

Modbus-Plus-Treiber

© 2018 PTC Inc. Alle Rechte vorbehalten.

Inhaltsverzeichnis

Modbus-Plus-Treiber	1
Inhaltsverzeichnis	2
Modbus-Plus-Treiber	4
Übersicht	5
Externe Abhängigkeiten	5
Setup	6
Kanaleigenschaften	7
Kanaleigenschaften - Allgemein	7
Kanaleigenschaften - Schreiboptimierungen	8
Kanaleigenschaften - Erweitert	9
Kanaleigenschaften - Adapter	9
Treibergeräteeigenschaften	11
Geräteeigenschaften - Allgemein	11
Geräte-ID	12
Geräteeigenschaften - Scan-Modus	16
Geräteeigenschaften - Zeitvorgabe	16
Geräteeigenschaften - Automatische Herabstufung	17
Geräteeigenschaften - Tag-Generierung	18
Geräteeigenschaften - Blockgrößen	20
Geräteeigenschaften - Variablenimporteinstellungen	21
Geräteeigenschaften - Redundanz	22
Automatische Tag-Datenbankgenerierung	23
Kommunikation optimieren	24
Datentypbeschreibung	26
Adressbeschreibungen	27
Modbus-Adressierung	27
Beschreibung von Funktionscodes	30
Gerät für globale Datenkommunikation konfigurieren	30
TIO-Moduladressierung	31
Ereignisprotokollmeldungen	33
Ungültige Adresse im Block. Blockbereich = <Start> bis <Ende>.	33
Ungültige Adresse im Block. Blockbereich = H<Start> bis H<Ende>.	33
Gerät MBPLUS.SYS konnte nicht gestartet werden.	33
Karte konnte nicht festgestellt werden, oder Modbus Plus-Dienste konnten nicht gestartet werden. Vergewissern Sie sich, dass die Karte und die Treiber *.sys von MBP ordnungsgemäß installiert sind.	33
Systemressourcen zum Ausführen dieses Treibers konnten nicht erstellt werden.	33
Kanal konnte nicht initialisiert werden.	34
Ungültiges Array. Array-Bereich = <Start> bis <Ende>.	34
Kanal konnte nicht geladen werden. Nur ein Kanal pro Hilscher-Adapter ist zulässig. Ändern Sie das Projekt so, dass jeder Kanal über einen eindeutigen Adapter verfügt, und laden Sie neu.	34
Fehler beim Öffnen der Datei für Tag-Datenbank-Import. Betriebssystemfehler = '<Fehler>'.	34

Fehler beim Öffnen des MBPLUS-Pfads. Pfad = '<Pfad>'.	34
Empfangene Blocklänge stimmt nicht mit erwarteter Länge überein. Empfangene Länge = <Anzahl> (Byte), Erwartete Länge = <Anzahl> (Byte).	34
Globale Daten nicht verfügbar von Gerät.	34
Fehler beim Lesen der globalen Daten vom Gerät.	34
Blockanfrage auf Gerät hat mit Ausnahme geantwortet. Blockbereich = <Start> bis <Ende>, Ausnahme = <Code>.	35
In Adresse auf Gerät kann nicht geschrieben werden. Gerät hat mit Ausnahme geantwortet. Adresse = '<Adresse>', Ausnahme = <Code>.	35
Von Adresse auf Gerät kann nicht gelesen werden. Gerät hat mit Ausnahme geantwortet. Adresse = '<Adresse>', Ausnahme = <Code>.	35
Blockadressenanfrage hat mit Ausnahme geantwortet. Blockbereich = H<Start> bis H<Ende>, Ausnahme = <Code>.	35
Warnung: Globale Daten sind deaktiviert. Zugriff erfordert Systemtreiber auf niedriger Ebene von Modicon 4.0.	36
Adapter kann nicht geöffnet werden. Adapter = <Name>.	36
Tag-Import wegen zu wenig Speicherressourcen fehlgeschlagen.	36
Beim Tag-Import ist eine Dateiausnahme aufgetreten.	36
Fehler beim Parsen von Datensatz in Importdatei. Datensatznummer = <Nummer>, Feld = <Anzahl>.	36
Beschreibung für Datensatz in Importdatei abgeschnitten. Datensatznummer = <Nummer>.	36
Importierter Tag-Name ist ungültig und wurde geändert. Tag-Name = '<Tag>', geänderter Tag-Name = '<Tag>'.	37
Tag konnte nicht importiert werden, da der Datentyp nicht unterstützt wird. Tag-Name = '<Tag>', nicht unterstützter Datentyp = '<Typ>'.	37
In Adresse auf Gerät kann nicht geschrieben werden. Platine hat mit Ausnahme geantwortet. Adresse = '<Adresse>', Ausnahme = <Code>.	37
Von Adresse auf Gerät kann nicht gelesen werden. Platine hat mit Ausnahme geantwortet. Adresse = '<Adresse>', Ausnahme = <Code>.	38
Gerät MBPLUS.SYS wurde gestartet	38
Tag-Datenbank wird importiert. Quelldatei = '<Dateiname>'.	38
Hilscher CIF Ausnahmecodes	38
Modbus Ausnahmecodes	39
Index	41

Modbus-Plus-Treiber

Hilfeversion 1.054

INHALT

Übersicht

Was ist Modbus-Plus-Treiber?

Geräte-Setup

Wie konfiguriere ich ein Gerät für die Verwendung mit diesem Treiber?

Automatische Tag-Datenbankgenerierung

Wie kann ich Tags für Modbus-Plus-Treiber konfigurieren?

Modbus-Plus-Kommunikation optimieren

Wie erziele ich die beste Leistung mit dem Treiber?

Datentypbeschreibung

Welche Datentypen unterstützt Modbus-Plus-Treiber?

Adressbeschreibungen

Wie adressiere ich eine Datenposition auf einem Modbus-Plus-Gerät?

Ereignisprotokollmeldungen

Welche Fehlermeldungen können bei Modbus-Plus-Treiber auftreten?

Übersicht

Der Modbus-Plus-Treiber bietet eine zuverlässige Möglichkeit, Modbus-Plus-Geräte mit OPC-Client-Anwendungen, u.a. HMI, SCADA, Historian, MES, ERP und zahlreichen benutzerdefinierten Anwendungen, zu verbinden. Es ist für die Verwendung mit einer Schnittstellenkarte vom Typ Modicon SA85, Schneider PCI-85 oder Hilscher CIF 50-MBP vorgesehen. Dieser Treiber unterstützt keine Konfigurationen, in denen SA85-/PCI-85- und Hilscher-CIF-Karten in demselben Computer vorhanden sind.

Externe Abhängigkeiten

Dieser Treiber hat externe Abhängigkeiten.

Ein SA85- oder PCI-85-Schnittstellenadapter sowie die zugehörige Gerätetreiber-Software (MBPLUS- oder MBX-Treiber) müssen mit dem Modbus-Plus-Netzwerk kommunizieren. Der Schnittstellenadapter und die Gerätetreiber können von Modicon/Schneider erworben werden. Der Server kann bis zu acht (8) Kanäle pro SA85- oder PCI-85-Karte unterstützen.

Für Konfigurationen mit der Hilscher-CIF-Karte erfordert Modbus-Plus-Treiber eine Installation der Hilscher-SyCon-Konfigurationssoftware auf dem Computer, auf dem die Karte installiert ist. Die Karte kann über SyCon konfiguriert und heruntergeladen werden. Die Hilscher-CIF-Karte unterstützt einen Kanal pro Adapter.

Setup

Für diesen Treiber werden die Begriffe "Slave" und "unaufgefordert" synonym verwendet. Modbus-Plus-Treiber unterstützt maximal acht Kanäle pro SA85-Kartenadapter oder einen Kanal pro Hilscher-CIF-Kartenadapter.

Unterstützt Schnittstellenkarten

SA85-Karte

Hilscher-CIF-Karte

● **Hinweis:** Benutzer können eine Verbindung zu einem Modbus-Plus-Netzwerk auch von einem seriellen Modbus-RTU-Rechner über einen USB-Adapter herstellen.

● **Weitere Informationen zu den Anforderungen für SA85-Karten und Hilscher CIF-Karten finden Sie unter [Externe Abhängigkeiten](#).**

Unterstützte Kommunikationsmodi

Modbus-Plus-Treiber unterstützt drei Kommunikationsmodi, die verwendet werden, um Daten vom Gerät abzurufen. Der Modus wird durch das Format der Geräte-ID angegeben. Zu den Optionen zählen angeforderter, unaufgeforderter und Postfach-Modus. Die Modi lassen sich wie folgt beschreiben:

- **Angefordert:** In diesem Modus generiert der Treiber Lese- und/oder Schreibforderungen für Daten an das Gerät. Zu den verfügbaren Adressen zählen Ausgangs-Coils, Eingangs-Coils, interne Register und Haltereister. Adressen von Ausgangs-Coils und Haltereistern haben Lese-/Schreibzugriff, wohingegen Eingangs-Coils und interne Register schreibgeschützten Zugriff haben. Das Format der Geräte-ID für den angeforderten Modus lautet *DM.r1.r2.r3.r4.r5* oder *S.r1.r2.r3.r4.r5*.
- **Unaufgefordert:** In diesem Modus fungiert der Treiber als virtuelle SPS im Netzwerk. Lese- und Schreibvorgänge gehen nicht vom Treiber aus. Jede Client-Anwendung, die auf einem unaufgeforderten Gerät liest oder schreibt, übergibt Daten an einen lokalen Cache, der für das Gerät zugeordnet ist, nicht an das physische Gerät. Dieser lokale Cache befindet sich auf dem PC, auf dem der Treiber ausgeführt wird. Geräte im Netzwerk führen Lese- und Schreibvorgänge durch unaufgeforderte Befehle in demselben Cache aus. Das Format der Geräte-ID für den unaufgeforderten Modus lautet *DS.r1.r2.r3.r4.r5*.
- **Postfach:** In diesem Modus fungiert der Treiber als Speicherbereich für jedes definierte Postfach-Gerät. Wenn ein unaufgeforderter Befehl empfangen wird, erkennt der Treiber den Routing-Pfad, von dem die Nachricht kommt, und platziert anschließend die Daten im für dieses Gerät zugeordneten Speicherbereich. Wenn die Nachricht von einem Gerät mit einem Routing-Pfad stammt, das nicht als Postfach-Gerät definiert wurde, wird die Nachricht nicht verarbeitet. Jede Client-Anwendung, die von einem Postfach-Gerät liest, liest Daten aus dem im Treiber enthaltenen Speicherbereich statt vom physischen Gerät. Schreibvorgänge werden jedoch sowohl an das physische Gerät als auch an den lokalen Cache gesendet. In diesem Modus werden nur Haltereister unterstützt. Der Datentyp Double wird nicht unterstützt. Das Format der Geräte-ID für den Postfach-Modus lautet *U.r1.r2.r3.r4.r5*.
 - **Hinweis:** Unaufgeforderte Postfach-Befehle werden durch die MSTR-Anweisung ermöglicht, die in bestimmten Modicon-Geräten verfügbar ist. Weitere Informationen finden Sie unter "Beispiel 2: Postfach-Modus - Einzelnes Netzwerk" und "Beispiel 3: Postfach-Modus - Netzwerk mit Bridge" unter [Geräte-ID](#).

