

DF1-Treiber von Allen-Bradley

© 2018 PTC Inc. Alle Rechte vorbehalten.

Inhaltsverzeichnis

DF1-Treiber von Allen-Bradley	1
Inhaltsverzeichnis	2
DF1-Treiber von Allen-Bradley	5
Übersicht	5
Kabelverbindungen	6
Kanal-Setup	7
Kanaleigenschaften - Allgemein	7
Kanaleigenschaften - Serielle Kommunikation	8
Kanaleigenschaften - Schreiboptimierungen	10
Kanaleigenschaften - Erweitert	11
Kanaleigenschaften - Kommunikationsserialisierung	12
Kanaleigenschaften - Verknüpfungseinstellungen	13
Halbduplex-Master	14
KF2/KF3-Halbduplex-Master	15
Geräte-Setup	17
Geräteeigenschaften - Allgemein	17
Geräteeigenschaften - Scan-Modus	19
Geräteeigenschaften - Ethernet-Kapselung	19
Geräteeigenschaften - Zeitvorgabe	20
Geräteeigenschaften - Automatische Herabstufung	21
Geräteeigenschaften - Protokolleinstellungen	22
Gleitkommazahl-Wörter	23
Funktionsdateioptionen	23
Steckplatzkonfiguration	25
Richtlinien für modulare E/A-Auswahl	26
Geräteeigenschaften - Redundanz	28
Datentypbeschreibung	29
Adressbeschreibungen	30
Ausgabedateien	30
Eingabedateien	33
Statusdateien	37
Binäre Dateien	37
Zeitgeberdateien	38
Zählerdateien	39
Steuerdateien	39
Ganzzahldateien	40
Float-Dateien	41
ASCII-Dateien	41
String-Dateien	42
BCD-Dateien	43
Long-Dateien	43
MicroLogix-PID-Dateien	44

PLC5-PID-Dateien	45
MicroLogix-Meldungsdateien	46
PLC5-Meldungsdateien	47
Blocktransferdateien	48
HSC-Datei (High Speed Counter, Hochgeschwindigkeitszähler)	49
RTC-Datei (Real Time Clock, Echtzeituhr)	50
Kommunikationsstatusdatei für Kanal 0 (CS0, Channel 0 Communication Status File)	51
Kommunikationsstatusdatei für Kanal 1 (CS1, Channel 1 Communication Status File)	51
IOS-Datei (I/O Module Status, E/A-Modulstatus)	52
Ereignisprotokollmeldungen	53
Block kann nicht auf Gerät gelesen werden. Blockanfang = '<Adresse>', Status = <Code>, erweiterter Status = <Code>.	53
Block kann nicht auf Gerät gelesen werden. Block deaktiviert. Blockanfang = '<Adresse>', Status = <Code>, erweiterter Status = <Code>.	53
Block kann nicht auf Gerät gelesen werden. Blockanfang = '<Adresse>', Status = <Code>, erweiterter Status = <Code>.	53
Block kann nicht auf Gerät gelesen werden. Block deaktiviert. Blockanfang = '<Adresse>', Status = <Code>, erweiterter Status = <Code>.	54
Funktionsdatei auf Gerät kann nicht gelesen werden. Funktionsdatei = '<Funktionsdateielement>', Status = <Code>, erweiterter Status = <Code>.	54
Funktionsdatei auf Gerät kann nicht gelesen werden. Block deaktiviert. Funktionsdatei = '<Funktionsdateielement>', Status = <Status>, erweiterter Status = <Status>.	54
Block kann nicht auf Gerät gelesen werden. Framing-Fehler. Blockanfang = '<Adresse>'.	55
Funktionsdatei auf Gerät kann nicht gelesen werden. Framing-Fehler. Funktionsdatei = '<Funktionsdateielement>'.	55
Block kann nicht auf Gerät gelesen werden. Prüfsummenfehler. Blockanfang = '<Adresse>'.	55
Funktionsdatei auf Gerät kann nicht gelesen werden. Prüfsummenfehler. Funktionsdatei = '<Funktionsdateielement>'.	55
Block kann nicht auf Gerät gelesen werden. Slave-Senke/Quelle ist voll. Blockanfang = '<Startadresse>'.	56
Funktionsdatei auf Gerät kann nicht gelesen werden. Slave-Senke/Quelle voll. Funktionsdatei = '<Funktionsdateielement>'.	56
Block kann nicht auf Gerät gelesen werden. Slave-Quelle leer. Blockanfang = '<Adresse>'.	56
Funktionsdatei auf Gerät kann nicht gelesen werden. Slave-Quelle leer. Funktionsdatei = '<Funktionsdateielement>'.	56
Fehler beim Schreiben in Adresse. Tag-Adresse = '<Adresse>', Status = <Status>, erweiterter Status = <Status>.	57
Fehler beim Schreiben in Adresse. Framing-Fehler. Tag-Adresse = '<Adresse>'.	57
Prüfsummenfehler beim Schreiben in Adresse. Tag-Adresse = '<Adresse>'.	57
Fehler beim Schreiben in Adresse. Slave-Senke/Quelle voll. Tag-Adresse = '<Adresse>'.	58
Fehler beim Schreiben in Adresse. Slave-Quelle leer. Tag-Adresse = '<Adresse>'.	58
Timeout von Gerät beim Schreiben in Adresse. Tag-Adresse = '<Adresse>'.	58
Block kann nicht auf Gerät gelesen werden. Blockanfang = '<Adresse>', Status = <Code>, erweiterter Status = <Code>.	58
Block kann nicht auf Gerät gelesen werden. Block deaktiviert. Blockanfang = '<Adresse>', Status = <Code>, erweiterter Status = <Code>.	59
Block kann nicht auf Gerät gelesen werden. Gerät hat mit NAK geantwortet. Blockanfang = '<Adresse>'.	59

Funktionsdatei auf Gerät kann nicht gelesen werden. Gerät hat mit NAK geantwortet. Funktionsdatei = '<Funktionsdateielement>'.	59
Block kann nicht auf Gerät gelesen werden. Speicherzuordnungsfehler. Blockanfang = '<Adresse>'.	59
Funktionsdatei auf Gerät kann nicht gelesen werden. Speicherzuordnungsfehler. Funktionsdatei = '<Funktionsdateielement>'.	59
Funktionsdatei auf Gerät kann nicht gelesen werden. Gerät hat unerwarteterweise mit NAK geantwortet. Überprüfen Sie das Verknüpfungsprotokoll des Geräts. Funktionsdatei = '<Funktionsdateielement>'.	60
Block kann nicht auf Gerät gelesen werden. Gerät hat unerwarteterweise mit NAK geantwortet. Überprüfen Sie das Verknüpfungsprotokoll des Geräts. Blockanfang = '<Adresse>'.	60
In Adresse auf Gerät kann nicht geschrieben werden. Paketlänge liegt außerhalb des zulässigen Bereichs. Tag-Adresse = '<Adresse>', Paketlängenbereich = <Min.> bis <Max.> (Byte).	60
In Adresse auf Gerät kann nicht geschrieben werden. TNS liegt außerhalb des zulässigen Bereichs. Tag-Adresse = '<Adresse>', TNS-Bereich = <Min.> bis <Max.>.	60
Fehlermaskendefinitionen	61
Appendix: Communicating with RSLogix5000 Family Controllers	62
Index	64

DF1-Treiber von Allen-Bradley

Hilfeversion 1.056

INHALT

Übersicht

Was ist DF1-Treiber von Allen-Bradley?

Kanal-Setup

Wie konfiguriere ich einen Kanal für die Verwendung mit diesem Treiber?

Geräte-Setup

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

Datentypbeschreibung

Welche Datentypen werden von diesem Treiber unterstützt?

Adressbeschreibungen

Wie adressiere ich eine Datenposition auf einem DF1-Gerät von Allen-Bradley?

Ereignisprotokollmeldungen

Welche Fehlermeldungen werden durch DF1-Treiber von Allen-Bradley erzeugt?

