Modbus Unsolicited Serial-Treiber

Inhaltsverzeichnis

Modbus Unsolicited Serial-Treiber

Hilfe-Version 1.044

INHALT

Übersicht

Modbus Unsolicited Serial-Treiber bietet eine zuverlässige Möglichkeit, serielle Modbus-Geräte mit Client-Anwendungen, u.a. HMI, SCADA, Historian, MES, ERP und zahlreichen benutzerdefinierten Anwendungen, zu verbinden. Es simuliert bis zu 255 Modbus-Slave-Geräte in einem seriellen Kommunikationsnetzwerk. Andere Geräte oder PCs können mithilfe des Modbus-Protokolls mit jedem simulierten Modbus-Slave-Gerät kommunizieren.

Hinweis: Für diesen Treiber werden die Begriffe "Slave" und "unaufgefordert" synonym verwendet.

Setup

Unterstützte Geräte

Modbus-kompatible Geräte

Kommunikationsprotokoll

Modbus-RTU-Protokoll

Unterstützte Funktionscodes

- Read Coil Status - Code 01H
- Read Input Status - Code 02H
- Read Holding Registers - Code 03H
- Read Internal Registers - Code 04H
- Force Single Coil - Code 05H
- Preset Single Register - Code 06H
- Diagnostic Loopback - Code 08H
- Force Multiple Coils - Code 0FH
- Preset Multiple Registers - Code 10H

Hinweis: Für alle anderen Funktionscodes gibt der Treiber den Ausnahmecode 01H (Funktion nicht implementiert) an das anfragende Gerät zurück.

Broadcast-Befehle

Modbus Unsolicited Serial-Treiber kann Broadcast-Schreibmeldungen empfangen. Broadcast-Meldungen werden durch Verwenden der Stations-ID 0 definiert. Wenn der Treiber eine Schreibmeldung (Funktion 05H, 06H, 0FH oder 10H) mit der Stations-ID 0 empfängt, wird der zu schreibende Wert in alle Geräte platziert, die unter dem Kanal definiert sind, für den der Befehl empfangen wurde. Der Broadcast-Befehl kann im Wesentlichen verwendet werden, um ein Datenelement an all die Geräte zu senden, die zur gleichen Zeit im Treiber konfiguriert wurden.

Für diesen Treiber werden die Begriffe "Slave" und "unaufgefordert" synonym verwendet.

Kanal-Setup

Einstellungen für serielle Kommunikation/Ports

Baudrate: 1200, 2400, 9600, 19200
Parität: "Ungerade", "Gerade", "Keine"
Daten-Bits: 8
Stopp-Bits: 1, 2
Flusssteuerung: Bei Verwendung eines RS232/RS485-Konverters hängt die Art der erforderlichen Flusssteuerung von den Anforderungen des Konverters ab. Einige Konverter benötigen keine Flusssteuerung, bei anderen ist RTS-Flusssteuerung erforderlich. Der Dokumentation zum Konverter entnehmen Sie die zugehörigen Flussanforderungen. Zu empfehlen ist die Verwendung eines RS485-Konverters, der automatische Flusssteuerung bietet.

Hinweise:

1. Bei Verwendung des vom Hersteller mitgelieferten Kommunikationskabels ist es manchmal erforderlich, eine Flusssteuerungseinstellung **RTS** oder **RTS immer** zu wählen.
2. Nicht alle Geräte unterstützen die aufgelisteten Konfigurationen.

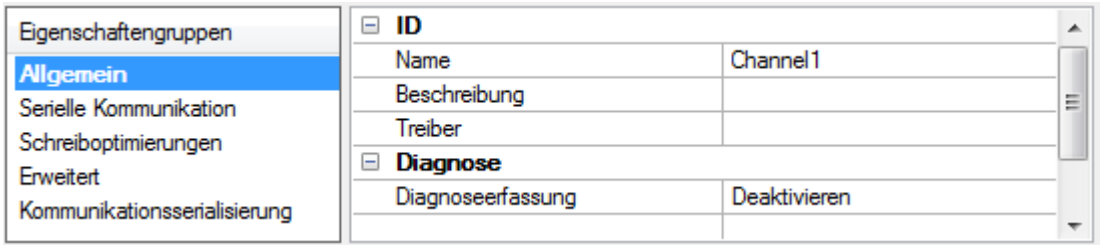
Zeitvorgabe

Siehe Kanaleigenschaften - Zeitvorgabe

Hinweis: Für diesen Treiber werden die Begriffe "Slave" und "unaufgefordert" synonym verwendet.

Kanaleigenschaften – Allgemein

Dieser Server unterstützt die Verwendung von gleichzeitigen Mehrfachkommunikationstreibern. 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: Benutzerdefinierte ID dieses Kanals. 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: Benutzerdefinierte Informationen über diesen Kanal.
Viele dieser Eigenschaften, einschließlich der Beschreibung, verfügen über ein zugeordnetes System-Tag.

