die für dieses Gerät angegebene Blockgröße für Halteregister nicht überschreiten.

Zeichenfolgenunterstützung

Das Applicom-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 Zeichenfolgenlänge kann zwischen 1 und 120 Wörtern liegen.
*Informationen zum Ausführen eines Block-Lesevorgangs für Zeichenfolgen-Tags finden Sie unter **Blockgrößen**.*

Hinweis: Die Zeichenfolgenlänge kann durch die maximale Größe der Schreibanforderung begrenzt werden, die das Gerät zulässt. Wenn im Server-Ereignisfenster die Fehlermeldung "In Adresse <Adresse> auf dem Gerät <Gerät> kann nicht geschrieben werden: Gerät hat mit Ausnahmecode 3 geantwortet." empfangen wird, unterstützt das Gerät die Zeichenfolgenlänge nicht. Um dies zu korrigieren, kürzen Sie die Zeichenfolge auf die unterstützte Länge.

Byte-Wechsel-Suffixe

Diese Suffixe werden verwendet, um die Byte zu wechseln, die Daten des Typs 16-Bit-Word, 32-Bit-DWord oder 32-Bit-Float formatieren. Der Byte-Wechsel wird angewendet, nachdem die Einstellungen für Modbus-Byte-Reihenfolge und Erstes Wort 'Low' auf Geräteebene angewendet wurden. Weitere Informationen finden Sie unter **Einstellungen**.

Byte-Wechsel-Suffixe können nur mit internen Registern und Halteregistern verwendet werden. Informationen zu den verschiedenen Wechseltypen, die vom Suffix und Datentyp des Elements abhängig sind, finden Sie in der folgenden Tabelle.

Suffix	16-Bit-Datentypen (Word, Short, BCD)	32-Bit-Datentypen (DWord, Long, LBCD, Float)
_X1	O1 O2 -> O2 O1 (Byte-Wechsel)	O1 O2 O3 O4 -> O4 O3 O2 O1 (Byte-Wechsel)
_X2	O1 O2 -> O2 O1 (Byte-Wechsel)	O1 O2 O3 O4 -> O3 O4 O1 O2 (Word-Wechsel)
_X3	O1 O2 -> O2 O1 (Byte-Wechsel)	O1 O2 O3 O4 -> O2 O1 O4 O3 (Byte in den Wörtern wechselnd)

TSX Premium

Alle Funktionscodes werden als Dezimalzahlen angezeigt. Weitere Informationen finden Sie unter **Beschreibung von Funktionscodes**.

Ausgangs-Coils

Adresse	Bereich	Datentyp	Zugriff	Funktionscode
%MXxxxxx	0–65535	Boolean	Lesen/Schreiben	01, 05, 15
%Mxxxxx	0–65535	Boolean	Lesen/Schreiben	01, 05, 15

Array-Unterstützung

Arrays werden für die Ausgangs-Coil-Adressen unterstützt. Die Syntax zum Deklarieren eines Arrays lautet wie folgt:
%MXxxxxx_cols mit einer angenommenen Zeilenanzahl von 1.
%MXxxxxx_rows_cols.

Die Basisadresse + (rows * cols) darf den Wert 65535 nicht überschreiten. Die Gesamtanzahl der angeforderten Coils darf die für dieses Gerät angegebene Blockgröße für Ausgangs-Coils nicht überschreiten.

Halteregister

Die Standard-Datentypen werden **fett** dargestellt.

Arrays werden für Halteregister-Speicherorte für alle Datentypen außer Boolean und String unterstützt.

Hinweis: Bei Servergeräten ist für schreibgeschützte Speicherorte das Lesen/Schreiben möglich.

Adresse	Bereich	Datentyp	Zugriff	Funktionscode
%MWxxxxx	0–65535 0–65534 0–65532	**Word**, Short, BCD Float, DWord, Long, LBCD Double	Lesen/Schreiben	03, 06, 16
%MWxxxxx.bb	xxxxx=0–65535 bb=0/1–15/16*	**Boolean**	Lesen/Schreiben	03, 06, 16, 22
%MWxxxxx:Xbb	xxxxx=0–65535 bb=0/1–15/16*	**Boolean**	Lesen/Schreiben	03, 06, 16, 22
%DWxxxxx oder %MDxxxxx	0–65534	**DWord**	Lesen/Schreiben	03, 06, 16
%FWxxxxx oder %MFxxxxx	0–65534	**Float**	Lesen/Schreiben	03, 06, 16
%MWxxxxx_S	0–65535	**Short**	Lesen/Schreiben	03, 06, 16
%MWxxxxx_B	0–65535	**BCD**	Lesen/Schreiben	03, 06, 16
%MWxxxxx_A**	0–65535	**String**	Schreibgeschützt	03,·16
%MWxxxxx_X<1, 2, 3>***	0–65535 0–65534	**Word**, Short, BCD Float, DWord, Long, LBCD	Lesen/Schreiben	03, 06, 16
%DWxxxxx_S	0–65534	**Long**	Lesen/Schreiben	03, 06, 16
%DWxxxxx_B	0–65534	**LBCD**	Lesen/Schreiben	03, 06, 16
%DWxxxxx_X<1, 2, 3>*** oder %MDxxxxx_X<1, 2, 3> ***	0–65534	**DWord**	Lesen/Schreiben	03, 06, 16
%FWxxxxx_X<1, 2, 3>*** oder %MFxxxxx_X<1, 2, 3> ***	0–65534	**Float**	Lesen/Schreiben	03, 06, 16
M_%MWxxxxx_n(H) Zeichenfolge mit Hi-Lo Byte-Reihenfolge (H optional)	xxxxx=0–65535 n ist die Zeichenfolgenlänge. Bereich liegt zwischen 1 und 120 Wörtern.	**String**	Lesen/Schreiben	03,·16
M_%MWxxxxx_nL Zeichenfolge mit Byte-Reihenfolge	xxxxx=0–65535 n ist die Zeichenfolgenlänge.	**String**	Lesen/Schreiben	03,·16

Adresse	Bereich	Datentyp	Zugriff	Funktionscode
Lo-Hi	Bereich liegt zwischen 1 und 120 Wörtern.			