Informationen zu den Kommunikationsmodi, die von den SA85- und Hilscher-CIF-Schnittstellenkarten unterstützt werden, finden Sie in der folgenden Tabelle.

Modus	SA85-Karte	Hilscher-CIF-Karte
Angefordert	x	x
Unaufgefordert	x	
Postfach	x	

● Weitere Informationen finden Sie unter der [Geräte-ID](#).

Daten von mehreren Geräten abrufen

Modbus-Plus-Treiber kann Daten von mehreren Geräten in einem Modbus-Plus-Netzwerk abrufen und auch als einzelnes Slave-Gerät im Modbus-Plus-Netzwerk fungieren, von dem andere Geräte Daten abrufen können. Der Treiber ist auf 8192 Geräte begrenzt und unterstützt bis zu 4 Adapter.

• Siehe auch:

[Kanaleigenschaften](#)

[Geräteeigenschaften](#)

Kanaleigenschaften

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.

Die einem Kanal zugeordneten Eigenschaften werden in logische Gruppierungen unterteilt. Zwar sind einige Gruppen für einen angegebenen Treiber oder ein angegebenes Protokoll bestimmt, doch handelt es sich bei Folgendem um allgemeine Gruppen:

[Allgemein](#)

[Ethernet oder Serielle Kommunikation](#)

[Schreiboptimierung](#)

[Erweitert](#)

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.

Eigenschaftengruppen	ID	
Allgemein	Name	Channel1
Serielle Kommunikation	Beschreibung	
Schreiboptimierungen	Treiber	
Erweitert	Diagnose	
Kommunikationsserialisierung	Diagnoseerfassung	Deaktivieren

ID

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.

• 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.

• **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.

● **Hinweis:** Diese Eigenschaft ist deaktiviert, wenn der Treiber Diagnosen nicht unterstützt.

● **Weitere Informationen dazu finden Sie in der Serverhilfe unter „Kommunikationsdiagnosen“.**

Kanaleigenschaften - Schreiboptimierungen

Wie bei jedem OPC-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.

Eigenschaftengruppen	☐ Schreiboptimierungen	
Allgemein	Optimierungsmethode	Nur den letzten Wert für alle Tags schr...
Serielle Kommunikation	Servicezyklus	10
Schreiboptimierungen		

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 Schreibforderungen 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.

Eigenschaftengruppen	<input type="checkbox"/> Nicht normalisierte Float-Handhabung	
Allgemein	Gleitkommawerte	Durch Null ersetzen
Serielle Kommunikation	<input type="checkbox"/> Verzögerung zwischen Geräten	
Schreiboptimierungen	Verzögerung zwischen Geräten...	0
Erweitert		
Kommunikationsserialisierung		

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-Gleitkommawerten 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.

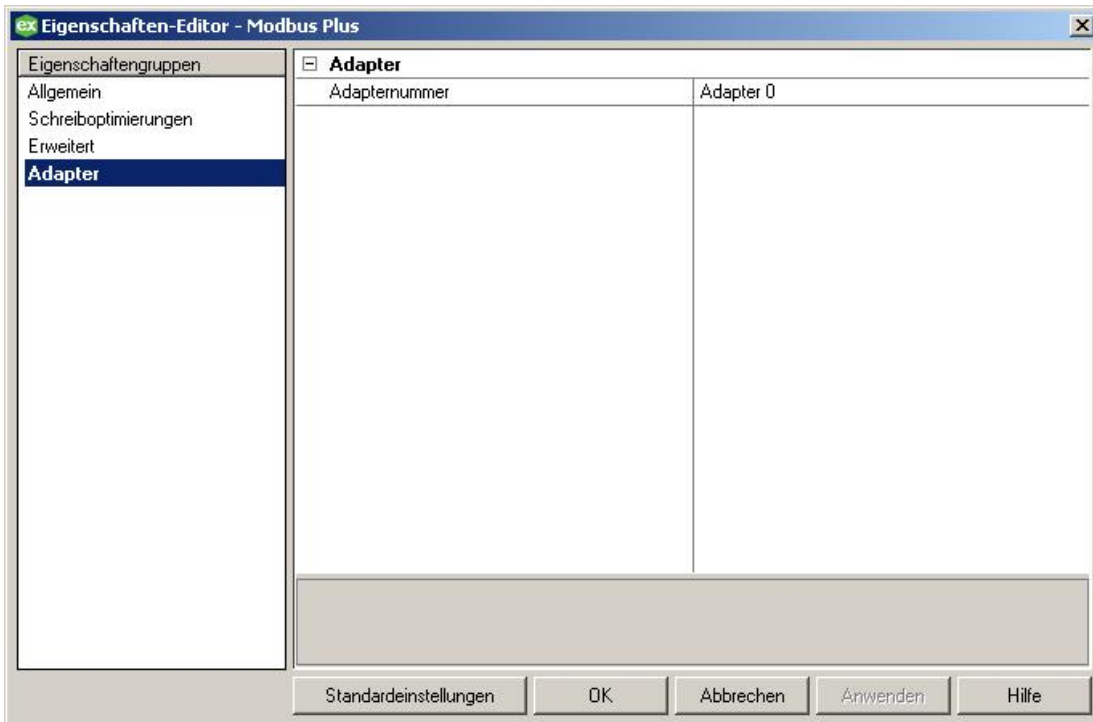
● **Hinweis:** Diese Eigenschaft ist deaktiviert, wenn der Treiber keine Gleitkommawerte unterstützt, oder wenn er nur die angezeigte Option unterstützt. Gemäß der Float-Normalisierungseinstellung des Kanals unterliegen nur Echtzeit-Treiber-Tags (wie z.B. Werte und Arrays) der Float-Normalisierung. Beispielsweise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

● *Weitere Informationen über die Gleitkommawerte finden Sie unter "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 - Adapter



Adapternummer: Gibt die Nummer des von der Modbus Plus-Karte zu verwendenden Adapters an. Gültige Adapternummern sind 0 bis 3. *Kartenspezifische Informationen finden Sie unter [Setup](#).*

Treibergeräteeigenschaften

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](#)

[Zeitvorgabe](#)

[Automatische Herabstufung](#)

[Tag-Generierung](#)

[Blockgrößen](#)

[Variablenimporteinstellungen](#)

[Einstellungen](#)

[Redundanz](#)

Geräteeigenschaften - Allgemein

Eigenschaftengruppen	
Allgemein	
Scan-Modus	
Zeitvorgabe	
Automatische Herabstufung	
Tag-Generierung	
Blockgrößen	
Variablenimporteinstellungen	
Einstellungen	
Redundanz	

Identifikation	
Name	Modbus
Beschreibung	
Treiber	Modbus Plus
Modell	Modbus
Kanalzuweisung	Modbus Plus
ID	DM.1.0.0.0.0

Betriebsmodus	
Datensammlung	Aktivieren
Simuliert	Nein

Name
Geben Sie die Identität dieses Objekts an.

Standardeinstellungen OK Abbrechen Anwenden Hilfe

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.

Modell: Die jeweilige Version des Geräts.

ID: Geben Sie den Pfad zu einem Knoten im Netzwerk an. Die Geräte-ID gibt den Pfad zu einem Knoten im Netzwerk an. Sie besteht neben einem Modus-Designator aus fünf aufeinanderfolgenden Routing-Bytes.

• Weitere Informationen finden Sie unter [Geräte-ID](#).


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 in der Serverhilfe 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 Elementes basiert auf dem Gruppenaktualisierungsintervall. 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-Aktualisierungsintervallen (Gruppenaktualisierungsintervall für OPC-Clients oder Scan-Intervall für native und DDE-Schnittstellen). Das bedeutet, dass zwei Clients, die dasselbe Element mit unterschiedlichen Aktualisierungsintervallen referenzieren, verschiedene Daten zurückgeben.

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

Geräte-ID

Die Geräte-ID gibt den Pfad zu einem Knoten im Netzwerk an. Sie besteht aus einem Modus-Designator und fünf aufeinanderfolgenden Routing-Bytes. Der Modus kann Data Master (DM), Data Slave (DS) oder Mailbox sein. Für diesen Treiber werden die Begriffe "Slave" und "unaufgefordert" synonym verwendet.

Angeforderter Modus (Data Master)

Data Master-Pfade beginnen mit dem Präfix DM oder S und werden für die Kommunikation mit einem anderen Knoten im Netzwerk verwendet. Bei einer solchen Kommunikation fungiert der Host-PC als Master. Ein DM-Pfad kann eine SPS oder andere Geräte identifizieren, die auf Modbus-Lese- oder Schreibbefehle antworten können, einschließlich eines anderen Host-PCs, auf dem Modbus-Plus-Treiber ausgeführt wird. Das Format eines DM-Pfades lautet *DM.r1.r2.r3.r4.r5* oder *S.r1.r2.r3.r4.r5*.

Unaufgeforderter Modus (Data Slave)

Ein einzelner Data Slave-Pfad kann pro SA85-Karte konfiguriert werden und besitzt das Format *DS.1.0.0.0.0*. Durch Definieren eines Slave-Pfades ermöglichen es Benutzer dem Host-PC, Modbus-Plus-Treiber auszuführen, um ein SPS-Gerät im Netzwerk zu simulieren, das auf Lese-/Schreibanforderungen von anderen Geräten antworten kann. Andere Geräte können mit diesem simulierten Gerät kommunizieren, indem sie einen Data Master-Pfad zu diesem öffnen.