Treiber: Ausgewähltes Protokoll/ausgewählter Treiber für diesen Kanal. Diese Eigenschaft gibt 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. Unter Berücksichtigung dessen sollten keine Änderungen an den Eigenschaften erfolgen, sobald eine große Client-Anwendung entwickelt wurde. Verwenden Sie den Benutzermanager, um zu verhindern, dass Operatoren Eigenschaften ändern, und um Zugriffsrechte auf Serverfunktionen zu beschränken.

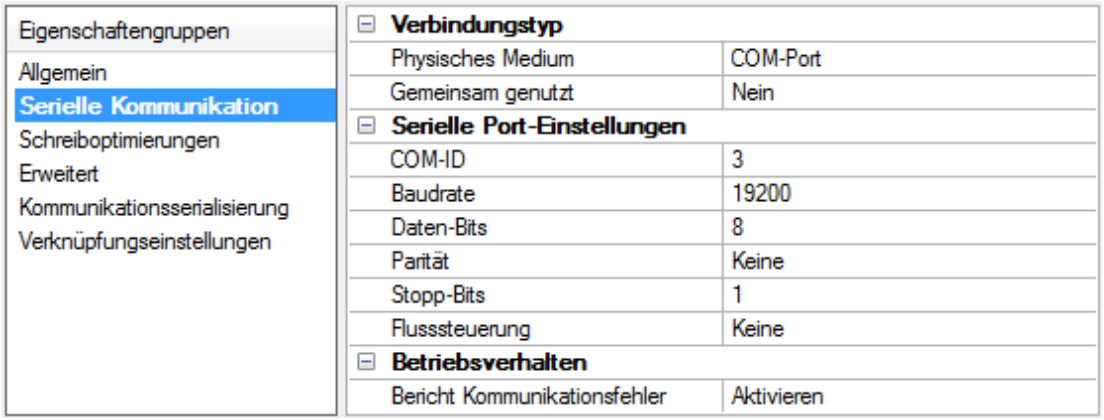
Diagnose

Diagnoseerfassung: Wenn diese Option aktiviert ist, stehen die Diagnoseinformationen des Kanals für OPC-Anwendungen zur Verfügung. Da für die Diagnosefunktionen des Servers eine minimale Mehraufwandsverarbeitung erforderlich ist, wird empfohlen, dass sie bei Bedarf verwendet werden und ansonsten deaktiviert sind. Die Standardeinstellung ist deaktiviert.
Hinweise: Diese Eigenschaft ist nicht verfügbar, wenn der Treiber Diagnosen nicht unterstützt.
Weitere Informationen dazu finden Sie in der Serverhilfe unter „Kommunikationsdiagnosen“.

Kanaleigenschaften – Serielle Kommunikation

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

Hinweis: Beim Online-Vollzeitbetrieb des Servers können diese Eigenschaften jederzeit geändert werden. Schränken Sie mit dem Benutzermanager Zugriffsrechte auf Serverfunktionen ein, da an diesen Eigenschaften vorgenommene Änderungen vorübergehend die Kommunikation beeinträchtigen können.



Verbindungstyp

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

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

Serielle Port-Einstellungen

COM-ID: Geben Sie die Kommunikations-ID an, die bei der Kommunikation mit dem Kanal zugewiesenen Geräten verwendet werden soll. Der gültige Bereich ist 1 bis 9991 bis 16. Die Standardeinstellung ist 1.

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

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

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

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

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

- **Keine**: Mit dieser Option werden keine Steuerleitungen umgeschaltet oder in den aktiven Zustand gebracht.
- **DTR**: Mit dieser Option wird die DTR-Leitung in den aktiven Zustand gebracht, wenn der Kommunikationsport geöffnet ist und es auch bleibt.

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

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

Betriebsverhalten

- **Bericht Komm. Kommunikationsfehler**: Aktivieren oder deaktivieren Sie die Berichterstellung über geringfügige Kommunikationsfehler. Wenn diese Option aktiviert ist, werden geringfügige Fehler beim Auftreten im Ereignisprotokoll angezeigt. Wenn diese Option deaktiviert ist, werden dieselben Fehler nicht angezeigt, selbst wenn es normale Anforderungsfehler sind. Die Standardeinstellung ist "Aktivieren".
- **Inaktive Verbindung schließen**: Wählen Sie diese Option, um die Verbindung zu schließen, wenn es keinerlei Tags mehr gibt, die von einem Client im Kanal referenziert werden. Die Standardeinstellung ist "Aktivieren".
- **Inaktivitätsdauer bis Schließen**: Geben Sie an, wie lang der Server warten soll, bis alle Tags vor dem Schließen des COM-Ports entfernt wurden. Der Standardwert ist 15 Sekunden.