Übersicht

DF1-Treiber von Allen-Bradley bietet eine zuverlässige Möglichkeit, DF1-Geräte von Allen-Bradley mit Client-Anwendungen, u.a. HMI, SCADA, Historian, MES, ERP und zahlreichen benutzerdefinierten Anwendungen, zu verbinden. Dieser Treiber unterstützt SPS der Micrologix-, SLC500- und PLC5-Reihe von Allen-Bradley.

Kabelverbindungen

Bei diesem Treiber handelt es sich um einen seriellen DF1-RS232-Treiber von Allen-Bradley. Er verarbeitet nicht die Token-Ring-Übergabe eines DH-485- oder DH+-Netzwerks. Überprüfen Sie beim Testen der Kabelverbindungen für diesen Treiber zuerst die Kommunikation mit der Programmiersoftware von Allen-Bradley.

- Wenn für SPS der Micrologix- und SLC500-Reihe der ausgewählte APS-Treiber "KF3/KE", "Voll duplex", Halbduplex-Slave" oder "Voll duplex (Micro)" lautet und die Kommunikation mit der SPS verfügbar ist, kann dieser Treiber mit denselben Verkabelungs- und Kommunikationseinstellungen kommunizieren.
- Wenn für SPS der PLC5-Reihe der ausgewählte Treiber für die PLC-5-Programmiersoftware für SPS oder KE/KF "seriell" ist und die Kommunikation mit der SPS verfügbar ist, kann dieser Treiber mit denselben Verkabelungs- und Kommunikationseinstellungen kommunizieren.

MicroLogix-Reihe

Verwenden Sie dasselbe Kabel wie bei Verwendung der APS-Software von Allen-Bradley.

SLC500-Reihe - Direkte Verbindung

Wenn die SPS über einen RS232-Port verfügt, kann sie direkt mit einem RS232-Nullmodem-Standardkabel (dabei handelt es sich um dasselbe Kabel, das für die APS-Software von Allen-Bradley verwendet wird) angeschlossen werden. Der SPS-Port muss für Allen-Bradley-DF1-Kommunikation und nicht für DH-485-Master konfiguriert werden.

● **Hinweis:** Dieser Treiber funktioniert nicht bei einer direkten Verbindung mit dem DH-485-Port einer SPS der SLC500-Reihe, die einen 1747-PIC-Konverter wie bei der APS-Software verwendet.

PLC5-Reihe - Direkte Verbindung

Eine direkte Verbindung kann über den CH0-Port erweiterter PLC5-Prozessoren mit einem RS232-Nullmodem-Standardkabel hergestellt werden. Der Port muss für Allen-Bradley-DF1-Kommunikation konfiguriert werden.

DH-485-Netzwerke

KF3 von Allen-Bradley oder ein kompatibles Gerät ist erforderlich, um den Treiber mit dem DH-485-Netzwerk zu verbinden. Der PC wird über ein Nullmodem-Standardkabel mit dem KF3-Gerät verbunden.

DH+-Netzwerke

Folgende drei Optionen sind für die Kommunikation mit einem Gerät im DH+-Netzwerk mithilfe von DF1-Treiber von Allen-Bradley möglich:

- KF2 von Allen-Bradley oder ein kompatibles Gerät. Der PC wird über ein Nullmodem-Standardkabel mit dem KF2-Gerät verbunden.
- DataLink DL-Schnittstellenkarten (PCI/ISA/PC104). Informationen zur DH+-Verkabelung finden Sie in der AB-Dokumentation.
- Ethernet-to-DH+-Konverter für DataLink DL4500. Informationen zur Verkabelung finden Sie in der Dokumentation zu DL4500.

Kanal-Setup

Unterstützte Verknüpfungsprotokolle

Allen-Bradley DF1 Vollduplex (Point-to-Point-Kommunikation)

Allen-Bradley DF1-Halbduplex-Master (Multi-Drop-Kommunikation), auch als Allen-Bradley DF1-Abfragemodus bekannt.*

Allen-Bradley DF1-Funkmodem (Point-to-Point- und Multi-Drop-Kommunikation).**

*Kommunikation von Slave zu Slave wird nicht unterstützt.

**Speicher- und Weiterleitungsfunktion wird nicht unterstützt.

● **Hinweis:** Informationen zu erforderlichen Firmware-Versionen für Allen-Bradley DF1-Funkmodemunterstützung finden Sie unter [Geräte-Setup](#).

Kanaleigenschaften

[Allgemein](#)

[Serielle Kommunikation](#)

[Schreiboptimierung](#)

[Erweitert](#)

[Kommunikationsserialisierung](#)

[Verknüpfungseinstellungen](#)

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

Identifikation

Name: Benutzerdefinierte ID dieses Kanals. Bei jedem Serverprojekt muss jeder Kanalname eindeutig sein. Zwar können Namen bis zu 256 Zeichen lang sein, doch haben einige Client-Anwendungen beim Durchsuchen des Tag-Raums des OPC-Servers ein eingeschränktes Anzeigefenster. Der Kanalname ist ein Teil der OPC-Browserinformationen.

● **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.

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

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

Kanaleigenschaften - Serielle Kommunikation

Eigenschaften für serielle Kommunikation stehen seriellen Treibern zur Verfügung und sind je nach Treiber, Verbindungstyp und ausgewählten Optionen unterschiedlich. Unten finden Sie eine Übermenge der möglichen Eigenschaften.

Klicken Sie, um zu einem der Abschnitte zu springen: [Verbindungstyp](#), [Serielle Port-Einstellungen](#) bzw. [Ethernet-Einstellungen](#) und [Betriebsverhalten](#).

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

Eigenschaftengruppen		
Allgemein		
Serielle Kommunikation		
Schreiboptimierungen		
Erweitert		
Kommunikationsserialisierung		
Verknüpfungseinstellungen		
	<input type="checkbox"/> Verbindungstyp	
	Physisches Medium	COM-Port
	Gemeinsam genutzt	Nein
	<input type="checkbox"/> Serielle Port-Einstellungen	
	COM-ID	3
	Baudrate	19200
	Daten-Bits	8
	Parität	Keine
	Stopp-Bits	1
	Flusssteuerung	Keine
	<input type="checkbox"/> Betriebsverhalten	
	Bericht Kommunikationsfehler	Aktivieren

Verbindungstyp

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

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

Serielle Port-Einstellungen

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

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


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

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

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

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


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

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

Betriebsverhalten

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

Ethernet-Einstellungen

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

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

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

Modemeinstellungen

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

Operation ohne Kommunikation

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

Kanaleigenschaften - Schreiboptimierungen

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

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 Schreibenforderungen 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		

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

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

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

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

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

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

Kanaleigenschaften - Kommunikationsserialisierung

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

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

Eigenschaftengruppen	☐ Einstellungen auf Kanalebene	
Allgemein	Virtuelles Netzwerk	Keine
Serielle Kommunikation	Transaktionen pro Zyklus	1
Schreiboptimierungen	☐ Globale Einstellungen	
Erweitert	Netzwerkmodus	Lastausgleich
Kommunikationsserialisier...		

Einstellungen auf Kanalebene

Virtuelles Netzwerk Mit dieser Eigenschaft wird der Modus der Kommunikationsserialisierung des Kanals festgelegt. Zu den Optionen gehören "Keine" sowie "Netzwerk 1 - Netzwerk 500". Die Standardeinstellung ist "Keine". Es folgen Beschreibungen der Optionen:

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

Transaktionen pro Zyklus Mit dieser Eigenschaft wird die Anzahl einzelner blockierter/nicht blockierter Lese-/Schreibtransaktionen festgelegt, die auf dem Kanal vorkommen können. Wenn einem Kanal die Gelegenheit zur Kommunikation gegeben wird, wird diese Anzahl von Transaktionen versucht. Der gültige Bereich liegt zwischen 1 und 99. Die Standardeinstellung ist 1.