Das simulierte SPS-Gerät verwendet eine Modbus-Byte-Reihenfolge: Bei der Datenkodierung wird das erste Wort als Low-DWord für 32-Bit- und 64-Bit-Werte und das erste DWord als Low-DWord für 64-Bit-Werte bewertet. Deshalb müssen die Datenkodierungsoptionen für das unaufgeforderte Gerät folgendermaßen festgelegt werden:

- Modbus-Byte-Reihenfolge
- Erstes Wort "Low"
- Erstes DWord "Low"

 Weitere Informationen finden Sie unter [Einstellungen](#).

Adressen zwischen 1 und 65536 sind für Ausgangs-Coils, Eingangs-Coils, interne Register und Haltereister implementiert. Der Treiber antwortet auf jede gültige Anfrage zum Lesen oder Schreiben dieser Werte von

externen Geräten (Funktionscodes [dezimal] 01, 02, 03, 04, 05, 06, 15, 16). Auf diese Positionen kann auch lokal durch den Host-PC zugegriffen werden, und zwar als dem Slave-Gerät zugewiesene Tags. Ein lesegeschützter Zugriff ist für unaufgeforderte Geräte nicht zulässig.

Wenn ein Slave-Pfad aktiviert wird, aktiviert Modbus-Plus-Treiber acht Slave-Pfade pro SA85-Karte. Dies erlaubt den Remote-SPS und anderen Modbus-Plus-Geräten den Zugriff auf den Slave-Speicher dieses Treibers über einen der acht Slave-Pfade. Dabei wird in allen Fällen auf denselben Speicher zugegriffen. Hinsichtlich einer MSTR-Anweisung können Benutzer einen Pfad zwischen 1 und 8 angeben, wenn sie definieren, mit welchem Pfad eine Verbindung auf der SA85-Karte hergestellt werden soll, die von diesem Treiber bedient wird. Dies kann nützlich sein, wenn die Anwendung über viele Remote-Geräte verfügt, die Daten an den PC senden. Wenn dies der Fall ist, können Benutzer die acht Slave-Pfade zur Verteilung der Last von Remote-Knoten verwenden. Jeder Slave-Pfad in diesem Treiber besitzt einen eigenen Ausführungs-Thread, um ein maximales Leistungsniveau zu gewährleisten.

Wenn im Projekt kein Slave-Gerät definiert ist, ignoriert der Treiber alle empfangenen unaufgeforderten Lese- oder Schreibanforderungen.

● **Hinweis:** Der unaufgeforderte Modus wird nicht für Hilscher-CIF-Karten unterstützt.

Postfach-Modus

Ein Postfach-Pfad beginnt mit dem Präfix U und gibt einen Pfad zu einem physischen Gerät an. Im Slave-Gerät, das im Projekt definiert ist, wird ein Speicherbereich für dieses physische Gerät angegeben. Obwohl das physische Gerät unaufgeforderte Schreibvorgänge an diesen Speicherbereich sendet, kann auch lokal durch den Host-PC darauf zugegriffen werden, und zwar als dem Slave-Gerät zugewiesene Tags. Das Format eines Postfach-Pfads lautet *U.r1.r2.r3.r4.r5*.

Der Treiber öffnet immer einen Slave-Pfad, wenn er unaufgeforderte Postfach-Daten empfängt. Der Pfad, den der Treiber öffnet, lautet *DS.1.0.0.0.0*. Geräte in demselben Modbus-Plus-Netzwerk kommunizieren mit dem Treiber, indem sie den Data Master-Pfad DM öffnen. <lokaler Knoten> .1.0.0.0, wobei der lokale Knoten die auf der SA85-Karte des Host-Computers festgelegte Adresse ist. Eine Beschreibung des Pfads, den Geräte in einem Netzwerk mit Bridge verwenden, finden Sie in [Beispiel 3](#).

Geräte verwenden die Modbus-Plus-MSTR-Anweisung, um dem Treiber Daten zu liefern. Damit der Treiber die Daten einem bestimmten Gerät zuordnen kann, muss der Geräte-ID-Pfad in die ersten fünf Register der empfangenen Daten eingebettet sein. Wenn die ersten fünf Register der Daten nicht mit dem Geräte-ID-Pfad des Geräts im Projekt übereinstimmen, werden die empfangenen Daten verworfen. Für die MSTR-Anweisung wird nur der Schreibbefehl unterstützt.

● **Hinweise:**

1. Der Postfach-Modus wird nicht für Hilscher-CIF-Karten unterstützt.
2. Der Geräte-ID-Pfad ist in den Pfad vom Host-PC zum Gerät eingebettet, nicht in den Gerätepfad zum Host-PC.
3. Ein TIO-Modulgerät unterstützt keine Slave-Netzwerkadressen.
4. Die Geräte-ID sollte nicht geändert werden, während Clients verbunden sind. Wenn dies geschieht, werden die Änderungen erst wirksam, wenn die Verbindung zu allen Clients getrennt und anschließend wiederhergestellt wurde.

Beispiel 1: Angeforderter Modus

Ein einzelnes Netzwerk besteht aus vier Knoten, wobei Knoten 1 und 4 Host-PCs sind, auf denen Software ausgeführt wird, die Modbus-Plus-Treiber verwendet. Knoten 2 und 3 sind SPS. In der folgenden Tabelle wird die Adressierung für das Netzwerk von jedem Knoten aus dargestellt.

Von	Zu Knoten 1	Zu Knoten 2	Zu Knoten 3	Zu Knoten 4
Knoten 1	-----	DM.2.0.0.0.0	DM.3.0.0.0.0	DM.4.1.0.0.0
Knoten 2	DM.1.1.0.0.0	-----	DM.3.0.0.0.0	DM.4.1.0.0.0
Knoten 3	DM.1.1.0.0.0	DM.2.0.0.0.0	-----	DM.4.1.0.0.0

Von	Zu Knoten 1	Zu Knoten 2	Zu Knoten 3	Zu Knoten 4
Knoten 4	DM.1.1.0.0.0	DM.2.0.0.0.0	DM.3.0.0.0.0	-----

● **Hinweis:** Um von einem Host-PC auf das simulierte Gerät zuzugreifen, lautet die letzte Zahl ungleich Null im Pfad immer 1. Dies zeigt den vom Treiber verwendeten Slave-Pfad an.

Beispiel 2: Postfach-Modus Einzelnes Netzwerk

Übertragen der Register 40020 bis 40029 vom Gerät in die Positionen 40001 bis 40010 des Host-PCs. Die Position des Steuerungsblocks kann anders lauten. Die Host-PC-Adresse lautet 7.0.0.0.0 und die Geräteadresse 3.0.0.0.0.

MSTR-Anweisung

Steuerungsblock	40001	-
Datenbereich	40015	Fünf Register vorher starten.
Länge	15	Fünf mehr als die tatsächlichen Daten.

Steuerungsblock

Register	Inhalt	Beschreibung
40001	1	Schreiboperation.
40002	0	Enthält Fehlercode.
40003	15	Anzahl der zu übertragenden Register.
40004	1	Startposition im Host-PC (Register 40001).
40005	7	Pfad zum Host-PC.
40006	1	Pfad zum Host-PC.
40007	0	Pfad zum Host-PC.
40008	0	Pfad zum Host-PC.
40009	0	Pfad zum Host-PC.

Datenbereich

Register	Inhalt	Beschreibung
40015	3	Pfad vom Host-PC zurück zum Gerät, die Geräte-ID.
40016	0	Pfad vom Host-PC zurück zum Gerät.
40017	0	Pfad vom Host-PC zurück zum Gerät.
40018	0	Pfad vom Host-PC zurück zum Gerät.
40019	0	Pfad vom Host-PC zurück zum Gerät.
40020	-	Tatsächlicher Datenbeginn.
40029	-	Tatsächliches Datenende.

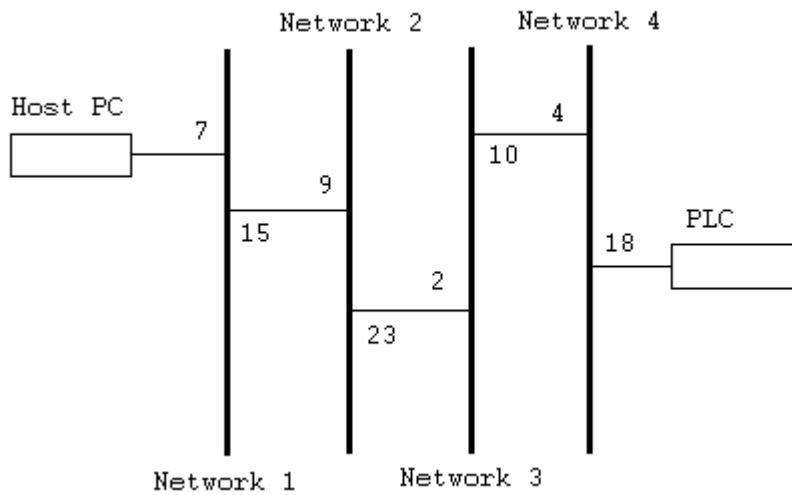
Nachdem der Treiber eine unaufgeforderte Meldung empfangen hat, führt er folgende Aktionen aus:

1. Wenn die Meldung verstanden wurde, sendet der Treiber eine Bestätigung an das sendende Gerät. Wenn Meldungen für andere Funktionen als **Preset Multiple Registers**, Code 0x10, empfangen wurden, gibt der Treiber eine Antwort zurück, dass die Funktion nicht implementiert ist. "Preset Multiple Registers" ist der Funktionscode, der von Geräten verwendet wird, die sich auf der Empfangsseite einer MSTR-Anweisung befinden. Der Treiber gibt eine Ausnahmeantwort zurück, wenn die Meldung nicht verstanden wird oder unvollständig ist.
2. Der Treiber versucht, die ersten fünf Register der empfangenen Daten dem Geräte-ID-Pfad eines Geräts im Projekt zuzuordnen. Wenn kein Pfad gefunden wird, werden die Daten verworfen. Wenn die Daten weniger als sechs Register umfassen, werden sie sofort verworfen.
3. Der Treiber kopiert n - 5 Datenregister ab dem sechsten Register der empfangenen Daten in die intern für das Gerät verwaltete Image Map (beginnend an der in der Meldung angegebenen Position). Der

Treiber muss möglicherweise Speicherplatz für die Image Map zuordnen, wenn es sich um die ersten für diese Positionen empfangenen Daten handelt.