*Weitere Informationen finden Sie unter Nullbasierte Bit-Adressierung unter **Einstellungen**.
**Die Länge der Zeichenfolge beträgt 2 Byte.
***Weitere Informationen finden Sie unter **Byte-Wechsel-Suffixe**.

Array-Unterstützung
Arrays werden für die Halteregister-Adressen unterstützt. Die Syntax zum Deklarieren eines Arrays mit dezimaler Adressierung lautet wie folgt:
%MXxxxxx_cols mit einer angenommenen Zeilenanzahl von 1.
%MWxxxxx_rows_cols.

Für Word-, Short- und BCD-Arrays darf die Basisadresse + (rows * cols) den Wert 65535 nicht überschreiten.
Für Float-, DWord-, Long- und Long BCD-Arrays darf die Basisadresse–+ (rows * cols * 2) den Wert 65534 nicht überschreiten.
Für alle Arrays darf die Gesamtanzahl der angeforderten Register die für dieses Gerät angegebene Blockgröße für Halteregister nicht überschreiten.

Zeichenfolgenunterstützung
Das Applicom-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 Zeichenfolgenlänge kann zwischen 1 und 120 Wörtern liegen. Weitere Informationen zum Ausführen eines Block-Lesevorgangs für Zeichenfolgen-Tags finden Sie unter **Blockgrößen**.

Hinweis: Die Zeichenfolgenlänge kann durch die maximale Größe der Schreibanforderung begrenzt werden, die das Gerät zulässt. Wenn im Server-Ereignisfenster die Fehlermeldung "In Adresse <Adresse> auf dem Gerät <Gerät> kann nicht geschrieben werden: Gerät hat mit Ausnahmecode 3 geantwortet." empfangen wird, unterstützt das Gerät die Zeichenfolgenlänge nicht. Um dies zu korrigieren, kürzen Sie die Zeichenfolge auf die unterstützte Länge.

Byte-Wechsel-Suffixe

Diese Suffixe werden verwendet, um die Byte zu wechseln, die Daten des Typs 16-Bit-Word, 32-Bit-DWord oder 32-Bit-Float formatieren. Der Byte-Wechsel wird angewendet, nachdem die Einstellungen für Modbus-Byte-Reihenfolge und Erstes Wort 'Low' auf Geräteebene angewendet wurden. Weitere Informationen finden Sie unter **Einstellungen**.

Byte-Wechsel-Suffixe können nur mit internen Registern und Halteregistern verwendet werden. Informationen zu den verschiedenen Wechseltypen, die vom Suffix und Datentyp des Elements abhängig sind, finden Sie in der folgenden Tabelle.

Suffix	16-Bit-Datentypen (Word, Short, BCD)	32-Bit-Datentypen (DWord, Long, LBCD, Float)
_X1	O1 O2 -> O2 O1 (Byte-Wechsel)	O1 O2 O3 O4 -> O4 O3 O2 O1 (Byte-Wechsel)
_X2	O1 O2 -> O2 O1 (Byte-Wechsel)	O1 O2 O3 O4 -> O3 O4 O1 O2 (Word-Wechsel)
_X3	O1 O2 -> O2 O1 (Byte-Wechsel)	O1 O2 O3 O4 -> O2 O1 O4 O3 (Byte in den Wörtern wechselnd)

CEG-Adressierung

Die Adressierung für das CEG-Gerätemodell ist mit der für das Modbus-Gerätemodell identisch.

*Weitere Informationen finden Sie unter **Modbus-Adressierung**.*

Fluenta-Adressierung

Die Standard-Datentypen werden **fett** dargestellt.

Adresse	Bereich	Datentyp	Zugriff
System	400000–409999	**Float**, Double	Lesen/Schreiben
Ausgabe	410000–410999 420000–420999 430000–430999	**Float**, Double	Schreibgeschützt
Benutzer	411000–411999 421000–421999 431000–431999	**Float**, Double	Lesen/Schreiben
Dienst	412000–412999 422000–422999 432000–432999	**Float**, Double	Lesen/Schreiben
Akkumulation	413000–413999 423000–423999 433000–433999	**Float**, Double	Schreibgeschützt

Instromet-Adressierung

Die Standard-Datentypen werden **fett** dargestellt.

Adresse	Bereich	Datentyp	Zugriff
Kurze Ganzzahlen	400000–400199	**Word**, Short	Schreibgeschützt
Lange Ganzzahlen	400200–400399	**DWord**, Long	Schreibgeschützt
Gleitkommazahlen	400400–400599	**Float**	Schreibgeschützt

Postfach-Adressierung

Die Standard-Datentypen werden **fett** dargestellt.

Dezimale Adressierung

Adresse	Bereich	Datentyp	Zugriff
4xxxxx	1–65536	Word, Short, BCD	Lesen/Schreiben
4xxxxx.bb	xxxxx=1–65536 bb=0–15	**Boolean**	Lesen/Schreiben
4xxxxx	1–65535	Float, DWord, Long, LBCD	Lesen/Schreiben

Hexadezimale Adressierung

Adresse	Bereich	Datentyp	Zugriff
H4yyyyy	1–10000	Word, Short, BCD	Lesen/Schreiben
H4yyyyy.c	yyyyy=1–10000 c=0–F	**Boolean**	Lesen/Schreiben
H4yyyy	1–FFFF	Float, DWord, Long, LBCD	Lesen/Schreiben

Hinweis: Modbus-Postfach unterstützt nicht Funktionscode 22 (0x16). Nur 0x10 (Holding Reg Write Multiple) und 0x6 (Holding Reg Write Single) werden unterstützt. Es ist möglich, in ein einzelnes Bit zu schreiben, indem Sie die Option **Halteregister-Bit-Schreibvorgänge** in den Geräteeigenschaften auf der Registerkarte "Einstellungen" deaktivieren. Dadurch wird die Verwendung der Sequenz Lesen/Bearbeiten/Schreiben erzwungen, anstatt direkt in das Bit zu schreiben. Damit dies funktioniert, muss nur die Einstellung des Client-Modbus-Geräts (nicht des Postfachs) geändert werden.

Arrays

Arrays werden auch für die Halteregister-Adressen unterstützt. Die Syntax zum Deklarieren eines Arrays mit dezimaler Adressierung lautet wie folgt:

4xxxx[cols] mit einer angenommenen Zeilenanzahl von 1.
4xxxx[rows][cols].