Globale Einstellungen

- **Netzwerkmodus:** Mit dieser Eigenschaft wird gesteuert, wie die Kanalkommunikation delegiert wird. Im Modus **Lastausgleich** wird jedem Kanal die Möglichkeit gegeben, nacheinander zu kommunizieren. Im Modus **Priorität** wird Kanälen die Möglichkeit gegeben, nach den folgenden Regeln (von der höchsten zur niedrigsten Priorität) zu kommunizieren:
 - Kanäle mit ausstehenden Schreibvorgängen haben den höchsten Vorrang.
 - Kanäle mit ausstehenden expliziten Lesevorgängen (durch interne Plugins oder externe Client-Schnittstellen) werden je nach Priorität des Lesevorgangs priorisiert.
 - Gescannte Lesevorgänge und andere periodische Ereignisse (treiberspezifisch).

Die Standardeinstellung ist "Lastausgleich" und wirkt sich auf *alle* virtuellen Netzwerke und Kanäle aus.

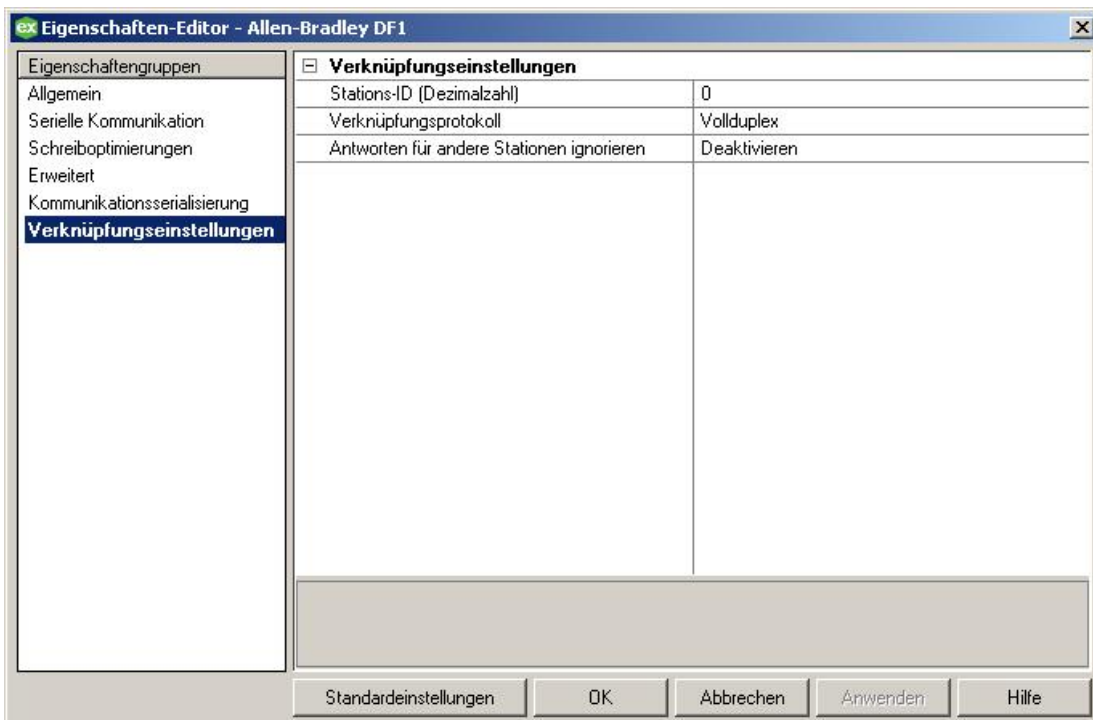
🔴 Geräte, die sich auf unaufgeforderte Antworten verlassen, sollten nicht in ein virtuelles Netzwerk eingefügt werden. In Situationen, wo die Kommunikationen serialisiert werden muss, wird empfohlen, dass "Automatische Herabstufung" aktiviert wird.

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

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

Kanaleigenschaften - Verknüpfungseinstellungen

Greifen Sie unter "Kanaleigenschaften" auf "Verknüpfungseinstellungen" zu.



Stations-ID: Die Stations-ID basiert auf dem Gerät, mit dem kommuniziert wird (ausschließlich Funkmodems):

- Wenn sich das Zielgerät in einem DH+- oder DH-485-Netzwerk befindet, muss die Kommunikation über Serial-to-DH+/DH-485-Konverter (d.h. KF2/KF3-Modul) erfolgen. In diesem Fall ist das Gerät, mit dem kommuniziert wird, der Konverter und nicht das Zielgerät selbst (Micrologix, SLC500 oder PLC-5). Die Stationsnummer für diese Konfiguration sollte auf die Knotenadresse des Converters eingestellt werden.
- Wenn sich das Zielgerät nicht in einem DH+- oder DH-485-Netzwerk befindet, ist das Gerät, mit dem kommuniziert wird, eine Micrologix-, SLC500- oder PLC-5-SPS. Für die Stationsnummer dieser Konfiguration kann eine beliebige eindeutige Adresse festgelegt werden.
- Wenn sich das Zielgerät in einer DH+- oder DH-485-Konverterkonfiguration befindet, ist die Stationsnummer gleich der Knotenadresse des Converters (z.B. KF2/KF3-Knotenadresse).
- Wenn sich das Zielgerät in einer seriellen Standardkonfiguration befindet, ist die Stations-ID eine beliebige eindeutige Adresse im Netzwerk für den lokalen PC. Der Bereich für DH-485 liegt zwischen 1 und 63. Ansonsten liegt er zwischen 0 und 255.

Verknüpfungsprotokoll: Folgende Verknüpfungsprotokolle werden unterstützt:

- **Vollduplex** (auch als Allen-Bradley DF1 bekannt): bei Punkt-zu-Punkt-Verknüpfung verwendet, was eine bidirektionale Kommunikation mit hoher Leistungsfähigkeit zwischen Peers ermöglicht.
- **Halbduplex-Master** (auch als Abfragemodus bekannt): ist ein Multi-Drop-Protokoll mit einem Master und mindestens einem Slave. *Weitere Informationen dazu finden Sie unter [Halbduplex-Master](#).*
- **KF2/KF3-Halbduplex-Master** ist ein Multi-Drop-Protokoll mit einem Master und mindestens einem Slave, was einen niedrigeren Datendurchsatz als bei Vollduplex ermöglicht. Jedoch besteht die Möglichkeit, mit mehreren KF2/KF3-Modulen über einen COM-Port zu kommunizieren. *Weitere Informationen dazu finden Sie unter [KF2/KF3-Halbduplex-Master](#).*
- **Funkmodem:** ein Befehls-/Antwortprotokoll ohne ACKs oder NAKs während des Anforderungs-/Antwortvorgangs. Dadurch wird die Anzahl Byte verringert, die Funkmodems zur Durchführung einer Transaktion senden und empfangen. Dieses Protokoll unterstützt Vollduplex-Kommunikation über eine Punkt-zu-Punkt-Verknüpfung, was eine bidirektionale Kommunikation mit hoher Leistungsfähigkeit zwischen Peers ermöglicht. Es wird auch Master/Slave-Kommunikation unterstützt, was Multi-Drop-Konfigurationen ermöglicht. Bei der Leistung werden Vollduplex- und Halbduplex-Protokolle übertroffen.

Slave-Abbruchverzögerung (ms): Der Treiber wurde so optimiert, dass er Master-Meldungen und Abrufe zur Erhöhung des Datendurchsatzes möglichst schnell sendet. Der Slave-Erstabruf wird nicht verzögert, weil keine Verzögerung erforderlich ist. Wenn der Slave Zeit zur Verarbeitung von Anforderungen benötigt, ist dies in der Erstabrufantwort offensichtlich. Daher sorgt der Treiber für eine Verzögerung und ruft den Slave erneut ab. Die Verzögerung lässt dem Slave Zeit, die Anforderung vor dem nächsten Abruf zu verarbeiten.