- Die Daten werden den Clients des Treibers zur Verfügung gestellt. Die Daten in diesem Beispiel werden als Tags mit den Adressen 40001 bis 40009 des Geräts mit der Geräte-ID U.3.0.0.0.0 referenziert. Der Client verwendet für das Gerät einen logischen Gerätenamen, der bei Erstellung des Geräts im Projekt zugewiesen wurde. Die Daten können auch als Array referenziert werden, z.B. 40001[10] oder 40001[2][5].

Beispiel 3: Postfach-Modus Netzwerk mit Bridge



Aus der SPS-Perspektive lautet die Host-PC-Adresse 4.2.9.7.1. Aus der Host-PC-Perspektive lautet die SPS-Adresse 15.23.10.18.0. Dies entspricht dem Geräte-ID-Pfad. Wenn dieselben Register von der SPS in dieselben Positionen im Host-PC übertragen werden sollen, wird in der MSTR-Anweisung folgender Steuerungsblock und Datenbereich verwendet (siehe Tabellen unten). Die Meldung wird auf dieselbe Weise verarbeitet.

Hinweis: Bei Verwendung dieses Treibers kann der Host-PC zusätzlich zu einem Gerät über maximal drei Netzwerke verfügen.

MSTR-Anweisung

Steuerungsblock	40001	
Datenbereich	40015	Fünf Register vorher starten.
Länge	15	Fünf mehr als die tatsächlichen Daten.

Steuerungsblock

Register	Inhalt	Beschreibung
40001	1	Schreiboperation.
40002	0	Enthält Fehlercode.
40003	15	Anzahl der zu übertragenden Register.
40004	1	Startposition im Host-PC (Register 40001).
40005	4	Pfad zum Host-PC.
40006	2	Pfad zum Host-PC.
40007	9	Pfad zum Host-PC.
40008	7	Pfad zum Host-PC.
40009	1	Pfad zum Host-PC.

Datenbereich

Register	Inhalt	Beschreibung
40015	15	Pfad vom Host-PC zurück zum Gerät, die Geräte-ID.
40016	23	Pfad vom Host-PC zurück zum Gerät.
40017	10	Pfad vom Host-PC zurück zum Gerät.
40018	18	Pfad vom Host-PC zurück zum Gerät.
40019	0	Pfad vom Host-PC zurück zum Gerät.
40020	-	Tatsächlicher Datenbeginn.
40029	-	Tatsächliches Datenende.

Geräteeigenschaften - Scan-Modus

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

Eigenschaftengruppen	Scan-Modus	
Allgemein	Scan-Modus	Vom Client angegebenes Scan-Intervall...
Scan-Modus	Anfangsaktualisierungen aus ...	Deaktivieren

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 das maximale Scan-Intervall an, das verwendet werden soll. 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.

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.

Eigenschaftengruppen	<input type="checkbox"/> Kommunikations-Timeouts	
Allgemein	Anforderungs-Timeout (ms)	5000
Scan-Modus	Erneute Versuche	3
Zeitvorgabe	<input type="checkbox"/> Zeitvorgabe	
Automatische Herabstufung	Verzögerung zwischen Anfragen (ms)	0

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

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

Zeitvorgabe

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

● **Hinweis:** Nicht alle Treiber unterstützen Verzögerung zwischen Anfragen. Diese Einstellung wird nicht angezeigt, wenn sie nicht zur Verfügung steht.

Geräteeigenschaften - Automatische Herabstufung

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

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 dazu finden Sie in den Eigenschaftsbeschreibungen unten.

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: Diese Eigenschaft gibt an, wann OPC-Tags automatisch generiert werden. Es folgen Beschreibungen der Optionen:

- **Nicht beim Start erstellen:** Mit dieser Option wird verhindert, dass der Treiber irgendwelche OPC-Tags dem Tag-Raum des Servers hinzufügt. Dies ist die Standardeinstellung.
- **Immer beim Start erstellen:** Das Auswählen dieser Option hat zur Folge, dass der Treiber das Gerät für Tag-Informationen bewertet. Es werden auch jedes Mal, wenn der Server gestartet wird, Tags dem Tag-Raum des Servers hinzugefügt.
- **Beim ersten Start erstellen:** Das Auswählen dieser Option hat zur Folge, dass der Treiber das Zielgerät für Tag-Informationen bewertet, wenn das Projekt zum ersten Mal ausgeführt wird. Es werden bei Bedarf auch sämtliche OPC-Tags dem Tag-Raum des Servers hinzugefügt.
- **Hinweis:** Wenn die Option zum automatischen Generieren von OPC-Tags ausgewählt wird, müssen sämtliche Tags, die dem Tag-Raum des Servers hinzugefügt werden, mit dem Projekt gespeichert werden. Benutzer können das Projekt konfigurieren, um automatisch über das Menü **Tools | Optionen** zu speichern.

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

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

- **Bei Erstellen löschen:** Mit dieser Option werden sämtliche Tags gelöscht, die zuvor dem Tag-Raum hinzugefügt wurden, bevor sämtliche neuen Tags hinzugefügt werden. Dies ist die Standardeinstellung.
- **Nach Bedarf überschreiben:** Mit dieser Option wird der Server angewiesen, nur die Tags zu entfernen, die der Kommunikationstreiber durch neue Tags ersetzt. Sämtliche Tags, die nicht überschrieben werden, bleiben im Tag-Raum des Servers.
- **Nicht überschreiben:** Mit dieser Option wird verhindert, dass der Server sämtliche Tags entfernt, die zuvor generiert wurden oder bereits auf dem Server vorhanden waren. Der Kommunikationstreiber kann nur Tags hinzufügen, die völlig neu sind.
- **Nicht überschreiben, Fehler protokollieren:** Diese Option hat denselben Effekt wie die vorherige Option und sendet auch eine Fehlermeldung an das Ereignisprotokoll des Servers, wenn eine Tag-Überschreibung stattgefunden hätte.
- **Hinweis:** Das Entfernen von OPC-Tags wirkt sich auf Tags, die automatisch vom Kommunikationstreiber generiert wurden, sowie auf sämtliche Tags aus, die unter Verwendung von Namen, die generierten Tags entsprechen, hinzugefügt wurden. Benutzer sollten es vermeiden, Tags dem Server unter Verwendung von Namen hinzuzufügen, die möglicherweise den Tags entsprechen, die automatisch vom Treiber generiert werden.

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

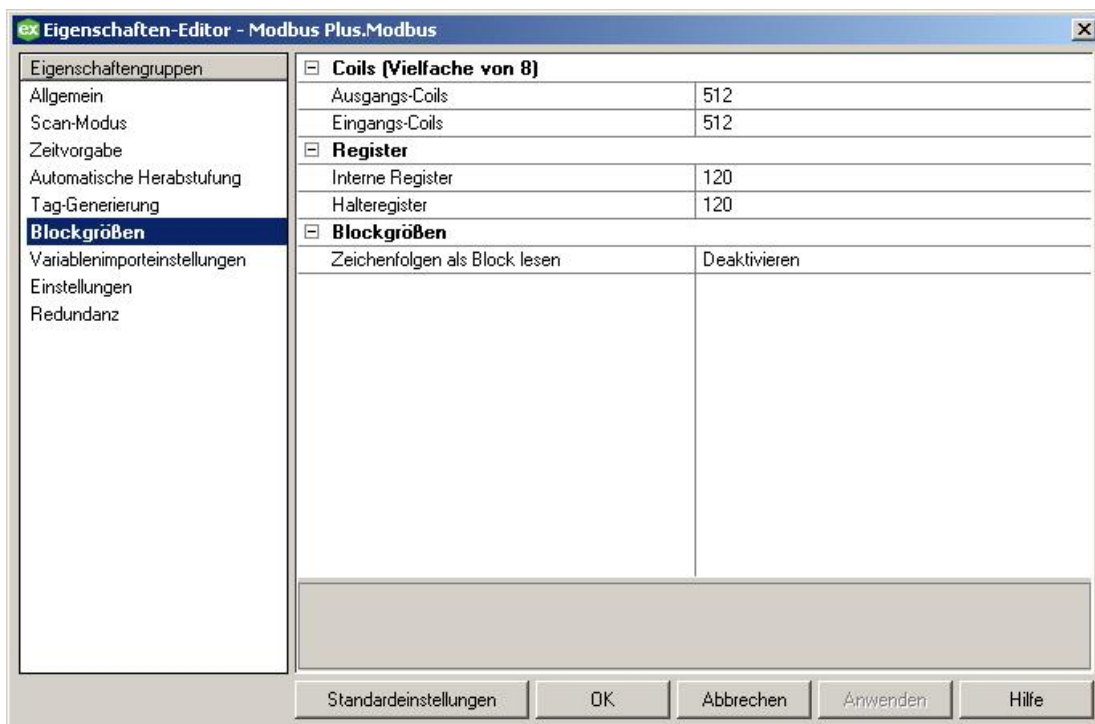
Automatisch generierte Untergruppen zulassen: Mit dieser Eigenschaft wird gesteuert, ob der Server automatisch Untergruppen für die automatisch generierten Tags erstellt. Dies ist die Standardeinstellung. Wenn diese Option deaktiviert ist, generiert der Server die Tags des Geräts in einer unstrukturierten Liste ohne jede Gruppierung. Im Serverprojekt werden die resultierenden Tags mit dem Adresswert benannt. Beispielsweise werden die Tag-Namen während des Generierungsprozesses nicht beibehalten.

● **Hinweis:** Wenn beim Generieren von Tags durch den Server einem Tag derselbe Name wie einem bestehenden Tag zugewiesen wird, erhöht das System automatisch auf die nächste höchste Nummer, sodass der Tag-Name nicht dupliziert wird. Beispiel: Wenn der Generierungsprozess das Tag "AI22" erstellt, das bereits existiert, wird stattdessen das Tag als "AI23" erstellt.

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

● **Hinweis:** **Tags erstellen** ist deaktiviert, wenn die Konfiguration ein Projekt offline bearbeitet.