Für Word-, Short- und BCD-Arrays darf die Basisadresse + (rows * cols) den Wert 65536 nicht überschreiten.

Für Float-, DWord-, Long- und Long BCD-Arrays darf die Basisadresse–+ (rows * cols * 2) den Wert 65535 nicht überschreiten.

Für alle Arrays darf die Gesamtanzahl der angeforderten Register die für dieses Gerät angegebene Blockgröße für Halteregister nicht überschreiten.

Modbus-Adressierung

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

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

Primäre Tabelle	Beschreibung
3	Interne Register
4	Halteregister

Modbus-Adressierung im Dezimalformat

Die Funktionscodes werden als Dezimalzahlen angezeigt. Weitere Informationen finden Sie unter **Beschreibung von Funktionscodes**.

Adresstyp	Bereich	Datentyp	Zugriff*	Funktionscodes
Ausgangs-Coils	000001–065536	Boolean	Lesen/Schreiben	01, 05, 15
Eingangs-Coils	100001–165536	Boolean	Schreibgeschützt	02
Interne Register	300001–365536 300001–365535 300001–365533 xxxxx=1–65536 bb=0/1–15/16** 300001.2H–365536.240H*** 300001.2L–365536.240L***	**Word**, Short, BCD Float, DWord, Long, LBCD Double **Boolean** String String	Schreibgeschützt Schreibgeschützt Schreibgeschützt Schreibgeschützt Schreibgeschützt Schreibgeschützt	04 04 04 04 04 04
Halteregister	400001–465536 400001–465535 400001–465533 xxxxx=1–65536 bb=0/1–15/16* 400001.2H–465536.240H*** 400001.2L–465536.240L***	**Word**, Short, BCD Float, DWord, Long, LBCD Double **Boolean** **Zeichenfolge** **Zeichenfolge**	Lesen/Schreiben Lesen/Schreiben Lesen/Schreiben Lesen/Schreiben Lesen/Schreiben Lesen/Schreiben	03, 06, 16 03, 06, 16 03, 06, 16 03, 06, 16, 22 03,·16 03,·16

*Bei Servergeräten ist für schreibgeschützte Speicherorte das Lesen/Schreiben möglich.
***Weitere Informationen finden Sie unter Nullbasierte Adressierung unter **Einstellungen**.*
***.Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte.

Modbus-Adressierung im Hexadezimalformat

Adresstyp	Bereich	Datentyp	Zugriff*
Ausgangs-Coils	H000001–H010000	Boolean	Lesen/Schreiben
Eingangs-Coils	H100001–H110000	Boolean	Schreibgeschützt
Interne Register	H300001–H310000 H300001–H30FFFF H300001–H30FFFD yyyyy=1–10000 cc=0/1–F/10 H300001.2H–	**Word**, Short, BCD Float, DWord, Long, LBCD Double **Boolean** String	Schreibgeschützt Schreibgeschützt Schreibgeschützt Schreibgeschützt Schreibgeschützt Schreibgeschützt

Adresstyp	Bereich	Datentyp	Zugriff*
	H3FFFF.240H** H300001.2L–H3FFFF.240L**	String	
Halteregister	H400001–H410000 H400001–H40FFFF H400001–H40FFFD yyyyy=1–10000 cc=0/1–F/10 H400001.2H–H4FFFF.240H H400001.2L–H4FFFF.240L	**Word**, Short, BCD Float, DWord, Long, LBCD Double **Boolean** **Zeichenfolge** **Zeichenfolge**	Lesen/Schreiben Lesen/Schreiben Lesen/Schreiben Lesen/Schreiben Lesen/Schreiben Lesen/Schreiben

*Bei Servergeräten ist für schreibgeschützte Positionen das Lesen/Schreiben möglich.
**.Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte.

Gepackte Coils

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. Sie ist nicht für Geräte verfügbar, die für den Zugriff auf die unaufgeforderte Speicherzuordnung konfiguriert sind oder sich im Postfach-Modus befinden. Die Dezimalsyntax lautet 0*xxxxx#nn*, wobei Folgendes gilt:
*Bei Servergeräten ist für schreibgeschützte Positionen das Lesen/Schreiben möglich.
**.Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte.

Gepackte Coils

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. Sie ist nicht für Geräte verfügbar, die für den Zugriff auf die unaufgeforderte Speicherzuordnung konfiguriert sind oder sich im Postfach-Modus befinden. Die Dezimalsyntax lautet 0*xxxxx#nn*, wobei Folgendes gilt:

- *xxxxx* steht für die Adresse der ersten Coil (Bereich 000001–065521).
- *nn* steht für die Anzahl der Coils, die in einen analogen Wert gepackt werden (Bereich 01–16).

Die Hexadezimalsyntax lautet H0*yyyyy#nn*, wobei Folgendes gilt:

- *yyyyy* steht für die Adresse der ersten Coil (Bereich H000001–H000FFF1).
- *nn* steht für die Anzahl der Coils, die in einen analogen Wert gepackt werden (Bereich 01–16).

Hinweise:

1. Der einzige gültige Datentyp ist Word. Ausgangs-Coils haben Lese-/Schreibzugriff, wohingegen Eingangs-Coils schreibgeschützten Zugriff haben. In dezimaler Adressierung unterstützen Ausgangs-Coils die Funktionscodes 01 und 15, wohingegen Eingangs-Coils Funktionscode 02 unterstützen.
2. Die Bit-Reihenfolge ist so angegeben, dass die Startadresse das niedrigstwertige Bit (LSB, Least Significant Bit) des analogen Werts ist.

Lesegeschützter Zugriff

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 "Schreibgeschützt" festgelegt werden, führt dies dazu, dass Schreibvorgänge in diesen Tags fehlschlagen und der Client immer 0 bzw. NULL für numerische bzw. Zeichenfolgenwerte empfängt.

Postfach-Modus

Im Postfach-Modus werden nur Halteregister unterstützt. Bei Lesevorgängen von einem Client werden die Daten lokal aus einem Cache gelesen, nicht aus einem physischen Gerät. Bei Schreibvorgängen von einem Client werden die Daten sowohl in den lokalen Cache als auch in das physische Gerät geschrieben, wie durch den Routing-Pfad der Geräte-ID bestimmt.
Hinweis: Der Datentyp Double wird nicht unterstützt.