Antworten von anderen Stationen ignorieren: Aktivieren Sie diese Option, um das Annehmen von Antworten auf jene für die im Feld Stations-ID angegebene Station zu beschränken.

● **Hinweis:** Die zur Verfügung stehenden Eigenschaften und Optionen sind je nach ausgewählten Optionen unterschiedlich.

Halbduplex-Master

Halbduplex-Protokoll ist ein Multi-Drop-Protokoll mit einem Master und mindestens einem Slave. Im Allgemeinen ermöglicht "Halbduplex" einen niedrigeren Datendurchsatz als "Vollduplex", bietet jedoch die Flexibilität, über einen einzigen COM-Port mit mehreren Geräten kommunizieren zu können. "Halbduplex" ist ein Master/Slave-Protokoll. Im Modus "Halbduplex-Master" ist der Treiber der Master und alle Geräte im Netzwerk sind Slaves. Es ist notwendig, dass alle Geräte im Netzwerk als "Halbduplex-Slave" konfiguriert werden, da nur ein Master im Netzwerk zulässig ist.

● *Weitere Informationen zur Konfiguration der Micrologix-, -, SLC500- und PLC5-Geräte mit RSLogix finden Sie in der Rockwell-Dokumentation.*

● **Hinweis:** Wenn sich das Zielgerät in einem DH-485- oder DH+-Netzwerk befindet, muss die Kommunikation jeweils über ein KF2/KF3-Modul erfolgen. Wenn das KF2/KF3-Modul als Halbduplex-Slave konfiguriert wird, muss das Verknüpfungsprotokoll für [KF2/KF3-Halbduplex-Master](#) ausgewählt werden.

Master-Zuständigkeiten und Aktualisierungsintervalle

Der Treiber (Master) ist dafür verantwortlich, die Slaves nach Daten abzufragen. Im Allgemeinen würden Slaves im Round-Robin-Verfahren abgefragt werden. Aufgrund der Art von OPC hängt es vom Aktualisierungsintervall der Slave-Tags ab, wie oft ein Slave abgefragt wird. Auf diese Weise werden Slaves nur abgefragt, wenn ein Lese-/Schreibvorgang von ihnen angefordert wird. Dadurch wird der Datenverkehr im Netzwerk reduziert und nicht benötigte Anforderungen werden verhindert. Der Datenverkehr im Netzwerk wird im Wesentlichen durch den Entwurf des Client-Projekts (insbesondere zugewiesene Aktualisierungsintervalle) bestimmt. Je schneller das Aktualisierungsintervall, desto häufiger wird ein Slave abgefragt.

Meldungen, Senke und Quelle

Es gibt drei Meldungen, die bei einem Lese-/Schreibvorgang zwischen Master und Slave ausgetauscht werden. Die erste Meldung ist die Master-Meldung, die einen Slave zur Durchführung eines Lese-/Schreibvorgangs anfordert. Der Slave antwortet nicht sofort mit Daten wie im Vollduplex-Modus. Die zweite Meldung ist eine Abrufmeldung vom Master zum Slave, in der die beim letzten Master-Meldungsvorgang erfassten Daten angefordert werden. Die dritte Meldung ist die Slave-Antwort mit den in der Master-Meldung angeforderten Daten. Eingehende Anforderungen an den Slave werden in eine sogenannte "Senke" eingefügt. Sobald der Slave die angeforderte Operation durchführt, fügt er das Ergebnis in eine sogenannte "Quelle" ein.

Anzahl der Versuche

Die Anzahl der Versuche für Master-Meldungen und Abrufe entspricht derselben Anzahl der Versuche wie unter Erneute Versuche auf dem Gerät konfiguriert. Diese Versuchsanzahl im Halbduplex-Modus täuscht, da in einer einzigen Datenanforderung mehrere Meldungen vom Master gesendet werden. Für alle Absichten und Zwecke:

Es gilt: $cnAttempts = \text{Erneute Versuche}$

Anzahl (#) Versuche für Master-Meldungs-Timeout = $cnAttempts$

Anzahl (#) Versuche für Abruf-Timeout = $cnAttempts$

Anzahl (#) Versuche für Anforderungs-Timeout = Anzahl (#) Versuche für Master-Meldungs-Timeout + Anzahl (#) Versuche für Abruf-Timeout == $cnAttempts \times 2$

Senke und Quelle voll

Senke und Quelle sind im Wesentlichen Puffer und haben Beschränkungen. Noch wichtiger ist die Möglichkeit für die Senke, mit Anforderungen gefüllt zu werden. Wenn dies geschieht, bestätigt der Slave keine Master-Meldungen, die er erhält. Wenn der Slave nach $cnAttempts$ dem Master keine Bestätigung sendet, ist wahrscheinlich die Slave-Senke voll. Der Treiber fragt anschließend den Slave ab, ob Antworten, über die der Slave möglicherweise verfügt, geleert wurden, wodurch Platz für in der vollen Senke enthaltene Antworten auf Anforderungen geschaffen werden kann. Dieser Abrufvorgang findet so lange statt, bis die Slave-Quelle nach $cnAttempts$ leer ist. Bei der nächsten Slave-Anforderung ist die Senke wahrscheinlich leer. Falls nicht, bedeutet es möglicherweise, dass der Treiber den Slave zu schnell abfragt. Wenn dies der Fall ist, erhöhen Sie die Verzögerung für das erneute Abrufen für den Slave. Ebenfalls kann die Slave-Quelle möglicherweise voll sein und der Treiber fragt den Slave erneut ab, bis die Quelle nach $cnAttempts$ leer ist.

Angenommene und gelöschte Slave-Antworten

In den oben genannten Abschnitten wurde erwähnt, dass der Slave abgefragt wird, bis er geleert ist. Dies ist möglich, wenn die Slave-Quelle mit in der Warteschlange befindlichen Antworten gefüllt ist. Bei jedem gegebenen Abruf wird nur die Antwort auf die letzte Master-Meldung akzeptiert, alle anderen werden gelöscht.

● **Hinweis:** Kommunikation von Slave zu Slave wird nicht unterstützt.

KF2/KF3-Halbduplex-Master

Halbduplex-Protokoll ist ein Multi-Drop-Protokoll mit einem Master und mindestens einem Slave. Im Allgemeinen ermöglicht "Halbduplex" einen niedrigeren Datendurchsatz als "Vollduplex", bietet jedoch die Flexibilität, über einen einzigen COM-Port mit mehreren KF2/KF3-Modulen kommunizieren zu können.

"Halbduplex" ist ein Master/Slave-Protokoll. Im Modus "Halbduplex-Master" ist der Treiber der Master und alle KF2/KF3-Module im Netzwerk sind Slaves. Es ist notwendig, dass alle Geräte im Netzwerk als "Halbduplex-Slave" konfiguriert werden, da nur ein Master im Netzwerk zulässig ist.

● *Weitere Informationen zur Konfiguration des KF2/KF3-Moduls für Halbduplex-Slave-Betrieb finden Sie in der Rockwell-Dokumentation.*

Master-Zuständigkeiten und Aktualisierungsintervalle

Der Treiber (Master) ist dafür verantwortlich, die Slaves nach Daten abzufragen. Im Allgemeinen würden Slaves im Round-Robin-Verfahren abgefragt werden. Aufgrund der Art von OPC hängt es vom Aktualisierungsintervall der Tags des Slaves ab, wie oft ein Slave abgefragt wird. Auf diese Weise werden Slaves nur abgefragt, wenn ein Lese-/Schreibvorgang von ihnen angefordert wird. Dadurch wird der Datenverkehr im Netzwerk reduziert und nicht benötigte Anforderungen werden verhindert. Der Datenverkehr im Netzwerk wird im Wesentlichen durch den Entwurf des Client-Projekts (insbesondere zugewiesene Aktualisierungsintervalle) bestimmt. Je schneller das Aktualisierungsintervall, desto häufiger wird ein Slave abgefragt.