Geräteeigenschaften - Blockgrößen



Coils

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

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

● **Hinweise:**

1. Coil-Größe muss ein Vielfaches von 8 sein.
2. Für MBX, NETLIB oder NONE beträgt der Standardwert 512 und der Maximalwert 2000.
3. Für HILSCHER beträgt der Standardwert und der Maximalwert 248.
4. Diese Eigenschaft ist deaktiviert, wenn der Client verbunden ist.


Register

Interne Register: Gibt die Blockgröße für interne Register in Bit an. Bis zu 125 Modbus-Standardregister (16 Bit) können auf einmal gelesen werden.

Halteregister: Gibt die Blockgröße für Halteregister in Bit an. Bis zu 125 Modbus-Standardregister (16 Bit) können auf einmal gelesen werden.

Hinweise:

1. Für MBX, NETLIB oder NONE beträgt der Standardwert 120 und der Maximalwert 125.
2. Für HILSCHER beträgt der Standardwert und der Maximalwert 95.
3. Diese Eigenschaft ist deaktiviert, wenn der Client verbunden ist.
4. Für ein TIO-Modul verwenden Sie diese Einstellung, um den Treiber darüber zu informieren, wie viele Byte zurückgegeben werden, wenn Datenposition 400001 gelesen wird. Für Module, die 2 Byte zurückgeben, setzen Sie dies auf 1. Für Module, die 3 Byte zurückgeben, setzen Sie dies auf 2. Der Treiber verwendet (unabhängig von dieser Einstellung) feste Blocklängen für alle anderen Datenpositionen.
5. Möglicherweise unterstützt das Gerät keine Block-Lesevorgänge/-Schreibvorgänge in der Standardgröße. Kleinere Modicon-SPS und Nicht-Modicon-Geräte unterstützen möglicherweise nicht die vom MBPlus-Netzwerk unterstützten maximalen Datenübertragungslängen.
6. Das Gerät kann nicht-zusammenhängende Adressen enthalten. Wenn dies der Fall ist und der Treiber versucht, einen Datenblock zu lesen, der einen nicht definierten Arbeitsspeicher umfasst, weist das Gerät wahrscheinlich die Anfrage zurück.

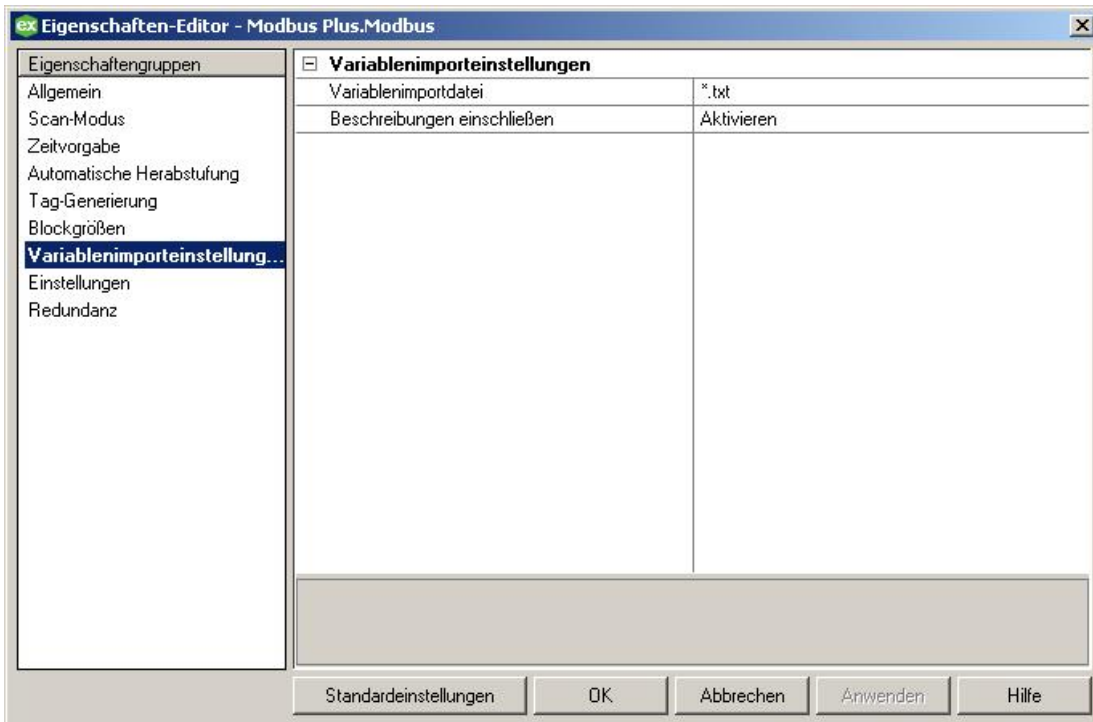
 Achtung: Wenn die Blockgrößen auf einen Wert über 120 festgelegt werden und ein 32- oder 64-Bit-Datentyp für Tags verwendet wird, kann der Fehler "Ungültige Adresse im Block" auftreten. Verringern Sie die Blockgröße auf den Wert 120, damit der Fehler nicht auftritt.

Blockgrößen

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

Geräteeigenschaften - Variablenimporteinstellungen

 Weitere Informationen zu CSV-Dateien für Modbus-Treiber finden Sie unter [CSV-Dateien für Kepware Modbus-Treiber erstellen](#).



Variablenimporteinstellungen

Variablenimportdatei: Gibt den genauen Speicherort der Variablenimportdatei an, die der Treiber verwenden soll, wenn die automatische Tag-Datenbankgenerierung für dieses Gerät aktiviert ist.

Beschreibungen einschließen: Aktivieren Sie diese Option, um importierte Tag-Beschreibungen zu verwenden (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](#).

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

Geräteeigenschaften - Redundanz

Eigenschaftengruppen	Redundanz	
Allgemein	Pfad des Sekundärgeräts	
Scan-Modus	Betriebsmodus	Fehler beim Einschalten
Zeitvorgabe	Überwachungselement	
Redundanz	Überwachungsintervall (s)	300
	Baldmöglichste Rückkehr zum Primärgerät	Ja

Redundanz steht mit dem Plugin für Redundanz auf Medienebene zur Verfügung.

• Weitere Informationen dazu erhalten Sie auf der Website, von einem Vertriebsrepräsentanten oder im Benutzerhandbuch.

Automatische Tag-Datenbankgenerierung

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

Variablenimportdatei erstellen

Die Importdatei muss im durch Semikola getrennten TXT-Format vorliegen, dem Standard-Exportdateiformat der Geräteprogrammieranwendung Concept. Mit der Geräteprogrammieranwendung ProWORX können ebenfalls Variablendaten in diesem Format exportiert werden.

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

Serverkonfiguration

Die automatische Tag-Datenbankgenerierung kann so angepasst werden, dass sie den Anforderungen der Anwendung entspricht. Die primären Steuerungsoptionen können entweder während der Datenbankerstellung über den Assistenten oder in den Geräteeigenschaften festgelegt werden.

• *Weitere Informationen dazu finden Sie in der Hilfedokumentation zum Server.*

Modbus-Plus-Treiber erfordert weitere Einstellungen zusätzlich zu den grundlegenden Einstellungen, die für alle Treiber gelten, die die automatische Tag-Datenbankgenerierung unterstützen. Zu diesen speziellen Einstellungen gehört, dass Name und Speicherort der Variablenimportdatei erforderlich sind. Diese Informationen können in den Geräteeigenschaften unter Variablenimporteinstellungen angegeben werden.

• *Weitere Informationen finden Sie unter [Variablenimporteinstellungen](#).*

Vorgang

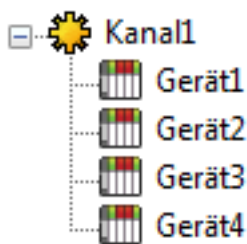
Je nach spezifischer Konfiguration wird die Tag-Generierung automatisch gestartet, sobald das Serverprojekt geöffnet wird, oder sie muss zu einem anderen Zeitpunkt manuell initiiert werden. Das Ereignisprotokoll zeigt an, wann der Tag-Generierungsprozess gestartet wurde, ob während der Verarbeitung der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

Kommunikation optimieren

Die folgenden Optimierungen gelten nur für die SA85-Karte. Hilscher-CIF-Kartenkonfigurationen unterstützen nur 1 Kanal pro Adapter.

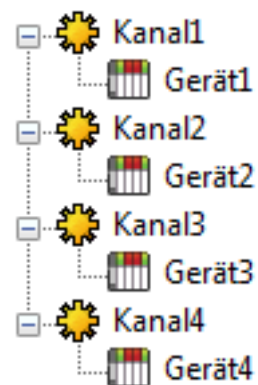
Modbus-Plus-Treiber wurde entwickelt, um einen besseren Durchsatz zu ermöglichen und vollständig Nutzen aus der SA85-Karte zu ziehen. Zuvor waren Benutzer von Modbus-Plus-Treiber darauf beschränkt, einen einzelnen Kanal im Serverprojekt zu definieren, und alle Modbus-Plus-Geräte, auf die zugegriffen werden sollte, mussten unter diesem Kanal definiert werden. Das bedeutete, dass der Treiber zwischen den jeweiligen Geräten bewegt werden musste, um Anfragen zu stellen. Da der OPC-Server bereits auf Effizienz ausgelegt war, reichte die Leistung mit dem Schema einzelner Kanäle für die meisten Anwendungen aus. Mit dem Aufkommen von OPC als unterstützender Technologie hat sich jedoch die Größe von Projekten erheblich erhöht. Um ein hohes Leistungsniveau aufrechtzuerhalten, ist Modbus-Plus-Treiber für ein hohes Niveau an Effizienz und Leistung konzipiert.

● **Hinweis:** Bevor Sie mit diesen Änderungen beginnen, sichern Sie das Server-Projektverzeichnis, um bei Bedarf zu den vorherigen Einstellungen zurückkehren zu können.



In diesem Projekt ist nur ein Kanal definiert. Alle Geräte, auf die zugegriffen werden muss, sind unter diesem einen Kanal definiert. Modbus-Plus-Treiber muss sich so schnell wie möglich 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 das Aktualisierungsintervall.