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. Durch das Anhängen von "H" oder "L" an die Adresse wird die Byte-Reihenfolge angegeben.
Weitere Informationen zum Ausführen von Block-Lesevorgängen für Zeichenfolgen-Tags des Modbus-Modells finden Sie unter ***Blockgrößen****.*

Beispiele

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

Array-Unterstützung

Arrays werden für interne und Halteregister-Speicherorte unterstützt (einschließlich aller Datentypen außer Boolean und String) sowie für Eingangs- und Ausgangs-Coils (Boolean-Datentypen). Es gibt zwei Möglichkeiten, ein Array zu adressieren. Die folgenden Beispiele gelten für Halteregister:
4*xxxx* [Zeilen] [Spalten]
4*xxxx* [cols] mit einer angenommenen Zeilenanzahl von 1.

Für Word-, Short- und BCD-Arrays darf die Basisadresse + (rows * cols) den Wert 65536 nicht überschreiten. Für Float-, DWord-, Long- und Long BCD-Arrays darf die Basisadresse–+ (rows * cols * 2) den Wert 65535 nicht überschreiten. Für alle Arrays darf die Gesamtanzahl der angeforderten Register die für dieses Gerät angegebene Blockgröße für Halteregister nicht überschreiten.

Roxar-Adressierung

Die Standard-Datentypen werden **fett** dargestellt.

Adresse	Bereich	Datentyp	Zugriff
Kurze Ganzzahlen	403000–403999	**Word**, Short	Lesen/Schreiben
Gleitkommazahlen	407000–407999	**Float**	Lesen/Schreiben
Gleitkommazahlen	409000–409999	**Float**	Schreibgeschützt

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 Nach-

Element	Datentyp	Zugriff	Beschreibung
			richten ("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.
_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).

* Die Statistik _RespBadChecksum ist nicht implementiert. Datenpaket-Prüfsummen werden vom TCP-Protokoll verarbeitet.

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.

Element	Datentyp	Zugriff	Beschreibung
_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.
_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).

* Die Statistik _RespBadChecksum ist nicht implementiert. Datenpaket-Prüfsummen werden vom TCP-Protokoll verarbeitet.

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

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.

Fehler beim Starten der Winsock-Kommunikation.

Fehlertyp:

Fehler

Unangeforderte Kommunikation konnte nicht gestartet werden.

Fehlertyp:

Fehler

Mögliche Ursache:

Der Treiber konnte keinen zu überwachenden Socket für die unangeforderte Kommunikation erstellen.

Mögliche Lösung:

Vergewissern Sie sich, dass der auf Kanalebene angegebene Port nicht von einer anderen Anwendung im System verwendet wird.

Hinweis:

Für diesen Treiber werden die Begriffe 'Modbus-Server' und 'unangefordert' synonym verwendet.

Unangeforderter Postfach-Zugriff für nicht definiertes Gerät. Socket wird geschlossen. | IP-Adresse = '<Adresse>'.

Fehlertyp:

Fehler

Mögliche Ursache:

1. Ein Gerät mit der angegebenen IP-Adresse hat versucht, eine Postfach-Nachricht an den Server zu senden. Die Nachricht wurde nicht validiert, da kein Gerät mit dieser IP-Adresse im Postfach-Projekt konfiguriert ist.
2. Ein Gerät mit der angegebenen IP-Adresse hat versucht, eine Postfach-Nachricht an den Server zu senden. Die Nachricht wurde nicht validiert; der Grund dafür ist, dass trotz eines konfigurierten Geräts keine Clients Datenabfragen von dem Gerät durchführen.

Mögliche Lösung:

Damit der Server Postfach-Nachrichten akzeptiert, muss die entsprechende Geräte-IP im Projekt konfiguriert sein. Mindestens ein Datenelement vom Gerät muss durch einen Client angefordert werden.

Unangeforderte, nicht unterstützte Anfrage erhalten. | IP-Adresse = '<Adresse>'.

Fehlertyp:

Fehler

Mögliche Ursache:

Eine unaufgeforderte Anfrage wurde von der angegebenen Geräte-IP erhalten. Das Format der Anfrage war ungültig und nicht in der Modbus-Spezifikation enthalten.

Mögliche Lösung:

Vergewissern Sie sich, dass die Geräte, die zum Senden von Postfach-Daten konfiguriert sind, gültige Anfragen senden.

Fehler bei unangeforderter Postfach-Speicherzuordnung. | IP-Adresse = '<Adresse>'.

Fehlertyp:

Fehler

Mögliche Ursache:

1. Ein Gerät mit der angegebenen IP-Adresse hat versucht, eine Postfach-Nachricht an den Server zu senden. Die Nachricht wurde nicht validiert, da kein Gerät mit dieser IP-Adresse im Postfach-Projekt konfiguriert ist.
2. Ein Gerät mit der angegebenen IP-Adresse hat versucht, eine Postfach-Nachricht an den Server zu senden. Die Nachricht wurde nicht validiert; der Grund dafür ist, dass trotz eines konfigurierten Geräts keine Clients Datenabfragen von dem Gerät durchführen.

Mögliche Lösung:

Damit der Server Postfach-Nachrichten akzeptiert, muss die entsprechende Geräte-IP im Projekt konfiguriert sein. Mindestens ein Datenelement vom Gerät muss durch einen Client angefordert werden.

Socket-Verbindung konnte nicht erstellt werden.

Fehlertyp:

Fehler

Mögliche Ursache:

Der Server konnte keine TCP/IP-Socket-Verbindung zum angegebenen Gerät herstellen, wird dies jedoch weiterhin versuchen.

Mögliche Lösung:

1. Vergewissern Sie sich, dass das Gerät online ist.
2. Vergewissern Sie sich, dass die Geräte-IP im Subnetz der IP liegt, an die der Server gebunden ist. Überprüfen Sie, dass ein gültiges Gateway verfügbar ist, das eine Verbindung zum anderen Netzwerk zulässt.

Fehler beim Öffnen der Datei für Tag-Datenbank-Import. | Betriebssystemfehler = '<Fehler>'.

Fehlertyp:

Fehler

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:

Überprüfen Sie die Größe des Speichers auf dem Gerät, und definieren Sie die Array-Länge entsprechend um.