Meldungen, Senke und Quelle

Es gibt drei Meldungen, die bei einem Lese-/Schreibvorgang zwischen Master und Slave ausgetauscht werden. Die erste Meldung ist die Master-Meldung, die einen Slave zur Durchführung eines Lese-/Schreibvorgangs anfordert. Der Slave antwortet nicht sofort mit Daten wie im Vollduplex-Modus. Die zweite Meldung ist eine Abrufmeldung vom Master zum Slave, in der die beim letzten Master-Meldungsvorgang erfassten Daten angefordert werden. Die dritte Meldung ist die Slave-Antwort mit den in der Master-Meldung angeforderten Daten. Eingehende Anforderungen an den Slave werden in eine sogenannte "Senke" eingefügt. Sobald der Slave die angeforderte Operation durchführt, fügt er das Ergebnis in eine sogenannte "Quelle" ein.

Anzahl der Versuche

Die Anzahl der Versuche für Master-Meldungen und Abrufe entspricht derselben Anzahl der Versuche wie unter Erneute Versuche auf dem Gerät konfiguriert. Diese Versuchsanzahl im Halbduplex-Modus täuscht, da in einer einzigen Datenanforderung mehrere Meldungen vom Master gesendet werden. Für alle Absichten und Zwecke:

Es gilt: $cnAttempts = \text{Erneute Versuche}$

Anzahl (#) Versuche für Master-Meldungs-Timeout = $cnAttempts$

Anzahl (#) Versuche für Abruf-Timeout = $cnAttempts$

Anzahl (#) Versuche für Anforderungs-Timeout = Anzahl (#) Versuche für Master-Meldungs-Timeout + Anzahl (#) Versuche für Abruf-Timeout = $cnAttempts \times 2$

Senke und Quelle voll

Senke und Quelle sind im Wesentlichen Puffer und haben Beschränkungen. Noch wichtiger ist die Möglichkeit für die Senke, mit Anforderungen gefüllt zu werden. Wenn dies geschieht, bestätigt der Slave keine Master-Meldungen, die er erhält. Wenn der Slave nach $cnAttempts$ dem Master keine Bestätigung sendet, ist höchstwahrscheinlich die Slave-Senke voll. Der Treiber fragt anschließend den Slave ab, ob Antworten, über die der Slave möglicherweise verfügt, geleert wurden, wodurch Platz für in der vollen Senke enthaltene Antworten auf Anforderungen geschaffen werden kann. Dieser Abrufvorgang findet so lange statt, bis die Slave-Quelle nach $cnAttempts$ leer ist. Bei der nächsten Slave-Anforderung ist die Senke wahrscheinlich leer. Falls nicht, bedeutet es möglicherweise, dass der Treiber den Slave zu schnell abfragt. Wenn dies der Fall ist, erhöhen Sie die Verzögerung für das erneute Abrufen für den Slave. Ebenfalls kann die Slave-Quelle ggf. voll sein und der Treiber fragt den Slave erneut ab, bis die Quelle nach $cnAttempts$ leer ist.

Angenommene und gelöschte Slave-Antworten

In den oben genannten Abschnitten wurde erwähnt, dass der Slave abgefragt wird, bis er geleert ist. Dies ist möglich, wenn die Slave-Quelle mit in der Warteschlange befindlichen Antworten gefüllt ist. Bei jedem gegebenen Abruf wird nur die Antwort auf die letzte Master-Meldung akzeptiert, alle anderen werden gelöscht.

● **Hinweis:** Kommunikation von Slave zu Slave wird nicht unterstützt.

Geräte-Setup

Geräteeigenschaften

[ID](#)

[Betriebsmodus](#)

[Scan-Modus](#)

[Ethernet-Kapselung](#)

[Zeitvorgabe](#)

[Automatische Herabstufung](#)

[Protokolleinstellungen](#)

[Steckplatzkonfiguration](#)

[Funktionsdateioptionen](#)

[Redundanz](#)

Unterstützte Geräte

Micrologix-Reihe*

SLC500-Reihe*

PLC-5-Reihe (ausschließlich der PLC-5/250- und PLC-5/VME-Reihe)

RSLogix5000-Controller mit DF1-Port

*Funkmodem-Verknüpfungsprotokoll erfordert die folgenden Firmware-Upgrades:

SLC 5/03, SLC 5/04 und SLC 5/05: Reihe C FRN6

MicroLogix 1200: Reihe C FRN7

MicroLogix 1500: Reihe C FRN8

DH-485- und DH+-Unterstützung

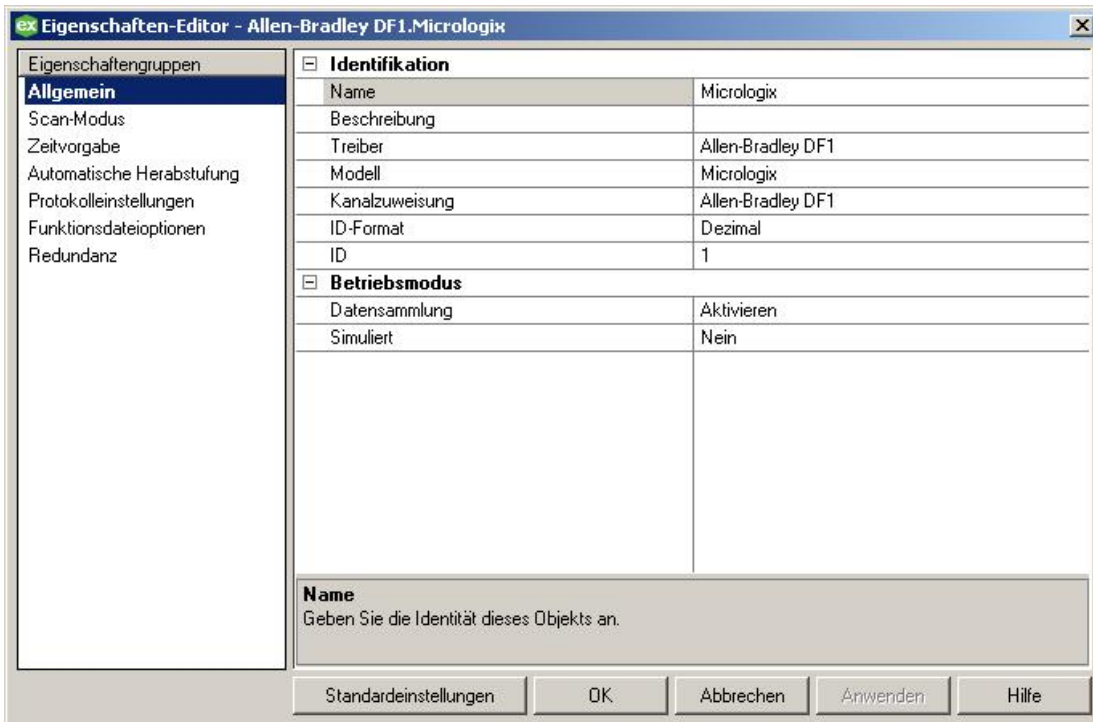
KF3 von Allen-Bradley oder ein kompatibles Gerät ist erforderlich, um den Treiber mit dem DH-485-Netzwerk zu verbinden. Folgende vier Optionen sind für die Kommunikation mit einem Gerät im DH+-Netzwerk möglich:

- KF2 von Allen-Bradley oder ein kompatibles Gerät.
- 1784-U2DHP-USB-Konverter. Dieser Konverter wird dem System als neuer serieller Port angezeigt.
- DataLink DL-Schnittstellenkarten (PCI/ISA/PC104). Diese Karten fügen virtuelle serielle Ports für die nahtlose Konfiguration hinzu.
- Ethernet-to-DH+-Konverter für DataLink DL4500. Konfigurieren Sie das Gerät für Ethernet-Kapselung. NIC ist erforderlich.

• **Siehe auch:**

[Kabelverbindungen](#)

Geräteeigenschaften - Allgemein



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-Format: Wählen Sie aus, wie die Geräte-ID formatiert wird. Zu den Option gehören Dezimal, Oktal und Hex.