Ethernet-Einstellungen

Hinweis: Nicht alle seriellen Treiber unterstützen Ethernet-Kapselung. Wird diese Gruppe nicht angezeigt, wird die Funktion nicht unterstützt.

Ethernet-Kapselung ermöglicht die Kommunikation mit seriellen Geräten, die im Ethernet-Netzwerk mit Terminalservern verbunden sind. Ein Terminalserver ist im Wesentlichen ein virtueller serieller Port, der TCP/IP-Meldungen im Ethernet-Netzwerk in serielle Daten konvertiert. Sobald die Meldung konvertiert wurde, können Benutzer Standardgeräte verbinden, die eine serielle Kommunikation mit dem Terminalserver unterstützen. Der serielle Port des Terminalservers muss richtig konfiguriert werden, um den Anforderungen des seriellen Geräts zu entsprechen, mit dem er verbunden ist. *Weitere Informationen dazu finden Sie in der Serverhilfe unter „So verwenden Sie Ethernet-Kapselung“.*

- **Netzwerkadapter**: Geben Sie für Ethernet-Geräte in diesem Kanal einen zu bindenden Netzwerkadapter an. Wählen Sie einen Netzwerkadapter für die Bindung, oder lassen Sie die Standardeinstellung vom Betriebssystem auswählen.
  Bestimmte Treiber zeigen unter Umständen zusätzliche Eigenschaften für Ethernet-Kapselung an. Weitere Informationen dazu finden Sie unter Kanaleigenschaften - Ethernet-Kapselung.

Modemeinstellungen

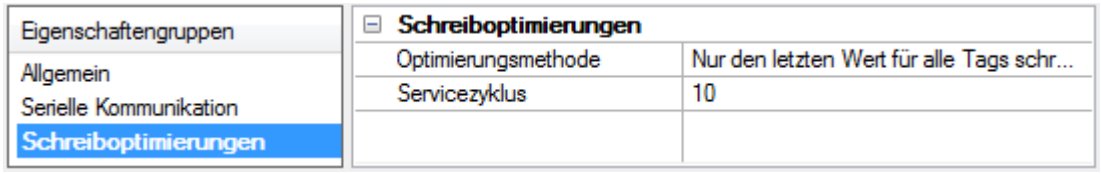
- **Modem**: Geben Sie das installierte Modem an, das für die Kommunikation verwendet werden soll.
- **Verbindungs-Timeout**: Diese Eigenschaft gibt an, wie lang auf das Herstellen von Verbindungen gewartet werden soll, bevor ein Lese- oder Schreibvorgang fehlschlägt. Der Standardwert ist 60 Sekunden.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

Operation ohne Kommunikation

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

Kanaleigenschaften – Schreiboptimierungen

Wie bei jedem Server ist das Schreiben von Daten auf das Gerät unter Umständen der wichtigste Aspekt der Anwendung. Der Server soll sicherstellen, dass die von der Client-Anwendung geschriebenen Daten rechtzeitig auf das Gerät gelangen. In Anbetracht dieses Ziels stellt der Server Optimierungseigenschaften bereit, anhand derer die jeweiligen Anforderungen erfüllt oder die Reaktionsfähigkeit der Anwendungen verbessert werden können.



Schreiboptimierungen

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

- **Alle Werte für alle Tags schreiben**: Mit dieser Option wird der Server gezwungen, für jeden Wert einen Schreibvorgang auf dem Controller zu versuchen. In diesem Modus sammelt der Server weiterhin Schreibanforderungen und fügt sie der internen Schreibwarteschlange des Servers hinzu. Der Server verarbeitet die Schreibwarteschlange und versucht, sie zu leeren, indem er so schnell wie möglich Daten auf das Gerät schreibt. In diesem Modus wird sichergestellt, dass alles, was von den Client-Anwendungen geschrieben wird, an das Zielgerät gesendet wird. Dieser Modus sollte ausgewählt werden, wenn die Reihenfolge des Schreibvorgangs oder der Inhalt des Schreibelements eindeutig auf dem Zielgerät zu finden sein muss.
- **Nur den letzten Wert für nicht boolesche Tags schreiben**: Viele aufeinander folgende Schreibvorgänge für denselben Wert können sich aufgrund der Zeit, die tatsächlich zum Senden der Daten auf das Gerät erforderlich ist, in der Schreibwarteschlange ansammeln. Wenn der Server einen Schreibwert aktualisiert, der bereits in die Schreibwarteschlange eingefügt wurde, sind weitaus weniger Schreibvorgänge