Ungültige Adresse im Block. | Blockbereich = <Adresse> bis <Adresse>.

Fehlertyp:

Fehler

Mögliche Ursache:

Der Treiber hat versucht, eine Position in einer SPS zu lesen, die nicht vorhanden ist und möglicherweise außerhalb des zulässigen Bereichs liegt. Wenn Sie beispielsweise in einer SPS, die nur über die Halteregister 40001 bis 41400 verfügt, die Adresse 41405 anfordern, wird dieser Fehler generiert. Nachdem dieser Fehler generiert wurde, fordert der Treiber diesen Datenblock von der SPS nicht erneut an. Alle anderen von diesem gleichen Block angeforderten Adressen werden als ungültig betrachtet.

Mögliche Lösung:

Aktualisieren Sie die Client-Anwendung, um Adressen im Bereich des Geräts anzufordern.

Siehe auch:

Fehlerbehandlung

Auflösen des Hosts fehlgeschlagen. | Host-Name = '<Name>'.

Fehlertyp:

Fehler

Mögliche Ursache:

Das Gerät wurde für die Verwendung eines DNS-Host-Namens und nicht einer IP-Adresse konfiguriert. Der Host-Name kann vom Server nicht in eine IP-Adresse aufgelöst werden.

Mögliche Lösung:

Vergewissern Sie sich, dass das Gerät online und bei der Domäne registriert ist.

Angegebene Blockgröße für Ausgangs-Coils überschreitet maximale Blockgröße. | Angegebene Blockgröße = <Anzahl> (Coils), Maximale Blockgröße = <Anzahl> (Coils).

Fehlertyp:

Fehler

Angegebene Blockgröße für Eingangs-Coils überschreitet maximale Blockgröße. | Angegebene Blockgröße = <Anzahl> (Coils), Maximale Blockgröße = <Anzahl> (Coils).

Fehlertyp:

Fehler

Angegebene Blockgröße für interne Register überschreitet maximale Blockgröße. | Angegebene Blockgröße = <Anzahl> (Register), Maximale Blockgröße = <Anzahl> (Register).

Fehlertyp:

Fehler

Angegebene Blockgröße für Halteregister überschreitet maximale Blockgröße. | Angegebene Blockgröße = <Anzahl> (Register), Maximale Blockgröße = <Anzahl> (Register).

Fehlertyp:

Fehler

Blockanfrage hat mit Ausnahme geantwortet. | Blockbereich = <Adresse> bis <Adresse>, Ausnahme = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

Das Gerät hat einen Ausnahmecode zurückgegeben.

Mögliche Lösung:

Weitere Informationen erhalten Sie in der Dokumentation zu Ausnahmecodes.

Siehe auch:

Modbus-Ausnahmecodes

Blockanfrage hat mit Ausnahme geantwortet. | Blockbereich = <Adresse> bis <Adresse>, Funktionscode = <Code>, Ausnahme = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

Das Gerät hat einen Ausnahmecode zurückgegeben.

Mögliche Lösung:

Weitere Informationen erhalten Sie in der Dokumentation zu Ausnahmecodes.

Siehe auch:

Modbus-Ausnahmecodes

Ungültige Blocklänge erhalten. | Blockbereich = <Start> bis <Ende>.

Fehlertyp:

Warnung

Mögliche Ursache:

Der Treiber hat versucht, einen Speicherblock in der SPS zu lesen. Die SPS hat ohne Fehler geantwortet, hat dem Treiber jedoch nicht die angeforderte Blockgröße an Daten bereitgestellt.

Mögliche Lösung:

Vergewissern Sie sich, dass der Speicherbereich für die SPS vorhanden ist.

Tag-Import wegen zu wenig Speicherressourcen fehlgeschlagen.

Fehlertyp:

Warnung

Mögliche Ursache:

Der Treiber konnte 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 = <Feld>.

Fehlertyp:

Warnung

Mögliche Ursache:

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

Mögliche Lösung:

Ändern Sie in der Variablenimportdatei das Feld, das den Fehler verursacht, sofern dies möglich ist.

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 in der Variablenimportdatei angegebene Datentyp gehört nicht zu den Typen, die von diesem Treiber unterstützt werden.

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

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

Fehlertyp:

Warnung

Mögliche Ursache:

Das Gerät hat einen Ausnahmecode zurückgegeben.

Mögliche Lösung:

Weitere Informationen erhalten Sie in der Dokumentation zu Ausnahmecodes.

Siehe auch:

Modbus-Ausnahmecodes

Ethernet-Manager wurde gestartet.

Fehlertyp:

Informationen

Ethernet-Manager wurde angehalten.

Fehlertyp:

Informationen

Tag-Datenbank wird importiert. | Quelldatei = '<Dateiname>'.

Fehlertyp:

Informationen

Eine Client-Anwendung hat die CEG-Erweiterung über das System-Tag _CEGExtension geändert. | Erweiterung = '<Erweiterung>'.

Fehlertyp:

Informationen

Mögliche Ursache:

Eine mit dem Server verbundene Client-Anwendung hat die CEG-Erweiterung auf dem angegebenen Gerät in 0 für Modbus oder 1 für CEG geändert.

Mögliche Lösung:

Diese Geräteeigenschaft gilt nur für CEG-Gerätemodelle. Änderungen wirken sich nicht auf andere Modelle aus. Damit die Client-Anwendung diese Eigenschaft nicht ändert, deaktivieren Sie die Client-Funktion zum Schreiben in Tags auf Systemebene über die OPC DA-Einstellungen.

Unangeforderte Kommunikation wird gestartet. | Protokoll = '<Name>', Port = <Nummer>.

Fehlertyp:
Informationen

Speicher für Modbus-Server-Gerät erstellt. | Modbus-Server-Geräte-ID = <Gerät>.

Fehlertyp:
Informationen

Alle Kanäle abonnieren ein virtuelles Netzwerk oder alle Geräte hören auf Remote-Adressen, wodurch unangeforderte Kommunikation beendet wird.

Fehlertyp:
Informationen

Kanal ist in einem virtuellen Netzwerk, alle Geräte verwenden jetzt wieder nur einen Socket pro Gerät.

Fehlertyp:
Informationen

Geräte-ID kann nicht von 'Modbus-Client-Modus' in 'Server-Modus' geändert werden, wenn ein Client verbunden ist.