Die neueste Version von Modbus-Plus-Treiber verwendet mehrere Kanaldefinitionen, um die Leistung der Anwendung zu steigern. In dieser Konfiguration stellt jeder Kanal im Server einen separaten Ausführungspfad dar. Indem bis zu 8 zusätzliche Kanäle hinzugefügt werden, wird die Arbeitsbelastung der Anwendung über die neuen Kanäle verteilt. Dadurch werden mehrere Ausführungspfade erstellt, die unabhängig ausgeführt werden, und es ergibt sich eine erhebliche Steigerung der Leistung. In der folgenden Abbildung wird dieselbe Anwendung angezeigt, nachdem sie für die Verwendung mehrerer Kanäle neu konfiguriert wurde.



Jedes Gerät wurde jetzt unter seinem eigenen Kanal festgelegt. In dieser neuen Konfiguration kann der OPC-Server einen einzelnen Ausführungspfad dediziert für die Aufgabe des Sammelns von Daten von einem einzelnen Gerät einsetzen, weil jedes über seinen eigenen dedizierten Kanal verfügt. Wenn die Anwendung über 8 oder weniger Geräte verfügt, kann sie optimiert werden wie hier gezeigt.

Selbst wenn die Anwendung über mehr als 8 Geräte verfügt, gibt es einen Vorteil. Zwar sind 8 oder weniger Geräte ideal, jedoch zieht die Anwendung weiterhin Nutzen aus zusätzlichen Kanälen. Obwohl das bedeutet, dass sich der Server innerhalb eines gegebenen Kanals von Gerät zu Gerät bewegen muss, kann dies jetzt mit weit weniger zu bearbeitenden Geräten auf einem einzigen Kanal erfolgen.

● **Hinweis:** Der Grenzwert von 8 Kanälen entspricht den vom Hersteller festgelegten Mehrpfad-Beschränkungen der SA85- und Hilscher-Karte.

Die Anwendung kann neu konzipiert werden, um mehrere Kanäle zu unterstützen, selbst wenn unter jedem Gerät eine große Anzahl von Tags definiert ist. Weitere Informationen dazu entnehmen Sie den folgenden Anweisungen.

1. Klicken Sie im vorhandenen, Einzelkanal-basierten Projekt auf **Konnektivität | Neuer Kanal**, und benennen Sie den Kanal wie gewünscht.
2. Schneiden Sie das Gerät **PLC2** aus dem Kanal **ModbusPlus** aus.
3. Fügen Sie es unter dem neuen Kanal ein. Mithilfe der Funktionen zum Ausschneiden und Einfügen können Sie schnell die Anwendung ändern, um Nutzen aus dem neuen Modbus-Plus-Treiber zu ziehen.

In diesen Beispielen werden die ersichtlichsten Optimierungen hervorgehoben, die mit Modbus-Plus-Treiber möglich sind. Zu den weiteren möglichen Optimierungen gehört der Einsatz eines dedizierten

einzelnen Kanals nur für globale Daten. Definieren Sie hierzu für jedes Gerät einen neuen Satz von Gerätenamen mit globalen Daten, auf die unter diesem neuen Kanal zugegriffen werden soll. Denken Sie daran, nur über diese neu definierten Gerätenamen auf globale Daten zuzugreifen.

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 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](#)

[TIO-Moduladressierung](#)

Modbus-Adressierung

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

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 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	Word , Short, BCD Float, DWord, Long, LBCD	Schreibgeschützt*	04
	300001-365535		Schreibgeschützt*	04
	300001-365533		Schreibgeschützt*	04
	3xxxxx.0/1-3xxxxx.15/16**	Boolean	Schreibgeschützt*	04
	300001.2H-365536.240H***	Zeichenfolge	Schreibgeschützt	04
	300001.2L-365536.240L***	String	Schreibgeschützt	04
Halteregister	400001-465536	Word , Short, BCD Float, DWord, Long, LBCD	Lesen/Schreiben	03, 06, 16
	400001-465535		Lesen/Schreiben	03, 06, 16
	400001-465533		Lesen/Schreiben	03, 06, 16
	4xxxxx.0/1-4xxxxx.15/16**	Boolean	Lesen/Schreiben	03, 06, 16, 22
	400001.2H-465536.240H***	Zeichenfolge	Lesen/Schreiben	03, 16
	400001.2L-465536.240L***	String	Lesen/Schreiben	03, 16

Adresstyp	Bereich	Datentyp	Zugriff	Funktionscodes
Globale Daten	G01-G32 G01-G31 G01-G29	Word , Short, BCD Float, DWord, Long, LBCD Double	Lesen/Schreiben Lesen/Schreiben Lesen/Schreiben	k.A. k.A. k.A.
	Gxx.0/1-Gxx.15/16**	Boolean	Schreibgeschützt	k.A.

*Bei Slave-Geräten ist für diese Positionen das Lesen/Schreiben möglich.

**Weitere Informationen finden Sie unter "Vergleich von nullbasierter mit eins-basierter Adressierung" unter [Einstellungen](#).

***.Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte.

Modbus-Adressierung im Hexadezimalformat

Adresstyp	Dezimalbereich	Datentyp	Zugriff
Ausgangs-Coils	H000001-H010000	Boolean	Lesen/Schreiben
Eingangs-Coils	H100001-H110000	Boolean	Schreibgeschützt
Interne Register	H300001-H310000 H300001-H30FFFF H300001-H30FFFD	Word , Short, BCD Float, DWord, Long, LBCD Double	Schreibgeschützt* Schreibgeschützt* Schreibgeschützt*
	H3yyyyy.0/1-H3yyyyy.F/10	Boolean	Schreibgeschützt*
	H300001.2H-H3FFFF.240H	Zeichenfolge	Schreibgeschützt
	H300001.2L-H3FFFF.240L	Zeichenfolge	Schreibgeschützt
Halteregister	H400001-H410000 H400001-H40FFFF H400001-H40FFFD	Word , Short, BCD Float, DWord, Long, LBCD Double	Lesen/Schreiben Lesen/Schreiben Lesen/Schreiben
	H4yyyyy.0/1-H4yyyyy.F/10	Boolean	Lesen/Schreiben
	H400001.2H-H4FFFF.240H	Zeichenfolge	Lesen/Schreiben
	H400001.2L-H4FFFF.240L	String	Lesen/Schreiben
Globale Daten	HG01-HG20 HG01-HG1F HG01-HG1D	Word , Short, BCD Float, DWord, Long, LBCD Double	Lesen/Schreiben Lesen/Schreiben Lesen/Schreiben
	HGyy.0/1-HGyy.F/10	Boolean	Schreibgeschützt

*Bei Slave-Geräten ist für diese 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 nur verfügbar, wenn sich das Modbus-Modell im Master-Modus befindet. Die Syntax lautet folgendermaßen:

Ausgangs-Coils: 0xxxxx#nn

Eingangs-Coils: 1xxxxx#nn

Hierbei gilt Folgendes:

- xxxxx steht für die Adresse der ersten Coil. Sowohl Dezimal- als auch Hexadezimalwerte sind zulässig.
- nn steht für die Anzahl der Coils, die in einen analogen Wert gepackt werden. Der gültige Bereich liegt zwischen -1 und 16, und nur Dezimalwerte sind zulässig.

● **Hinweis:** Der einzige gültige Datentyp ist Word. Ausgangs-Coils haben Lese-/Schreibzugriff, wohingegen Eingangs-Coils schreibgeschützten Zugriff haben. 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 "Lesegeschützt" 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 Haltereister 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. Weitere Informationen finden Sie unter [Postfach-Modus](#).

● **Hinweis:** Der Datentyp Double wird nicht unterstützt.

Zeichenfolgenunterstützung

Das Modbus-Modell unterstützt das Lesen und Schreiben im Haltereisterspeicher als ASCII-Zeichenfolge. Bei Verwendung von Haltereister 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.

Beispiele

- 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.
- 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 bei Verwendung eines Zeichenfolgen-Tags die Fehlermeldung "In Adresse <Adresse> auf dem Gerät <Gerät> kann nicht geschrieben werden: Gerät hat mit Ausnahmecode 3 geantwortet." im Server-Ereignisfenster empfangen wird, wurde die Länge der Zeichenfolge vom Gerät abgelehnt. Benutzer sollten die Zeichenfolge nach Möglichkeit kürzen.

Array-Unterstützung

Arrays werden für interne und Haltereister-Positionen 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 Haltereister:

4xxx [Zeilen] [Spalten]

4xxx [Spalten] mit einer angenommenen Zeilenanzahl von 1.

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

● **Hinweis:** Die Basisadresse für globale Daten darf den Wert 32 nicht überschreiten.

Beschreibung von Funktionscodes

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

Gerät für globale Datenkommunikation konfigurieren

Globale Daten werden von SA85- und Hilscher-CIF-Schnittstellenkarten unterstützt. Ein Zugriff kann nur von einem einzelnen Netzwerk erfolgen. Beispielsweise kann mit "7.0.0.0.0" auf globale Daten zugegriffen werden, mit "7.1.0.0.0" jedoch nicht.

● **Hinweis:** Der unaufgeforderte Modus unterstützt keine globalen Daten.

Globale Daten in ein Gerät schreiben

Aus der SPS-Perspektive lautet die Host-PC-Adresse 2.0.0.0.0. Aus der Host-PC-Perspektive lautet die SPS-Adresse 9.0.0.0.0. Dies entspricht dem Geräte-ID-Pfad. Benutzer müssen in der Programmiersoftware die Adressen konfigurieren, in die das Gerät lesen und von denen es schreiben kann.

Steuerungsblock

Register	Inhalt	Beschreibung
Steuerung [1]	5	Funktionscode zum Schreiben globaler Daten
Steuerung [2]	- 0 = kein Fehler	Der Fehlercode. Kann nicht geändert werden.
Steuerung [3]	32	Die Anzahl von Wörtern, die vom statischen RAM in den globalen Arbeitsspeicher geschrieben werden kann. Die maximale Anzahl beträgt 32 Bit.
Steuerung [4]	-	Reserviert*
Steuerung [5]	2	Die Modbus-Plus-Knotenadresse, an die Daten gesendet werden.
Steuerung [6]	0	Pfad zum Host-PC.
Steuerung [7]	0	Pfad zum Host-PC.
Steuerung [8]	0	Pfad zum Host-PC.
Steuerung [9]	0	Pfad zum Host-PC.

*Dieses Register ist anwendungsspezifisch.