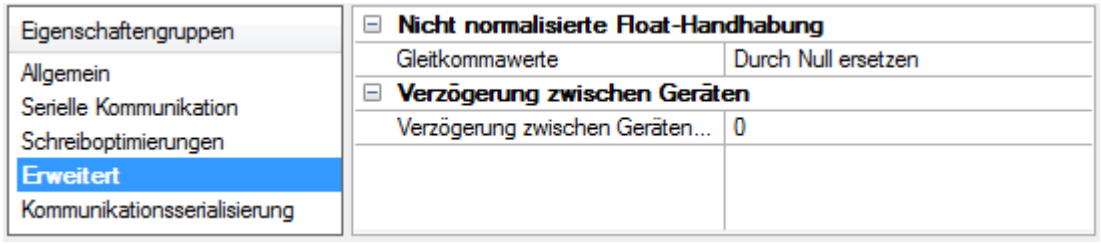
erforderlich, um denselben Endausgabewert zu erhalten. Auf diese Weise sammeln sich keine zusätzlichen Schreibvorgänge in der Warteschlange des Servers an. Wenn der Benutzer den Schiebeschalter nicht mehr verschiebt, erreicht der Wert im Gerät praktisch in derselben Zeit den richtigen Wert. Dem Modus entsprechend wird jeder Wert, der kein boolescher Wert ist, in der internen Warteschlange des Servers aktualisiert und bei der nächstmöglichen Gelegenheit an das Gerät gesendet. Dies kann die Anwendungsleistung erheblich verbessern.
Hinweis: Mit dieser Option wird nicht versucht, Schreibvorgänge in Boolesche Werte zu optimieren. Dadurch können Benutzer den HMI-Datenvorgang optimieren, ohne Probleme mit Booleschen Operationen (z.B. eine vorübergehende Schaltfläche) zu verursachen.

- **Nur den letzten Wert für alle Tags schreiben**: Mit dieser Option wird die hinter der zweiten Optimierungsmethode stehende Theorie auf alle Tags angewendet. Sie ist besonders nützlich, wenn die Anwendung nur den letzten Wert an das Gerät senden muss. In diesem Modus werden alle Schreibvorgänge optimiert, indem die derzeit in der Schreibwarteschlange befindlichen Tags vor dem Senden aktualisiert werden. Dies ist der Standardmodus.

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

Kanaleigenschaften – 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.



Behandlung nicht normalisierter Gleitkommazahlen: 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 senden.

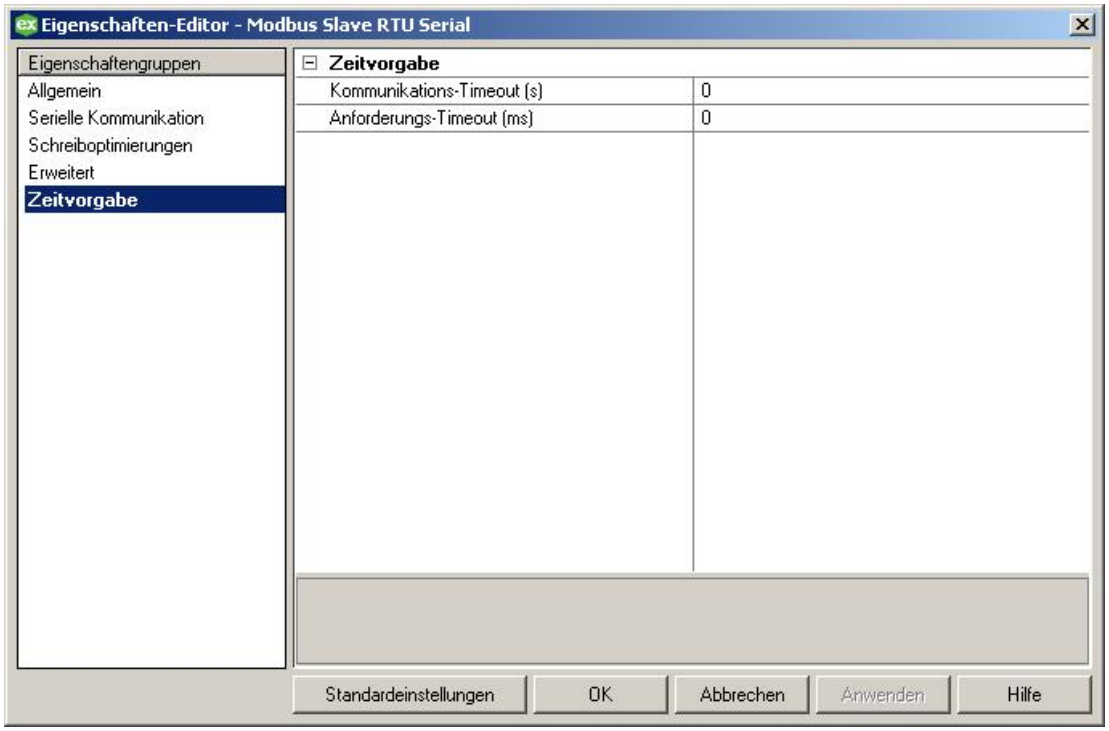
Hinweis: Diese Eigenschaft ist nicht verfügbar, wenn der Treiber keine Gleitkommawerte unterstützt, oder wenn er nur die angezeigte Option unterstützt. Gemäß der Float-Normalisierungseinstellung des Kanals unterliegen nur Echtzeit-Treiber-Tags (wie z.B. Werte und Arrays) der Float-Normalisierung. Beispielsweise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