Fehlertyp:
Informationen

Geräte-ID kann nicht von 'Modbus-Server-Modus' in 'Client-Modus' geändert werden, wenn ein Client verbunden ist.

Fehlertyp:
Informationen

Modbus-Server-Modus ist nicht zulässig, wenn sich der Kanal in einem virtuellen Netzwerk befindet. Die Geräte-ID darf keinen Loopback oder keine lokale IP-Adresse enthalten.

Fehlertyp:
Informationen

Postfach-Modell nicht zulässig, wenn sich der Kanal in einem virtuellen Netzwerk befindet.

Fehlertyp:
Informationen

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.

Modbus Ethernet-Kanaleigenschaften

Nachfolgend finden Sie eine vollständige Liste aller Eigenschaften für die Modbus Ethernet-Kanalebene.


```
{ "common.ALLTYPES_NAME": "MyChannel", "common.ALLTYPES_DESCRIPTION": "", "ser-
vermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Modbus TCP/IP Ethernet", "servermain.CHANNEL_
DIAGNOSTICS_CAPTURE": false, "servermain.CHANNEL_UNIQUE_ID": 721923342, "ser-
vermain.CHANNEL_ETHERNET_COMMUNICATIONS_NETWORK_ADAPTER_STRING": "", "ser-
vermain.CHANNEL_WRITE_OPTIMIZATIONS_METHOD": 2, "servermain.CHANNEL_WRITE_
OPTIMIZATIONS_DUTY_CYCLE": 10, "servermain.CHANNEL_NON_NORMALIZED_FLOATING_POINT_
HANDLING": 0, "servermain.CHANNEL_COMMUNICATIONS_SERIALIZATION_VIRTUAL_NETWORK": 0,
"servermain.CHANNEL_COMMUNICATIONS_SERIALIZATION_TRANSACTIONS_PER_CYCLE": 1, "ser-
vermain.CHANNEL_COMMUNICATIONS_SERIALIZATION_NETWORK_MODE": 0, "modbus_ether-
net.CHANNEL_USE_ONE_OR_MORE_SOCKETS_PER_DEVICE": 1, "modbus_ethernet.CHANNEL_MAXIMUM_
SOCKETS_PER_DEVICE": 1 }
```


Modbus Ethernet-Geräteeigenschaften

Nachfolgend finden Sie eine vollständige Liste aller Eigenschaften für die Modbus Ethernet-Geräteebene.


```
{ "common.ALLTYPES_NAME": "MyDevice", "common.ALLTYPES_DESCRIPTION": "", "ser-
vermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Modbus TCP/IP Ethernet", "servermain.DEVICE_
MODEL": 0, "servermain.DEVICE_UNIQUE_ID": 70949968, "servermain.DEVICE_CHANNEL_
ASSIGNMENT": "MyChannel", "servermain.DEVICE_ID_FORMAT": 0, "servermain.DEVICE_ID_
STRING": "<0.0.0.0>.0", "servermain.DEVICE_ID_HEXADECIMAL": 0, "servermain.DEVICE_ID_
DECIMAL": 0, "servermain.DEVICE_ID_OCTAL": 0, "servermain.DEVICE_DATA_COLLECTION":
true, "servermain.DEVICE_SIMULATED": false, "servermain.DEVICE_SCAN_MODE": 0, "ser-
vermain.DEVICE_SCAN_MODE_RATE_MS": 1000, "servermain.DEVICE_SCAN_MODE_PROVIDE_
INITIAL_UPDATES_FROM_CACHE": false, "servermain.DEVICE_CONNECTION_TIMEOUT_SECONDS":
3, "servermain.DEVICE_REQUEST_TIMEOUT_MILLISECONDS": 1000, "servermain.DEVICE_RETRY_
ATTEMPTS": 3, "servermain.DEVICE_INTER_REQUEST_DELAY_MILLISECONDS": 0, "ser-
vermain.DEVICE_AUTO_DEMOTION_ENABLE_ON_COMMUNICATIONS_FAILURES": false, "ser-
vermain.DEVICE_AUTO_DEMOTION_DEMOTE_AFTER_SUCCESSIVE_TIMEOUTS": 3,
"servermain.DEVICE_AUTO_DEMOTION_PERIOD_MS": 10000, "servermain.DEVICE_AUTO_DEMOTION_
DISCARD_WRITES": false, "servermain.DEVICE_TAG_GENERATION_ON_STARTUP": 0, "ser-
vermain.DEVICE_TAG_GENERATION_DUPLICATE_HANDLING": 0, "servermain.DEVICE_TAG_
GENERATION_GROUP": "", "servermain.DEVICE_TAG_GENERATION_ALLOW_SUB_GROUPS": true,
"modbus_ethernet.DEVICE_VARIABLE_IMPORT_FILE": "normal.txt", "modbus_ethernet.DEVICE_
VARIABLE_IMPORT_INCLUDE_DESCRIPTIONS": 1, "modbus_ethernet.DEVICE_DEACTIVATE_TAGS_ON_
ILLEGAL_ADDRESS": 1, "modbus_ethernet.DEVICE_SUB_MODEL": 1, "modbus_ethernet.DEVICE_
ETHERNET_PORT_NUMBER": 502, "modbus_ethernet.DEVICE_ETHERNET_IP_PROTOCOL": 1, "mod-
bus_ethernet.DEVICE_ETHERNET_CLOSE_TCP_SOCKET_ON_TIMEOUT": true, "modbus_ether-
net.DEVICE_ZERO_BASED_ADDRESSING": true, "modbus_ethernet.DEVICE_ZERO_BASED_BIT_
ADDRESSING": true, "modbus_ethernet.DEVICE_HOLDING_REGISTER_BIT_MASK_WRITES": true,
"modbus_ethernet.DEVICE_MODBUS_FUNCTION_06": true, "modbus_ethernet.DEVICE_MODBUS_
FUNCTION_05": true, "modbus_ethernet.DEVICE_MODBUS_BYTE_ORDER": true, "modbus_ether-
net.DEVICE_FIRST_WORD_LOW": true, "modbus_ethernet.DEVICE_FIRST_DWORD_LOW": true,
"modbus_ethernet.DEVICE_MODICON_BIT_ORDER": false, "modbus_ethernet.DEVICE_TREAT_
LONGS_AS_DOUBLE_PRECISION_UNSIGNED_DECIMAL": false, "modbus_ethernet.DEVICE_OUTPUT_
COILS": 32, "modbus_ethernet.DEVICE_INPUT_COILS": 32, "modbus_ethernet.DEVICE_
INTERNAL_REGISTERS": 32, "modbus_ethernet.DEVICE_HOLDING_REGISTERS": 32, "modbus_
ethernet.DEVICE_PERFORM_BLOCK_READ_ON_STRINGS": 0 }
```


Hinweis: Der Parameter servermain.DEVICE_MODEL verwendet standardmäßig das generische Modbus-Modell. Falls dies nicht gewünscht ist, stellen Sie sicher, dass dieser Parameter korrekt definiert ist.