ID: Bei der Geräte-ID handelt es sich um die Netzwerkadresse der DF1-SPS von Allen-Bradley. Für SPS in einem DH-485- oder DH+-Netzwerk liegt der Bereich bei 1-63. Ansonsten liegt er bei 0-255.

Betriebsmodus

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

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

Geräteeigenschaften - Scan-Modus

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

Eigenschaftengruppen	☐ Scan-Modus	
Allgemein	Scan-Modus	Vom Client angegebene 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 - Ethernet-Kapselung

Ethernet-Kapselung ist für die Kommunikation mit seriellen Geräten vorgesehen, die im Ethernet-Netzwerk mit Terminalservern verbunden sind. Ein Terminalserver ist im Wesentlichen ein virtueller serieller Port. Er wandelt TCP/IP-Meldungen im Ethernet-Netzwerk in serielle Daten um. Sobald die Meldung in ein serielle Format

umgewandelt wurde, können Benutzer Standardgeräte verbinden, die eine serielle Kommunikation mit dem Terminalserver unterstützen.

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

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

Eigenschaftengruppen	[-] Ethernet-Einstellungen	
Allgemein	IP-Adresse	
Scan-Modus	Port	2101
Ethernet-Kapselung	Protokoll	TCP/IP

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

Port: Diese Eigenschaft wird zur Konfiguration des Ethernet-Ports verwendet, der beim Herstellen der Verbindung mit einem Remote-Terminalserver verwendet wird.

Protokoll: Diese Eigenschaft wird verwendet, um TCP/IP- oder UDP-Kommunikation auszuwählen. Die Auswahl hängt von der Art des verwendeten Terminalservers ab. Die Standardprotokollauswahl ist TCP/IP. Weitere Informationen über verfügbare Protokolle finden Sie in der Hilfedokumentation des Terminalservers.

• Hinweise

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

Geräteeigenschaften - Zeitvorgabe

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

Property Groups	[-] Communication Timeouts	
General	Connect Timeout (s)	3
Scan Mode	Connect Attempts	3
Ethernet Encapsulation	Request Timeout (ms)	1000
Timing	Attempts Before Timeout	3
Auto-Demotion	[-] Timing	
Tag Generation	Inter-Request Delay (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.

Verbindungsversuche: Diese Eigenschaft (die in erster Linie von auf Ethernet-Kapselung basierenden Treibern verwendet wird) beschränkt die Anzahl der Verbindungsversuche zwischen dem Treiber und dem Zielgerät. Ist die Grenze erreicht, ist die Verbindungsanfrage fehlgeschlagen. Die Eigenschaft Verbindungs-Timeout gibt das Zeitintervall zwischen Verbindungsversuchen an. Der gültige Bereich liegt zwischen 1 und 10. Der Standardwert ist 3 Versuche. Wenn diese Einstellung nicht vom Treiber unterstützt wird, ist sie deaktiviert.

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.

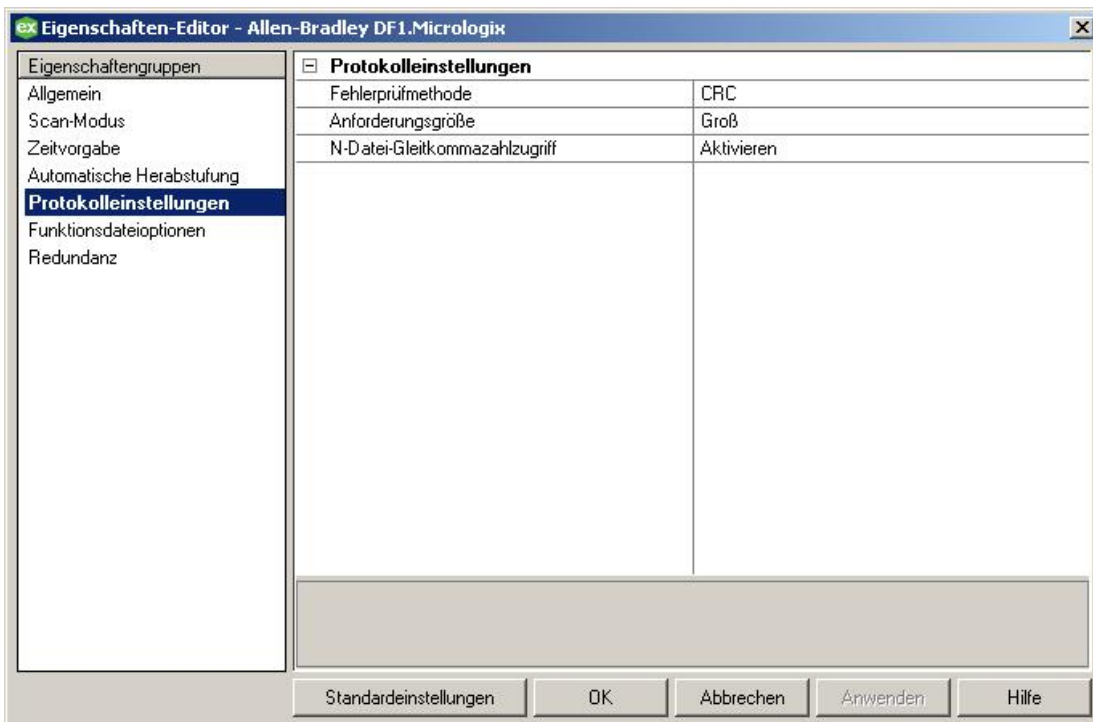
● **Tipp:** Ermitteln Sie, wenn sich ein Gerät im Nicht-Scan-Modus befindet, indem Sie seinen herabgestuften Status mit dem `_AutoDemoted`-System-Tag überwachen.

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 - Protokolleinstellungen



Fehlerprüfmethode: Es stehen zwei Methoden zur Fehlerprüfung zur Verfügung: BCC (Block Check Character) und 16-Bit-CRC (Cyclic Redundancy Check). Wählen Sie die vom Gerät erwartete Prüfsummenmethode, ansonsten antwortet das Gerät nicht.

Gleitkommazahlwörter tauschen PLC-5-Geräte von Allen-Bradley übertragen das High-Wort immer zuerst, gefolgt vom Low-Wort, sodass die Gleitkommazahl-Wörter ausgetauscht werden müssen. Dies ist die Standardeinstellung. Wenn das Gerät das Low-Wort zuerst überträgt, ist für das High-Wort im Paket über die serielle Verbindung kein Wortaustausch erforderlich. *Weitere Informationen dazu finden Sie unter [Gleitkommazahl-Wörter](#).*

Anforderungsgröße: Definieren oder ändern Sie die angeforderte Datenmenge, was zum Verfeinern der Anwendungsleistung wichtig ist. Wenn die Anwendung nacheinander auf große Bereiche des SPS-Speichers zugreift, ist eine große Anforderungsgröße unter Umständen nützlich. Wenn die Daten in der SPS verteilt werden, ist eine kleine Anforderungsgröße unter Umständen nützlich. Die Standardeinstellung ist Groß.

N-Datei-Gleitkommazahlzugriff: Wählen Sie aus, ob der Treiber Gleitkommazahlzugriff auf Ganzzahldateien nativ unterstützt. Die Standardeinstellung ist "Aktivieren".

Gleitkommazahl-Wörter

PLC-5-Gleitkommazahlen richten sich nach dem IEEE 754-Standard. Sie enthalten ein Vorzeichen-Bit S, einen Exponenten E und eine Mantisse M. Das 32-Bit-Layout dieser IEEE 754-Gleitkommazahl sieht wie unten gezeigt aus.