Weitere Informationen über die Gleitkommawerte finden Sie unter "So arbeiten Sie mit nicht normalisierten Gleitkommawerten" 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 – Zeitvorgabe



Kommunikations-Timeout: Geben Sie an, wie lange der Treiber auf eine eingehende Anforderung wartet, bevor er die Tag-Qualität für alle unaufgeforderten Geräte im Kanal als schlecht festlegt. Nachdem das Kommunikations-Timeout abgelaufen ist, besteht die einzige Möglichkeit, das Timeout zurückzusetzen und eine normale Verarbeitung aller Tags zu ermöglichen, darin, dass Sie die Kommunikation mit dem Gerät wiederherstellen oder das Timeout deaktivieren, indem Sie in der Gruppe "Zeitvorgabe" der Kanaleigenschaften die Einstellung Kommunikations-Timeout auf 0 (Null) setzen. Deaktiviert: 0, aktiviert: 1 --> 64.800 Sekunden (18 Stunden).

Anforderungs-Timeout: Geben Sie an, wie lange der Treiber darauf wartet, einen vollständigen Anforderungs-Frame zu empfangen. Die verstrichene Zeit wird ab dem Moment berechnet, in dem das erste Byte einer neuen Anforderung empfangen wird. Wenn während dieser Zeit kein vollständiger Frame empfangen wird, leert der Treiber die empfangenen Datenpuffer und nimmt an, dass das nächste empfangene Byte der Anfang einer neuen Anforderung ist.

Tipps:

Wählen Sie diese Einstellung sorgfältig. Werte für die Einstellung Anforderungs-Timeout liegen zwischen 0 und 30.000 ms, wobei der Standardwert 0 beträgt. Wenn 0 eingegeben wird, berechnet der Treiber einen angemessenen Timeout durch die Verwendung der folgenden Formel:

$T_{Standard}$ = 1000 * (Bits pro Byte) * 512 *3 / Baud

Dies ist das Dreifache der Zeit, die erforderlich ist, um einen Frame von 512 Byte zu senden. Die Anzahl der Bits pro Byte schließt das Start-Bit sowie die angegebene Anzahl von Daten- und Stopp-Bits ein. Bei einer Baudrate von 9600 sowie 8 Daten-Bits und 1 Stopp-Bit ergibt sich beispielsweise ein Standard-Timeout von 1600 ms. Wenn die Hardware relativ kurze Anforderungs-Frames sendet und eine fehl-

geschlagene Anforderung in weniger als der berechneten Standardzeit (in diesem Beispiel 1600 ms) erneut versucht, konfigurieren Sie einen kürzeren Anforderungs-Timeout.

Der Anforderungs-Timeout sollte nie kürzer sein als die Zeit, die es dauert, den längsten Anforderungs-Frame zu empfangen, der von einem Gerät auf dem Kanal gesendet wird. Dies kann mit der folgenden Formel berechnet werden:

T_{min} = 1000 * (Bits pro Byte) * (max. Frame-Länge) / Baud.

Geräte-Setup

Dieser Treiber simuliert bis zu 255 Modbus-Slave-Geräte in einem seriellen Kommunikationsnetzwerk.

Geräteeigenschaften

Geräteeigenschaften werden in folgende Gruppen unterteilt. Klicken Sie für Details zu den Einstellungen der jeweiligen Gruppe auf einen der nachstehenden Links.

ID
Betriebsmodus
Scan-Modus
Speicher
Redundanz

Unterstützte Funktionscodes

- Read Coil Status - Code 01H
- Read Input Status - Code 02H
- Read Holding Registers - Code 03H
- Read Internal Registers - Code 04H
- Force Single Coil - Code 05H
- Preset Single Register - Code 06H
- Diagnostic Loopback - Code 08H
- Force Multiple Coils - Code 0FH
- Preset Multiple Registers - Code 10H