Datenbereich

Register	Inhalt	Beschreibung
Datenfeld [1]-Datenfeld [32]	Daten	k.A.

Globale Daten vom Gerät lesen

Aus der SPS-Perspektive lautet die Host-PC-Adresse 2.0.0.0.0. Aus der Host-PC-Perspektive lautet die SPS-Adresse 9.0.0.0.0. Dies entspricht dem Geräte-ID-Pfad.

Steuerungsblock

Register	Inhalt	Beschreibung
Steuerung [1]	6	Funktionscode zum Lesen globaler Daten
Steuerung [2]	- 0 = kein Fehler	Der Fehlercode. Kann nicht geändert werden.
Steuerung [3]	32	Die Anzahl von Wörtern, die vom statischen RAM in den globalen Arbeitsspeicher geschrieben werden kann. Die maximale Anzahl beträgt 32 Bit.
Steuerung [4]	-	Reserviert*
Steuerung [5]	2	Die Modbus-Plus-Knotenadresse, von der Daten gelesen werden.
Steuerung [6]	0	Pfad zum Host-PC.
Steuerung [7]	0	Pfad zum Host-PC.
Steuerung [8]	0	Pfad zum Host-PC.
Steuerung [9]	0	Pfad zum Host-PC.

*Dieses Register ist anwendungsspezifisch.

Datenbereich

Register	Inhalt	Beschreibung
Datenfeld [1]-Datenfeld [32]	Daten	k.A.

TIO-Moduladressierung

Der Postfach-Modus wird für dieses Modell nicht unterstützt.

TIO-Moduladressierung im Dezimalformat

Adresstyp	Bereich	Datentyp	Zugriff
Daten-E/A*	400001 400001.0/1-400001.15/16**	Word, Short Boolean	Lesen/Schreiben Lesen/Schreiben
Dateneingang - Eingeklinkt	400257 400257.0/1-400257.15/16**	Word, Short Boolean	Schreibgeschützt Schreibgeschützt
Modul-Timeout	461441 461441.0/1-461441.15/16**	Word, Short Boolean	Lesen/Schreiben Lesen/Schreiben
Modulstatus	463489-463497 4xxxxx.0/1-4xxxxx.15/16**	Word, Short Boolean	Schreibgeschützt Schreibgeschützt
Modul-ASCII-Header	464513	String	Schreibgeschützt

*Der von einer Daten-E/A-Position gelesene Wert kommt aus dem Eingangsregister des Moduls. Beim Schreiben in diese Position ändert der Wert, der gesendet wird, das Ausgangsregister des Moduls. Deshalb entspricht der an dieser Position gelesene Wert nicht dem zuvor in diese Position geschriebenen Wert.

**Weitere Informationen finden Sie unter "Vergleich von nullbasierter mit eins-basierter Adressierung" unter [Einstellungen](#).

TIO-Moduladressierung im Hexadezimalformat

Adresstyp	Bereich	Datentyp	Zugriff
Daten-E/A*	H40001 H40001.0/1-H40001.F/10	Word, Short Boolean	Lesen/Schreiben Lesen/Schreiben
Dateneingang - Eingeklinkt	H40101 H40101.0/1-40101.F/10	Word, Short Boolean	Schreibgeschützt Schreibgeschützt
Modul-Timeout	H4F001 H4F001.0/1-H4F001. F/10	Word, Short Boolean	Lesen/Schreiben Lesen/Schreiben
Modulstatus	H4F801-H4F809 H4yyyy.0/1-H4yyyy. F/10	Word, Short Boolean	Schreibgeschützt Schreibgeschützt
Modul-ASCII-Header	H4FC01	String	Schreibgeschützt

*Der von einer Daten-E/A-Position gelesene Wert kommt aus dem Eingangsregister des Moduls. Beim Schreiben in diese Position ändert der Wert, der gesendet wird, das Ausgangsregister des Moduls. Deshalb entspricht der an dieser Position gelesene Wert nicht dem zuvor in diese Position geschriebenen Wert.

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).

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

Fehlertyp:

Fehler

Mögliche Ursache:

Es wurde versucht, einen nicht vorhandenen Speicherplatz auf dem angegebenen Gerät zu referenzieren.

Mögliche Lösung:

Überprüfen Sie die Adressen aller dem Gerät zugewiesenen Tags, und entfernen Sie die Adressen, die ungültige Speicherorte referenzieren.

Ungültige Adresse im Block. | Blockbereich = H<Start> bis H<Ende>.

Fehlertyp:

Fehler

Mögliche Ursache:

Es wurde versucht, einen nicht vorhandenen Speicherplatz auf dem angegebenen Gerät zu referenzieren.

Mögliche Lösung:

Überprüfen Sie die Adressen aller dem Gerät zugewiesenen Tags, und entfernen Sie die Adressen, die ungültige Speicherorte referenzieren.

Gerät MBPLUS.SYS konnte nicht gestartet werden.

Fehlertyp:

Fehler

Mögliche Ursache:

Der Treiber für MBPLUS.SYS ist nicht ordnungsgemäß konfiguriert.

Mögliche Lösung:

Vergewissern Sie sich, dass das MBPLUS-Gerät manuell gestartet und gestoppt werden kann, indem Sie das Applet unter 'Systemsteuerung' | 'Geräte' verwenden. Wenn der Treiber MBPLUS.SYS manuell gestartet wird, kann der Treiber modbus_unsolicited.dll den Treiber starten.

Karte konnte nicht festgestellt werden, oder Modbus Plus-Dienste konnten nicht gestartet werden. Vergewissern Sie sich, dass die Karte und die Treiber *.sys von MBP ordnungsgemäß installiert sind.

Fehlertyp:

Fehler

Systemressourcen zum Ausführen dieses Treibers konnten nicht erstellt werden.

Fehlertyp:

Fehler

Kanal konnte nicht initialisiert werden.

Fehlertyp:

Fehler

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

Fehlertyp:

Fehler

Mögliche Ursache:

Das 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.

Kanal konnte nicht geladen werden. Nur ein Kanal pro Hilscher-Adapter ist zulässig. Ändern Sie das Projekt so, dass jeder Kanal über einen eindeutigen Adapter verfügt, und laden Sie neu.

Fehlertyp:

Fehler

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

Fehlertyp:

Fehler

Fehler beim Öffnen des MBPLUS-Pfads. | Pfad = '<Pfad>'.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Der Treiber für MBPLUS.SYS ist nicht ordnungsgemäß konfiguriert.
2. Der Treiber kann keinen Pfad im angegebenen Adapter öffnen.

Mögliche Lösung:

1. Folgen Sie den Anweisungen für die Installation und Konfiguration des MBPLUS-Treibers.
2. Vergewissern Sie sich, dass höchstens acht Kanäle der gleichen Adapternummer zugewiesen sind.

Empfangene Blocklänge stimmt nicht mit erwarteter Länge überein. | Empfangene Länge = <Anzahl> (Byte), Erwartete Länge = <Anzahl> (Byte).

Fehlertyp:

Warnung

Globale Daten nicht verfügbar von Gerät.

Fehlertyp:

Warnung

Fehler beim Lesen der globalen Daten vom Gerät.

Fehlertyp:

Warnung

Blockanfrage auf Gerät hat mit Ausnahme geantwortet. | Blockbereich = <Start> bis <Ende>, Ausnahme = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

Der angeforderte Knoten antwortet nicht.

Mögliche Lösung:

Überprüfen Sie Verkabelung, Schaltung und Pins.

• **Siehe auch:**

Hilscher CIF-Ausnahmecodes

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

Fehlertyp:

Warnung

Mögliche Ursache:

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

Mögliche Lösung:

Siehe 'Modbus-Ausnahmecodes'.

• **Siehe auch:**

Modbus-Ausnahmecodes

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

Fehlertyp:

Warnung

Mögliche Ursache:

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

Mögliche Lösung:

Siehe 'Modbus-Ausnahmecodes'.

• **Siehe auch:**

Modbus-Ausnahmecodes

Blockadressenanfrage hat mit Ausnahme geantwortet. | Blockbereich = H<Start> bis H<Ende>, Ausnahme = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

Der angeforderte Knoten antwortet nicht.

Mögliche Lösung:

Überprüfen Sie Verkabelung, Schaltung und Pins.

Siehe auch:

Hilscher CIF-Ausnahmecodes

Warnung: Globale Daten sind deaktiviert. Zugriff erfordert Systemtreiber auf niedriger Ebene von Modicon 4.0.

Fehlertyp:

Warnung

Adapter kann nicht geöffnet werden. | Adapter = <Name>.

Fehlertyp:

Warnung

Tag-Import wegen zu wenig Speicherressourcen fehlgeschlagen.

Fehlertyp:

Warnung

Mögliche Ursache:

Der Treiber kann den für die Verarbeitung der Variablenimportdatei erforderlichen Speicherplatz nicht zuweisen.

Mögliche Lösung:

Schließen Sie alle nicht benötigten Anwendungen, und versuchen Sie es erneut.

Beim Tag-Import ist eine Dateiausnahme aufgetreten.

Fehlertyp:

Warnung

Mögliche Ursache:

Die Variablenimportdatei konnte nicht gelesen werden.

Mögliche Lösung:

Regenerieren Sie die Variablenimportdatei.

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

Fehlertyp:

Warnung

Mögliche Ursache:

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

Mögliche Lösung:

Ändern Sie in der Variablenimportdatei das Feld, das den Fehler verursacht.

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

Fehlertyp:

Warnung

Mögliche Ursache:

Die Tag-Beschreibung im angegebenen Datensatz ist zu lang.

Mögliche Lösung:

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

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

Fehlertyp:

Warnung

Mögliche Ursache:

Der in der Variablenimportdatei gefundene Tag-Name enthält ungültige Zeichen.

Mögliche Lösung:

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

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

Fehlertyp:

Warnung

Mögliche Ursache:

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

Mögliche Lösung:

Ändern Sie den Datentyp in der Variablenimportdatei 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 auf Gerät kann nicht geschrieben werden. Platine hat mit Ausnahme
geantwortet. | Adresse = '<Adresse>', Ausnahme = <Code>'.**

Fehlertyp:

Warnung

Mögliche Ursache:

1. Der Adapter ist möglicherweise nicht vorhanden.
2. Abhängig vom angegebenen Fehlercode.

Mögliche Lösung:

Vergewissern Sie sich, dass die richtige Adapternummer in den Kanaleigenschaften gewählt wurde. Verwenden Sie SyCon, um die Adapterreihenfolge festzustellen.