High-Wort Low-Wort
SEEEEEEE EMMMMMMM MMMMMMMM MMMMMMMM
Byte 3 Byte 2 Byte 1 Byte 0

PLC-5-Geräte von Allen-Bradley übertragen binäre Gleitkommazahl-Daten über die serielle Verbindung in der folgenden Reihenfolge:

High-Wort Low-Wort
Byte 2 Byte 3 Byte 0 Byte 1

Dies bedeutet, dass das High-Wort zuerst empfangen wird, gefolgt vom Low-Wort. Aufgrund dieser Anordnung ist ein Austausch der Wörter erforderlich:

Low-Wort High-Wort
Byte 0 Byte 1 Byte 2 Byte 3

Das an den Client weitergeleitete Ergebnis sieht wie folgt aus:

Byte 3 Byte 2 Byte 1 Byte 0

Einige emulierte PLC-5-Geräte (wie Avtron ADDvantage-32) übertragen bereits binäre Gleitkommazahl-Daten über die serielle Verbindung mit dem Low-Wort zuerst.

Low-Wort High-Wort
Byte 0 Byte 1 Byte 2 Byte 3

In diesem Fall muss die Reihenfolge nicht getauscht werden. Das an den Client weitergeleitete Ergebnis sieht wie folgt aus:

Byte 3 Byte 2 Byte 1 Byte 0

Allgemein gilt: Wenn das Gerät das Low-Wort zuerst überträgt, ist für das High-Wort im Paket über die serielle Verbindung kein Wortaustausch erforderlich. Dies gilt nur für emulierte PLC-5-Geräte, also für Geräte, die das DF1-Protokoll von Allen-Bradley mit PLC-5-Befehlen verwenden. PLC-5-Geräte von Allen-Bradley übertragen das High-Wort immer zuerst, gefolgt vom Low-Wort, sodass die Gleitkommazahl-Wörter ausgetauscht werden müssen.

Funktionsdateioptionen



Bei anwendbaren Funktionsdateien können Daten in einem Schreibvorgang in das Gerät geschrieben werden. Wenn Daten in ein Unterelement der Funktionsdatei (Feld innerhalb der Funktionsdateistruktur) geschrieben werden, findet sofort eine Schreiboperation für dieses Tag statt. Für solche Dateien wie die RTC-Datei, deren Unterelemente Stunde (HR), Minute (MIN) und Sekunde (SEC) enthalten, sind einzelne Schreibvorgänge nicht immer akzeptabel. Bei solchen Unterelementen, die sich einzig auf die Zeit verlassen, müssen Werte in einer Operation geschrieben werden, um zu vermeiden, dass Zeit zwischen den Schreibvorgängen der Unterelemente verstreicht. Deshalb gibt es die Option, diese Unterelemente "blockweise zu schreiben".

Anwendbare Funktionsdateien/Unterelemente

RTC	
Jahr	YR
Monat	MON
Tag	DAY
Wochentag	DOW
Stunde	HR
Minute	MIN
Sekunde	SEC

Funktionsweise von Blockschreibvorgängen

Beim Blockschreiben werden die Werte jedes Lesen/Schreiben-Unterelements in der Funktionsdatei in einem einzigen Schreibvorgang in das Gerät geschrieben. Es ist nicht erforderlich, vor der Durchführung eines Blockschreibvorgangs in jedes Unterelement zu schreiben. Für Unterelemente, die nicht davon betroffen sind (geschrieben werden), wird dessen aktueller Wert wieder in sie zurückgeschrieben. Beispiel: Wenn das aktuelle (zuletzt gelesene) Datum und die Uhrzeit der 1.1.2001, 12:00.00, DOW = 3 ist und die Stunde auf 1 Uhr geändert wurde, würden folgende Werte in das Gerät geschrieben: 1.1.2001, 1:00.00, DOW = 3.

Anweisungen

1. Rufen Sie in den Geräteeigenschaften die Funktionsdateioptionen auf. Aktivieren Sie **Blockschreibvorgänge für Funktionsdateien zulassen**. Dadurch wird der Treiber darüber benachrichtigt, Blockschreibvorgänge bei Funktionsdateien zu verwenden, die Blockschreibvorgänge unterstützen. Die Änderungen sind unmittelbar nach dem Klicken auf **OK** oder **Anwenden** wirksam.

- Schreiben Sie den gewünschten Wert in die entsprechenden Unterelement-Tags. Die Unterelement-Tags übernehmen unmittelbar die darin geschriebenen Werte.

● **Hinweis:** Nachdem ein Unterelement mindestens einmal im Blockschreibvorgangsmodus geschrieben wurde, stammt der Wert des Tags nicht vom Controller, sondern aus dem Schreibcache des Treibers. Nachdem der Blockschreibvorgang abgeschlossen ist, stammen alle Tag-Werte des Unterelements vom Controller.

- Sobald die gesamten gewünschten Unterelemente geschrieben wurden, kann ggf. der Blockschreibvorgang durchgeführt werden, bei dem diese Werte an den Controller gesendet werden. Referenzieren Sie zum Instanzieren eines Blockschreibvorgangs die Tag-Adresse **RTC:<Element>._SET**. Das Festlegen dieses Tag-Werts auf "Wahr" hat zur Folge, dass ein Blockschreibvorgang basierend auf den aktuellen (zuletzt gelesenen) Unterelementen und den betroffenen Unterelementen (in die geschrieben wurde) stattfindet. Das _SET-Tag wird als lesegeschütztes Tag behandelt, was bedeutet, dass ein Schreibvorgang in dieses Tag sich nicht in dessen nachfolgenden Lesevorgängen widerspiegelt. Durch Festlegen des Werts dieses Tags auf "Falsch" wird keine Aktion durchgeführt.

Bei Funktionsdateien handelt es sich um strukturbasierte Dateien (die PD- und MG-Datendateien ähnlich sind) und für die Micrologix 1200- und -1500-Geräte eindeutig sind.

Wenn Sie weitere Informationen zu bestimmten Funktionsdateien, die in DF1-Treiber von Allen-Bradley unterstützt werden, erhalten möchten, wählen Sie eine Verknüpfung in der Liste unten aus.

[HSC-Datei \(High Speed Counter, Hochgeschwindigkeitszähler\)](#)

[RTC-Datei \(Real Time Clock, Echtzeituhr\)](#)

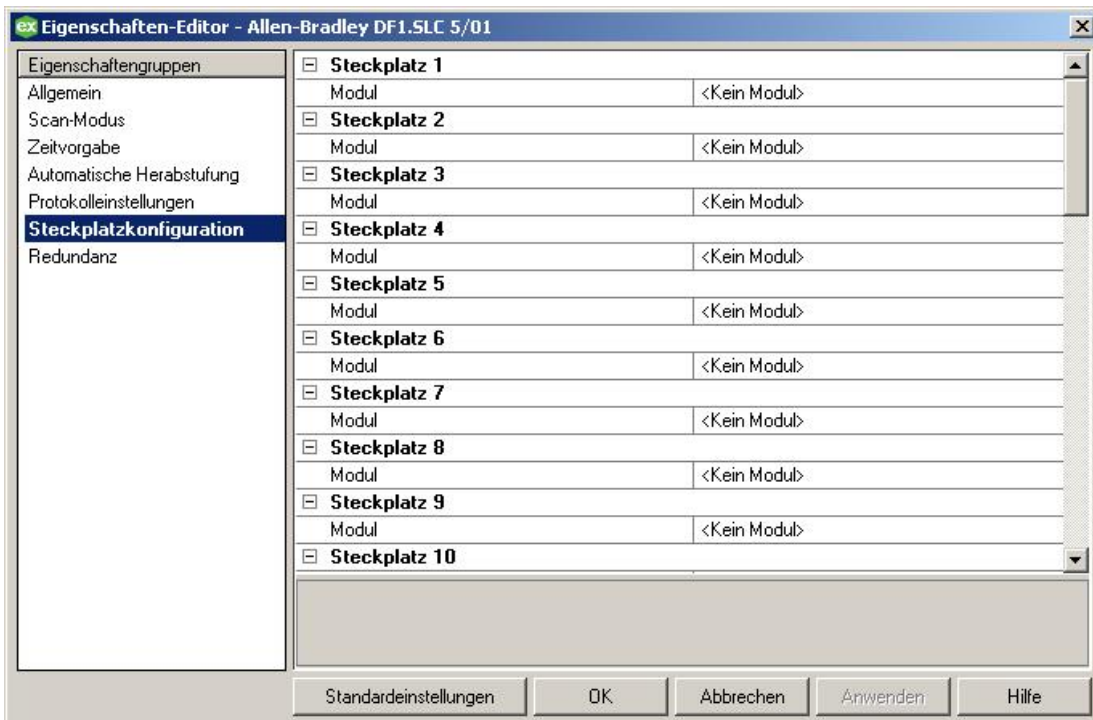
[CS0-Datei \(Channel 0 Communication Status, Kommunikationsstatus für Kanal 0\)](#)