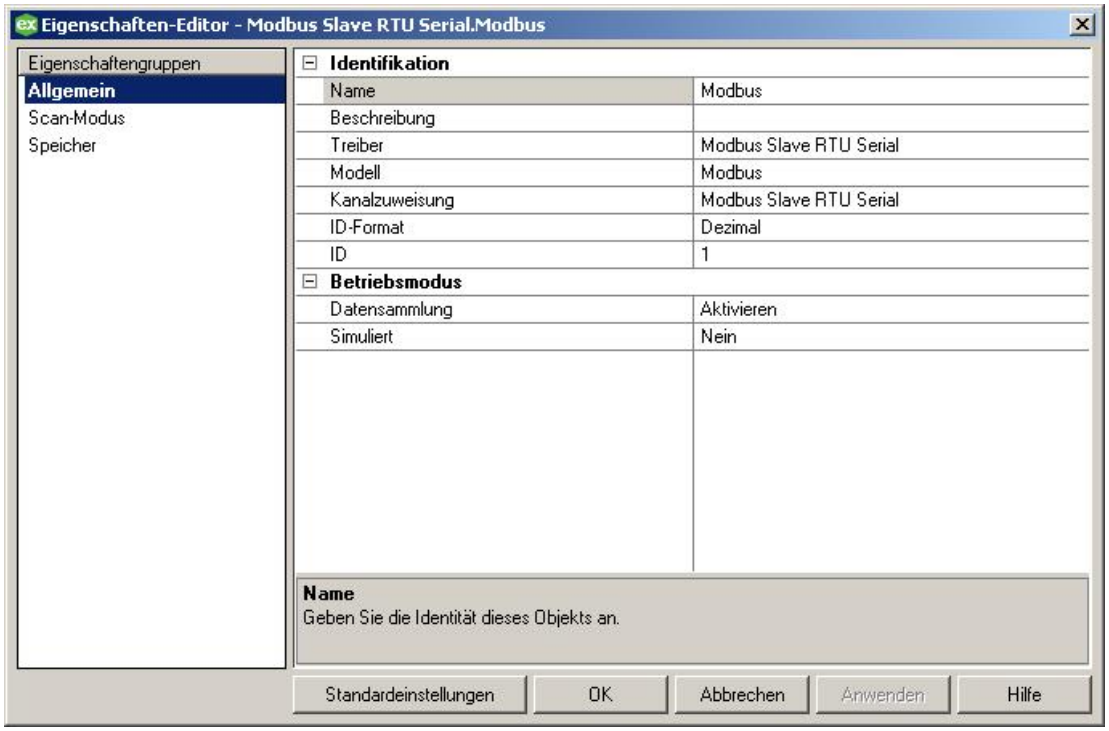
Hinweis: Für alle anderen Funktionscodes gibt der Treiber den Ausnahmecode 01H (Funktion nicht implementiert) an das anfragende Gerät zurück.

Broadcast-Befehle

Modbus Unsolicited Serial-Treiber kann Broadcast-Schreibmeldungen empfangen. Broadcast-Meldungen werden durch Verwenden der Stations-ID 0 definiert. Wenn der Treiber eine Schreibmeldung (Funktion 05H, 06H, 0FH oder 10H) mit der Stations-ID 0 empfängt, wird der zu schreibende Wert in alle Geräte platziert, die unter dem Kanal definiert sind, für den der Befehl empfangen wurde. Der Broadcast-Befehl kann im Wesentlichen verwendet werden, um ein Datenelement an all die Geräte zu senden, die zur gleichen Zeit im Treiber konfiguriert wurden.

Hinweis: Für diesen Treiber werden die Begriffe "Slave" und "unaufgefordert" synonym verwendet.

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 Geräte.

Modell: Die jeweilige Version des Geräts.

ID-Format: Wählen Sie aus, wie die Geräte-ID formatiert wird. Zu den Optionen zählen Formatierungen als Dezimal-, Oktal- oder Hexadezimalzahl.

ID: Seriellen Modbus-Geräten werden Geräte-IDs im Bereich 0 bis 255 zugewiesen.

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".

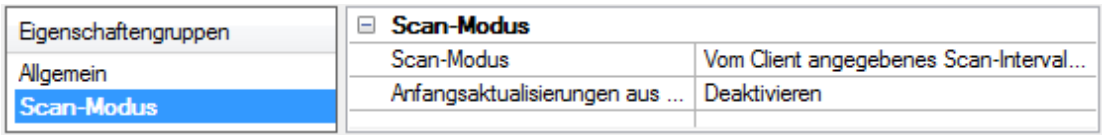
Hinweise:

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.

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: Gibt 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 Clients, nach 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.

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

Adressbeschreibungen

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

Modbus-Adressierung
Daniels/Enron-Adressierung

Modbus-Adressierung

5-stellige Adressierung vs. 6-stellige 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 ist eine 2-Byte-Ganzzahl ohne Vorzeichen (65.535). Sechs Stellen sind erforderlich, um die gesamte Adresstabelle und das Element darzustellen. Deshalb werden Adressen, die im Handbuch des Geräts als 0xxxx, 1xxxx, 3xxxx oder 4xxxx angegeben sind, mit einer zusätzlichen Null aufgefüllt, sobald sie auf das Adressfeld eines Modbus-Tags angewendet werden.