● Hinweis:

Gilt nicht für die SA85-Karte. Code -1, -33 für die Hilscher CIF-Karte.

• Siehe auch:

SyCon-Benutzerhandbuch

Von Adresse auf Gerät kann nicht gelesen werden. Platine hat mit Ausnahme geantwortet. | Adresse = '<Adresse>', Ausnahme = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Der Adapter ist möglicherweise nicht vorhanden.
2. Abhängig vom angegebenen Fehlercode.

Mögliche Lösung:

Vergewissern Sie sich, dass die richtige Adapternummer in den Kanaleigenschaften gewählt wurde. Verwenden Sie SyCon, um die Adapterreihenfolge festzustellen.

● **Hinweis:**

Gilt nicht für die SA85-Karte. Code -1, -33 für die Hilscher CIF-Karte.

● **Siehe auch:**

SyCon-Benutzerhandbuch

Gerät MBPLUS.SYS wurde gestartet

Fehlertyp:

Informationen

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

Fehlertyp:

Informationen

Hilscher CIF Ausnahmecodes

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

CIF-Code	Name	Bedeutung
111	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.
114	SLAVE DEVICE FAILURE	Ein nicht wiederherstellbarer Fehler ist aufgetreten, während der Server (oder Slave) versucht hat, die angeforderte Aktion auszuführen.

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

Modbus Ausnahmecodes

Die unten dargestellten 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 (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

Code dezi- mal/hexadezimal	Name	Bedeutung
		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. Bedeutet normalerweise, dass das Gerät im Netzwerk nicht vorhanden ist.

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

Index

5

5-stellige Adressierung 27

6

6-stellige Adressierung 27

A

Abrufen 6

Adapter 9

Adapter kann nicht geöffnet werden. | Adapter = <Name>. 36

Adaptornummer 10

Adressbeschreibungen 27

Alle Datenanfragen im Scan-Intervall 16

Alle Werte für alle Tags schreiben 8

Anfangsaktualisierungen aus Cache 16

Anforderungs-Timeout 17

Anfragen verwerfen, wenn herabgestuft 18

Angefordert 6, 12

Array-Unterstützung 29

Ausgangs-Coils 20, 27

Ausnahmecodes 38-39

Automatische Tag-Datenbankgenerierung 23

B

BCD 26

Bei doppeltem Tag 19

Bei Eigenschaftsänderung 19

Bei Gerätestart 19

Beim Tag-Import ist eine Dateiausnahme aufgetreten. 36

Beschreibung 11

Beschreibung für Datensatz in Importdatei abgeschnitten. | Datensatznummer = <Nummer>. 36

Beschreibung von Funktionscodes 30

Beschreibungen einschließen 22

Blockadressenanfrage hat mit Ausnahme geantwortet. | Blockbereich = H<Start> bis H<Ende>, Ausnahme = <Code>. 35

Blockanfrage auf Gerät hat mit Ausnahme geantwortet. | Blockbereich = <Start> bis <Ende>, Ausnahme = <Code>. 35

Blockgrößen 20
Boolean 26

D

Data Master 12
Data Slave 12
Datenanfrage nicht schneller als Scan-Intervall 16
Datenbankerstellung 23
Datensammlung 12
Datentypbeschreibung 26
Dezimalformat 27, 30-31
Diagnose 8
Double 26
Durch Tag angegebenes Scan-Intervall berücksichtigen 16
DWord 26

E

Eingangs-Coils 20, 27
Eingeklinkt 31
Einzelkanal 25
Einzelnes Netzwerk 14
Elternteilgruppe 20
Empfangene Blocklänge stimmt nicht mit erwarteter Länge überein. | Empfangene Länge = <Anzahl> (Byte),
Erwartete Länge = <Anzahl> (Byte). 34
Ereignisprotokollmeldungen 33
Erstellen 20
Erweiterte Kanaleigenschaften 9
Externe Abhängigkeiten 5

F

Fehler beim Lesen der globalen Daten vom Gerät. 34
Fehler beim Öffnen der Datei für Tag-Datenbank-Import. | Betriebssystemfehler = '<Fehler>'. 34
Fehler beim Öffnen des MBPLUS-Pfads. | Pfad = '<Pfad>'. 34
Fehler beim Parsen von Datensatz in Importdatei. | Datensatznummer = <Nummer>, Feld = <Anzahl>. 36
Float 26
Force Multiple Coils 30
Force Single Coil 30
Funktionscodes 27

G

Generieren 19
Gepackte Coils 28
Gerät für globale Datenkommunikation konfigurieren 30
Gerät MBPLUS.SYS konnte nicht gestartet werden. 33
Gerät MBPLUS.SYS wurde gestartet 38
Geräte-ID (SPS-Netzwerksadresse) 12
Geräteeigenschaften 11
Geräteeigenschaften - Automatische Herabstufung 17
Geräteeigenschaften - Tag-Generierung 18
Gleitkommazahl 26
Globale Daten 30
Globale Daten nicht verfügbar von Gerät. 34

H

Halteregister 21, 27
Herabstufen bei Fehler 18
Herabstufungszeitraum 18
Hexadezimalformat 28, 32
Hexadezimalwert 30
Hilscher 13
Hilscher-CIF-Ausnahmecodes 38
Hilscher-CIF-Karte 5-6
Hilscher SyCon 5

I

ID 11
Identifikation 11
IEEE-754-Gleitkomma 9
Importierter Tag-Name ist ungültig und wurde geändert. | Tag-Name = '<Tag>', geänderter Tag-Name = '<Tag>'. 37
In Adresse auf Gerät kann nicht geschrieben werden. Gerät hat mit Ausnahme geantwortet. | Adresse = '<Adresse>', Ausnahme = <Code>. 35
In Adresse auf Gerät kann nicht geschrieben werden. Platine hat mit Ausnahme geantwortet. | Adresse = '<Adresse>', Ausnahme = <Code>. 37
Interne Register 21, 27

K

Kanal konnte nicht geladen werden. Nur ein Kanal pro Hilscher-Adapter ist zulässig. Ändern Sie das Projekt so, dass jeder Kanal über einen eindeutigen Adapter verfügt, und laden Sie neu. 34

Kanal konnte nicht initialisiert werden. 34
Kanaleigenschaften 7
Kanaleigenschaften - Allgemein 7
Kanaleigenschaften - Schreiboptimierungen 8
Kanalzuweisung 11
Karte konnte nicht festgestellt werden, oder Modbus Plus-Dienste konnten nicht gestartet werden. Vergewissern Sie sich, dass die Karte und die Treiber *.sys von MBP ordnungsgemäß installiert sind. 33
Kommunikation optimieren 24
Kommunikations-Timeouts 16-17
Konzept 23

L

LBCD 26
Leistung 24
Lesegeschützter Zugriff 29
Long 26
Löschen 19

M

Masked Write Register 30
Master 12
MBPLUS 5
MBX 5
Mehrere Kanäle 24
Mit Vorzeichen 26
Modbus-Adressierung 27
Modbus-Anwendungsprotokoll 38
Modbus-Ausnahmecodes 39
Modell 11
Modicon 5
Modicon SA85-Netzwerkkarte 5
MSTR 13
MSTR-Anweisung 15

N

Name 11
Netzwerk mit Bridge 15
Nicht normalisierte Float-Handhabung 9
Nicht scannen, nur Abruf anfordern 16
Nur den letzten Wert für alle Tags schreiben 8

Nur den letzten Wert für nicht boolesche Tags schreiben 8

O

Ohne Vorzeichen 26

Optimierungsmethode 8

P

PCI-85 5

Postfach 6, 12

Postfach-Modus 13, 29

Preset Multiple Registers 30

Preset Single Register 30

Projekt 24

ProWORX 23

ProWORX-Geräteprogrammieranwendung 23

R

Read Coil Status 30

Read Holding Registers 30

Read Input Status 30

Read Internal Registers 30

Redundanz 22

S

SA8 5

SA85-Karte 6, 24

Scan-Modus 16

Schneider 5

Schnittstellenkarten 6

Schreiboptimierungen 8

Servicezyklus 8

Setup 6

Short 26

Simuliert 12

Slave 12

Slave-Pfad (unaufgefordert) 12

Steuerungsblock 15

String 26

Systemressourcen zum Ausführen dieses Treibers konnten nicht erstellt werden. 33

T

- Tag-Datenbank wird importiert. | Quelldatei = '<Dateiname>'. 38
- Tag-Generierung 18
- Tag-Import wegen zu wenig Speicherressourcen fehlgeschlagen. 36
- Tag konnte nicht importiert werden, da der Datentyp nicht unterstützt wird. | Tag-Name = '<Tag>', nicht unterstützter Datentyp = '<Typ>'. 37
- Timeout bis zum Herabstufen 18
- TIO-Modul 21
- TIO-Moduladressierung 31
- Treiber 7, 11

U

- Überschreiben 19
- Übersicht 5
- Unaufgefordert 6
- Ungültige Adresse im Block. | Blockbereich = <Start> bis <Ende>. 33
- Ungültige Adresse im Block. | Blockbereich = H<Start> bis H<Ende>. 33
- Ungültiges Array. | Array-Bereich = <Start> bis <Ende>. 34
- Untergruppen zulassen 20
- Unterstützt 6

V

- Variablenimportdatei 22-23
- Variablenimporteinstellungen 21
- Verbindungs-Timeout 17
- Versuche vor Timeout 17
- Verzögerung zwischen Anfragen 17
- Vom Client angegebenes Scan-Intervall berücksichtigen 16
- Von Adresse auf Gerät kann nicht gelesen werden. Gerät hat mit Ausnahme geantwortet. | Adresse = '<Adresse>', Ausnahme = <Code>. 35
- Von Adresse auf Gerät kann nicht gelesen werden. Platine hat mit Ausnahme geantwortet. | Adresse = '<Adresse>', Ausnahme = <Code>. 38

W

- Warnung
 - Globale Daten sind deaktiviert. Zugriff erfordert Systemtreiber auf niedriger Ebene von Modicon 4.0. 36
- Word 26

Z

Zeichenfolgen als Block lesen 21

Zeichenfolgenunterstützung 29