Modbus Ethernet-Tag-Eigenschaften

Nachfolgend finden Sie eine vollständige Liste aller Modbus Ethernet-Tag-Eigenschaften.


```
{ "common.ALLTYPES_NAME": "MyTag", "common.ALLTYPES_DESCRIPTION": "", "ser-
vermain.TAG_ADDRESS": "400001", "servermain.TAG_DATA_TYPE": 5, "servermain.TAG_READ_
WRITE_ACCESS": 1, "servermain.TAG_SCAN_RATE_MILLISECONDS": 100, "servermain.TAG_
AUTOGENERATED": false, "servermain.TAG_SCALING_TYPE": 0, "servermain.TAG_SCALING_RAW_
LOW": 0, "servermain.TAG_SCALING_RAW_HIGH": 1000, "servermain.TAG_SCALING_SCALED_
DATA_TYPE": 9, "servermain.TAG_SCALING_SCALED_LOW": 0, "servermain.TAG_SCALING_
SCALED_HIGH": 1000, "servermain.TAG_SCALING_CLAMP_LOW": false, "servermain.TAG_
SCALING_CLAMP_HIGH": false, "servermain.TAG_SCALING_NEGATE_VALUE": false, "ser-
vermain.TAG_SCALING_UNITS": "" }
```

Index

A

Adressbeschreibungen 38

Adresse 32

Akkumulation 50

Alle Kanäle abonnieren ein virtuelles Netzwerk oder alle Geräte hören auf Remote-Adressen, wodurch unangeforderte Kommunikation beendet wird. 65

Alle Werte für alle Tags schreiben 11

Allgemein 16

Anfangsaktualisierungen aus Cache 18

Anforderungs-Timeout 19

Anfragen verwerfen, wenn herabgestuft 20

Angegebene Blockgröße für Ausgangs-Coils überschreitet maximale Blockgröße. | Angegebene Blockgröße = <Anzahl> (Coils), Maximale Blockgröße = <Anzahl> (Coils). 61

Angegebene Blockgröße für Eingangs-Coils überschreitet maximale Blockgröße. | Angegebene Blockgröße = <Anzahl> (Coils), Maximale Blockgröße = <Anzahl> (Coils). 61

Angegebene Blockgröße für Halteregister überschreitet maximale Blockgröße. | Angegebene Blockgröße = <Anzahl> (Register), Maximale Blockgröße = <Anzahl> (Register). 61

Angegebene Blockgröße für interne Register überschreitet maximale Blockgröße. | Angegebene Blockgröße = <Anzahl> (Register), Maximale Blockgröße = <Anzahl> (Register). 61

Anmerkung 32

Applicom 31

Applicom-Adressierung 39

Array-Unterstützung 40-42, 47, 49, 54

Arrays 51

Auflösen des Hosts fehlgeschlagen. | Host-Name = '<Name>'. 60

Aufzählungen 31

Aus benutzerdefinierten Anwendungen importieren 32

Ausgabe 50

Ausgangs-Coils 27, 40, 43, 47, 52

Automatische Herabstufung 19

Automatische Tag-Datenbankgenerierung 32

B

BCD 36

Bei doppeltem Tag 21

Bei Eigenschaftsänderung 21
Bei Gerätestart 21
Beim Tag-Import ist eine Dateiausnahme aufgetreten. 62
Benutzer 50
Benutzerdefinierte Tags 32
Beschreibung 16
Beschreibung für Datensatz in Importdatei abgeschnitten. | Datensatznummer = <Nummer>. 63
Beschreibung von Funktionscodes 39
Beschreibungen einschließen 23
Blockanfrage hat mit Ausnahme geantwortet. | Blockbereich = <Adresse> bis <Adresse>, Ausnahme = <Code>. 61
Blockanfrage hat mit Ausnahme geantwortet. | Blockbereich = <Adresse> bis <Adresse>, Funktionscode = <Code>, Ausnahme = <Code>. 62
Blockgrößen 27
BOOL 32
Boolean 36
BYTE 32
Byte-Wechsel-Suffixe 43, 49

C

CEG 31
CEG-Adressierung 50
CEGExtension 38
CSV 32

D

Datensammlung 17
Datensatz 32
Datentypbeschreibung 36
Dezimale Adressierung 50
Diagnose 9, 55
Dienst 50
DINT 32
Double 36
Durch Null ersetzen 12
Durch Tag angegebenes Scan-Intervall berücksichtigen 18
DWord 36

E

Eine Client-Anwendung hat die CEG-Erweiterung über das System-Tag _CEGExtension geändert. | Erweiterung = '<Erweiterung>'. 64

Eingangs-Coils 27, 40, 44, 52

Einstellungen auf Kanalebene 13

Elternteilgruppe 22

Ereignisprotokollmeldungen 58

Erstellen 22

Ethernet 14, 26

Ethernet-Einstellungen 11

Ethernet-Manager wurde angehalten. 64

Ethernet-Manager wurde gestartet. 64

Ethernet-zu-Modbus-Plus-Bridge 6

F

Fehler bei unangeforderter Postfach-Speicherzuordnung. | IP-Adresse = '<Adresse>'. 59

Fehler beim Öffnen der Datei für Tag-Datenbank-Import. | Betriebssystemfehler = '<Fehler>'. 60

Fehler beim Parsen von Datensatz in Importdatei. | Datensatznummer = <Nummer>, Feld = <Feld>. 63