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

Modbus-Adressierung

Die folgenden Adressbeschreibungen gelten für den Zugriff der Client-Anwendung auf jedes simulierte Modbus-Slave-Gerät. Die Client-Anwendung steuert den Speicher des simulierten Modbus-Slave-Geräts. Daher besteht für alle Bereiche Lese-/Schreibzugriff. Die Standard-Datentypen für dynamisch definierte Tags werden **fett** dargestellt.

Adresse	Bereich*	Datentyp	Zugriff
Ausgangs-Coils	000001-065536	**Boolean**	Lesen/Schreiben
Eingangs-Coils	100001-165536	**Boolean**	Lesen/Schreiben
Interne Register	300001-365536 300001-365535 3xxxxx.0-3xxxxx.15	**Word**, Short, BCD Float, DWord, Long, LBCD Boolean, Double	Lesen/Schreiben
Interne Register als Zeichenfolge mit Byte-Reihenfolge Hi-Lo**	300001.2H-365536.240H .Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte.	**String**	Schreibgeschützt
Interne Register als Zeichenfolge mit Byte-Reihenfolge Lo-Hi**	300001.2L-365536.240L .Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte.	**String**	Schreibgeschützt
Halteregister	400001-465536 400001-465535 4xxxxx.0-4xxxxx.15	**Word**, Short, BCD Float, DWord, Long, LBCD Boolean, Double	Lesen/Schreiben
Halteregister als Zeichenfolge mit Byte-Reihenfolge Hi-Lo	400001.2H-465536.240H .Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte.	**String**	Lesen/Schreiben

Adresse	Bereich*	Datentyp	Zugriff
Halteregister als Zeichenfolge mit Byte-Reihenfolge Lo-Hi	400001.2L-465536.240L .Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte.	**String**	Lesen/Schreiben

*Der Maximalbereich wird durch den in der Geräteeigenschaft "Speicher" festgelegten Wert bestimmt. Weitere Informationen finden Sie unter Speicher.
**Diese Adresse unterstützt Funktionscode 04 und gilt nur für dezimale Adressierung.

Array-Unterstützung

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

4xxxx [Zeilen] [Spalten]

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

Zeichenfolgenunterstützung

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

Zeichenfolgenbeispiele

- Um eine Zeichenfolge zu adressieren, die bei 400200 beginnt sowie eine Länge von 100 Byte und die Byte-Reihenfolge Hi-Lo aufweist, geben Sie "400200.100H" ein.
- Um eine Zeichenfolge zu adressieren, die bei 400500 beginnt sowie eine Länge von 78 Byte und die Byte-Reihenfolge Lo-Hi aufweist, geben Sie "400500.78L" ein.

Hinweis: Für diesen Treiber werden die Begriffe "Slave" und "unaufgefordert" synonym verwendet.

Daniels/Enron-Adressierung

Die folgenden Adressbeschreibungen gelten für den Zugriff der Client-Anwendung auf jedes simulierte Daniels/Enron-Slave-Gerät. Die Client-Anwendung steuert den Speicher des simulierten Slave-Geräts. Daher besteht für alle Bereiche Lese-/Schreibzugriff.

Die Standard-Datentypen für dynamisch definierte Tags werden ggf. **fett** dargestellt. In der folgenden Tabelle wird angenommen, dass das Slave-Gerät für den maximal zulässigen Adressbereich von 0 bis 65535 konfiguriert wurde. Weitere Informationen finden Sie unter Speicher.

Adresse	Bereich*	Datentyp	Zugriff
Ausgangs-Coils	000000-065535	**Boolean**	Lesen/Schreiben
Eingangs-Coils	100000-165535	**Boolean**	Lesen/Schreiben
Interne Register	300000-365535 300000-365534 300000-365532 300000.0-365535,15	**Word**, Short, BCD Float, DWord, Long, LBCD Double Boolean	Lesen/Schreiben

Adresse	Bereich*	Datentyp	Zugriff
Halteregister	400000-405000 406000-407000 408000-465535 400000-404999 406000-406999 408000-465534 400000-404999 405001-405999 406000-406999 408000-465534 400000-404999 406000-406999 407001-407999 408000-465534 400000-404997 406000-406997 408000-465532	**Word**, Short, BCD DWord, LBCD Long Float Double	Lesen/Schreiben
Halteregister als boolesche Werte	400000.xx-405000.xx 405001.yy-405999.yy 406000.xx-465535.xx xx ist die Bit-Nummer von 0 bis 15 yy ist die Bit-Nummer von 0 bis 31	**Boolean**	Lesen/Schreiben
Halteregister als Zeichenfolge mit Byte-Reihenfolge Hi-Lo	400000.xxxH-405000.xxxH 406000.xxxH-407000.xxxH 408000.xxxH-465535.xxxH xxx ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte.	**String**	Lesen/Schreiben
Halteregister als Zeichenfolge mit Byte-Reihenfolge Lo-Hi	400000.xxxL-405000.xxxL 406000.xxxL-407000.xxxL 408000.xxxL-465535.xxxL xxx ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte.	**String**	Lesen/Schreiben

*Der Maximalbereich wird durch den Wert in der Geräteeigenschaft "Speicher" bestimmt. Weitere Informationen finden Sie unter Speicher.