Fehler beim Starten der Winsock-Kommunikation. 58

Fehlerbehandlung 25

Float 36

Fluenta 6

Fluenta-Adressierung 50

Fluenta FGM 31

Force Multiple Coils 39

Force Single Coil 39

Fünfstellige Adressierung 51

G

Generieren 21

Generische Modbus-Adressierung 40

Geräte-ID kann nicht von 'Modbus-Client-Modus' in 'Server-Modus' geändert werden, wenn ein Client verbunden ist. 65

Geräte-ID kann nicht von 'Modbus-Server-Modus' in 'Client-Modus' geändert werden, wenn ein Client verbunden ist. 65

Geräteeigenschaften – Automatische Herabstufung 19

Geräteeigenschaften – Redundanz 28
Geräteeigenschaften – Tag-Generierung 20
Geräteeigenschaften – Zeitvorgabe 18
Gleitkommazahlen 50, 55
Globale Einstellungen 13

H

Halteregister 28, 41, 45, 48, 52
Herabstufen bei Fehler 20
Herabstufungszeitraum 20
Hexadezimale Adressierung 50
HoldingRegisterBlockSize 39

I

ID 16
Identifikation 9
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 Ausnahme geantwortet. | Adresse = '<Adresse>', Ausnahme = <Code>. 64
Inhalt der Hilfe 5
InputCoilBlockSize 38
Instromet 6, 31
Instromet-Adressierung 50
INT 32
InternalRegisterBlockSize 38
Interne Register 28, 40, 44, 52
Interne Tags 38
IP-Protokoll 15, 27

K

Kanal ist in einem virtuellen Netzwerk, alle Geräte verwenden jetzt wieder nur einen Socket pro Gerät. 65
Kanaleigenschaften – Allgemein 9
Kanaleigenschaften – Erweitert 12
Kanaleigenschaften – Ethernet-Kommunikation 11

Kanaleigenschaften – Kommunikationsserialisierung 13
Kanaleigenschaften – Schreiboptimierungen 11
Kanalzuweisung 16
Kommunikations-Timeout 24
Kommunikations-Timeouts 18
Kurze Ganzzahlen 50, 55

L

Lange Ganzzahlen 50
Lastausgleich 14
LBCD 36
Lesegeschützter Zugriff 53
Long 36
Löschen 21

M

Masked Write Register 39
Max. Sockets pro Gerät 15
Modbus-Adressierung 51
Modbus-Ausnahmecodes 67
Modbus-Client 6
Modbus-Ethernet-Kommunikation optimieren 34
Modbus-Postfach 51
Modbus-Server-Modus ist nicht zulässig, wenn sich der Kanal in einem virtuellen Netzwerk befindet. Die Geräte-ID darf keinen Loopback oder keine lokale IP-Adresse enthalten. 65
Modbus – unangefordert 6
Modell 16
Modelle 6

N

Name 16
Netzwerk 1 – Netzwerk 500 13
Netzwerkadapter 11
Netzwerksmodus 14
Nicht geändert 12

Nicht normalisierte Float-Handhabung 12
Nicht scannen, nur Abruf anfordern 18
Nur den letzten Wert für alle Tags schreiben 12
Nur den letzten Wert für nicht boolesche Tags schreiben 11

O

OPC-Qualität schlecht 24
Optimierungsmethode 11
OutputCoilBlockSize 38

P

Port 15, 27, 38
Postfach 6, 31
Postfach-Adressierung 50
Postfach-Modell nicht zulässig, wenn sich der Kanal in einem virtuellen Netzwerk befindet. 65
Postfach-Modus 54
Preset Multiple Registers 39
Preset Single Register 39
Priorität 14

R

Read Coil Status 39
Read Holding Registers 39
Read Input Status 39
Read Internal Registers 39
REAL 32
Redundanz 28
Roxar 7
Roxar-Adressierung 55
Roxar RFM 31

S

Scan-Modus 17
Sechsstellige Adressierung 51

Servicezyklus 12
Setup 7
Short 36
Simuliert 17
Socket-Verbindung konnte nicht erstellt werden. 59
Socket-Verwendung 14, 35
Socket bei Timeout schließen 27
Speicher für Modbus-Server-Gerät erstellt. | Modbus-Server-Geräte-ID = <Gerät>. 65
Statistikelemente 55
String 36
STRING 32
System 50
System-Tags 38

T

Tag-Datenbank wird importiert. | Quelldatei = '<Dateiname>'. 64
Tag-Generierung 20
Tag-Import wegen zu wenig Speicherressourcen fehlgeschlagen. 62
Tag-Zähler 10
Tag konnte nicht importiert werden, da der Datentyp nicht unterstützt wird. | Tag-Name = '<Tag>', nicht unterstützter Datentyp = '<Typ>'. 63
Tags bei unzulässiger Adresse deaktivieren 26
TIME 32
Timeout bis zum Herabstufen 20
Transaktionen pro Zyklus 13
Treiber 16
Treibersystem-Tag-Adressierung 38
TSX Premium 47
TSX Quantum 43

U

Überlegungen zu Modbus-Clients und -Servern 25
Überschreiben 21
Übersicht 5
UDINT 32
UINT 32
Unangefordert 23

Unangeforderte Kommunikation konnte nicht gestartet werden. 58

Unangeforderte Kommunikation wird gestartet. | Protokoll = '<Name>', Port = <Nummer>. 65

Unangeforderte, nicht unterstützte Anfrage erhalten. | IP-Adresse = '<Adresse>'. 59

Unangeforderter Postfach-Zugriff für nicht definiertes Gerät. Socket wird geschlossen. | IP-Adresse = '<Adresse>'. 58

Ungültige Adresse im Block. | Blockbereich = <Adresse> bis <Adresse>. 60

Ungültige Blocklänge erhalten. | Blockbereich = <Start> bis <Ende>. 62

Ungültiges Array. | Array-Bereich = <Start> bis <Ende>. 60

Untergruppen zulassen 22

Unterstützt 6

V

Variable 32

Variablenimporteinstellungen 22

Verbindungs-Timeout 19

Versuche vor Timeout 19

Verzögerung zwischen Geräten 13

Virtuelles Netzwerk 13

W

Wert definieren 32

Word 36

WORD 32

Z

Zeichenfolgen als Block lesen 28

Zeichenfolgenunterstützung 43, 49, 54

Zeitvorgabe 18