Array-Unterstützung

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

4xxxx [Zeilen] [Spalten]

4xxxx [Spalten]: Bei dieser Methode wird angenommen, dass "Zeilen" gleich 1 ist.
Für Word-, Short- und BCD-Arrays darf die Basisadresse + (Zeilen * Spalten) den Wert 65535 nicht überschreiten.
Für Float-, DWord-, Long- und Long BCD-Arrays darf die Basisadresse-+ (Zeilen * Spalten * 2) den Wert 65534 nicht überschreiten.

Zeichenfolgenunterstützung

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

Zeichenfolgenbeispiele

- Um eine Zeichenfolge zu adressieren, die bei 400200 beginnt sowie eine Länge von 100 Byte und die Byte-Reihenfolge Hi-Lo aufweist, geben Sie "400200.100H" ein.
- Um eine Zeichenfolge zu adressieren, die bei 400500 beginnt sowie eine Länge von 78 Byte und die Byte-Reihenfolge Lo-Hi aufweist, geben Sie "400500.78L" ein.

Hinweis: Für diesen Treiber werden die Begriffe "Slave" und "unaufgefordert" synonym verwendet.

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 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).

Adressengröße geändert. | Vorherige Größe = <Nummer>, Aktuelle Größe = <Nummer>.

Fehlertyp:
Fehler

Mögliche Ursache:
Die Adressengröße des angegebenen Geräts wurde geändert.

Mögliche Lösung:
Vergewissern Sie sich, dass die neue Adressengröße richtig ist.

Fehlermaskendefinitionen

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

Modbus-Ausnahmecodes

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

Code dezimal/hexadezimal	Name	Beschreibung
01/0x01	ILLEGAL FUNCTION	Der in der Abfrage erhaltene Funktionscode ist keine zulässige Aktion für den Server (oder Slave). 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 (oder Slave) 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 (oder Slave). 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, und 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 (oder Slave). 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	SLAVE DEVICE FAILURE	Ein nicht wiederherstellbarer Fehler ist aufgetreten, während der Server (oder Slave) versucht hat, die angeforderte Aktion auszuführen.
05/0x05	ACKNOWLEDGE	Der Slave hat die Anfrage akzeptiert und verarbeitet sie, aber dies wird viel Zeit in Anspruch nehmen. Diese Antwort wird zurückgegeben, um einen Timeout-Fehler im Master zu verhindern. Der Master kann als Nächstes eine Meldung ausgeben, dass das Abrufprogramm abgeschlossen ist, um zu ermitteln, ob Verarbeitung abgeschlossen ist.
06/0x06	SLAVE DEVICE BUSY	Der Slave ist mit der Verarbeitung eines lang dauernden Programmbefehls beschäftigt. Der Master muss die Nachricht später erneut senden, wenn der Slave frei ist.
07/0x07	NEGATIVE ACKNOWLEDGE	Der Slave kann die in der Abfrage erhaltene Programmfunktion nicht ausführen. Dieser Code wird für eine erfolglose Programmieranfrage mit Funktionscode 13 oder 14 (dezimal) zurückgegeben. Der Master muss Diagnose- oder Fehlerinformationen vom Slave anfordern.
08/0x08	MEMORY PARITY ERROR	Der Slave hat versucht, Erweiterungsspeicher zu lesen, aber dabei einen Paritätsfehler im Arbeitsspeicher gefunden. Der Master kann die Anfrage erneut versuchen, aber möglicherweise muss das Slave-Gerät gewartet werden.
10/0x0A	GATEWAY PATH UNAVAILABLE	Die spezielle Verwendung in Verbindung mit Gateways deutet darauf hin, dass das Gateway keinen internen Nachrichtenpfad 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	Die spezielle Verwendung in Verbindung mit Gateways deutet

Code dezimal/hexadezimal	Name	Beschreibung
	TARGET DEVICE FAILED TO RESPOND	darauf hin, dass keine Antwort vom Zielgerät empfangen wurde. Bedeutet normalerweise, dass das Gerät im Netzwerk nicht vorhanden ist.

Hinweis: Für diesen Treiber werden die Begriffe "Slave" und "unaufgefordert" synonym verwendet.

Index

3

5

6

A

B

D

E

F

G

H

I

K

L

M

N

P

R

S

T

U

W

Z