[CS1 \(Channel 1 Communication Status, Kommunikationsstatus für Kanal 1\)](#)

[IOS-Datei \(I/O Module Status, E/A-Modulstatus\)](#)

Steckplatzkonfiguration

SLC 500-Modelle (mit modularen E/A-Racks) müssen für die Verwendung mit diesem Treiber konfiguriert werden, wenn durch den Treiber auf die E/A zugegriffen werden soll. Bis zu 30 Steckplätze können pro Gerät konfiguriert werden.



Wenn Sie die Steckplatzkonfiguration verwenden möchten, beachten Sie die nachstehenden Anweisungen:

1. Wählen Sie den zu konfigurierenden Steckplatz aus, indem Sie im Modul-Listefeld auf die Zeile klicken.
2. Klicken Sie zum Auswählen eines Moduls in der Dropdown-Liste mit den verfügbaren Modulen darauf.
3. Konfigurieren Sie bei Bedarf die Eingabewörter und Ausgabewörter.
4. Wenn Sie einen Steckplatz bzw. ein Modul entfernen möchten, wählen Sie in der Dropdown-Liste mit den verfügbaren Modulen **Kein Modul** aus.
5. Klicken Sie nach Abschluss auf **OK**.

Tip: Mit dem 0000-Generic Module können Sie die nicht in der Liste mit den verfügbaren Modulen enthaltene E/A konfigurieren.

Hinweis: Es ist üblich, offene Steckplätze im Rack zu haben, in denen sich kein physisches Modul befindet. Damit ein korrekter Zugriff auf Daten für die verschiedenen Steckplätze, die ein Modul aufweisen, erfolgen kann, muss den vorhergehenden Modulen die richtige Anzahl von Wörtern zugeordnet sein. Beispiel: Wenn nur Interesse an E/A in Steckplatz 3 besteht, die Steckplätze 1 und 2 jedoch E/A-Module enthalten, müssen von dieser Gruppe zur Steckplatzkonfiguration aus die richtigen Module für die Steckplätze 1, 2 und 3 ausgewählt werden.

0000-Generic Module

Mit dem Generic Module können Sie Eingabe- und Ausgabewörter für Module zuordnen, die in der Liste der verfügbaren Module nicht enthalten sind. Um das Generic Module ordnungsgemäß zu verwenden, müssen Benutzer die Anzahl der für jedes Modul erforderlichen Eingabe- und Ausgabewörter kennen.

• *Schlagen Sie zur Bestätigung von Eingabe- und Ausgabeanforderungen in der E/A-Benutzerhandbuchdokumentation von Allen-Bradley nach und denken Sie daran, dass die Anforderungen je nach Operationen der Klasse 1 oder 3 unterschiedlich sein können.*

• *Weitere Informationen zur Anzahl der für jedes E/A-Modul zur Verfügung stehenden Eingabe- und Ausgabewörter finden Sie unter [Richtlinien für modulare E/A-Auswahl](#).*

Richtlinien für modulare E/A-Auswahl

In der folgenden Tabelle wird die Anzahl der Eingabe- und Ausgabewörter aufgelistet, die für jedes E/A-Modul in der Liste der Steckplatzkonfiguration zur Verfügung stehen.

Tip: Mit dem Generic Module können Sie Eingabe- und Ausgabewörter für Module zuordnen, die in der Liste der verfügbaren Module nicht enthalten sind. Der Bereich der akzeptierten Werte wird in der Tabelle unten angezeigt. Schlagen Sie im Benutzerhandbuch von Allen-Bradley das jeweilige zu konfigurierende E/A-Modul nach, um die Eingabe- und Ausgabeanforderungen zu bestätigen. Die Anforderungen sind je nach Operation der Klasse 1 oder 3 möglicherweise unterschiedlich.

Modultyp	Eingabewörter	Ausgabewörter
0000-Generic Module	0-255	0-255
1203-SM1 - SCANport-Kommunikationsmodul - Einfach	8	8
1203-SM1 - SCANport-Kommunikationsmodul - Erweitert	32	32
1394-SJT - GMC-Turbosystem	32	32
1746-BAS - Basismodul 500 5/01 (Konfiguration)	8	8
1746-BAS - Basismodul 5/02 (Konfiguration)	8	8
1746-HS - Einzelachsen-Bewegungs-Controller	4	4
1746-HSCE - Hochgeschwindigkeitszähler/Encoder	8	1
1746-HSRV - Bewegungssteuerungsmodul	12	8
1746-HSTP1 - Stepper-Controller-Modul	8	8
1746-I*16 - Jedes einzelne 16-pt-Eingabemodul	1	0
1746-I*32 - Jedes einzelne 32-pt-Eingabemodul	2	0
1746-I*8 - Jedes einzelne 8-pt-Eingabemodul	1	0
1746-IA16 - 16 Input 100/120 VAC	1	0

Modultyp	Eingabewörter	Ausgabewörter
1746-IA4 - 4 Input 100/120 VAC	1	0
1746-IA8 - 8 Input 100/120 VAC	1	0
1746-IB16 - 16 Input (Senke) 24 VDC	1	0
1746-IB32 - 32 Input (Senke) 24 VDC	2	0
1746-IB8 - 8 Input (Senke) 24 VDC	1	0
1746-IC16 - 16 Input (Senke) 48 VDC	1	0
1746-IG16 - 16 Input [TTL] (Quelle) 5 VDC	1	0
1746-IH16 - 16 Input [Trans] (Senke) 125 VDC	1	0
1746-IM16 - 16 Input 200/240 VAC	1	0
1746-IM4 - 4 Input 200/240 VAC	1	0
1746-IM8 - 8 Input 200/240 VAC	1	0
1746-IN16 - 16 Input 24 VAC/VDC	1	0
1746-INI4I - Analog 4 Ch. Isol. Stromeingang	8	8
1746-INI4VI - Analog 4 Ch. Isol. Spannungs-/Stromeingang	8	8
1746-INO4I Analog 4 Ch. Isol. Stromeingang	8	8
1746-INO4VI Analog 4 Ch. Isol. Spannungs-/Stromeingang	8	8
1746-INT4 4 Ch. Isolierter Thermoelementeingang	8	8
1746-IO12 - 6 In 100/120 VAC 6 Out [Rly] VAC/VDC	1	1
1746-IO12DC - 6 Input 12 VDC, 6 Output [Rly]	1	1
1746-IO4 - 2 In 100/120 VAC 2 Out [Rly] VAC/VDC3	1	1
1746-IO8 - 4 In 100/120 VAC 4 Out [Rly] VAC/VDC4	1	1
1746-ITB16 - 16 Input [Schnell] (Senke) 24 VDC	1	0
1746-ITV16 - 16 Input [Schnell] (Quelle) 24 VDC	1	0
1746-IV16 - 16 Input (Quelle) 24 VDC	1	0
1746-IV32 - 32 Input (Quelle) 24 VDC	2	0
1746-IV8 - 8 Input (Quelle) 24 VDC	1	0
1746-NI4 - 4 Ch Analoge Eingabe	4	0
1746-NI8 - 8 Ch Analoge Eingabe, Klasse 1	8	8
1746-NI8 - 8 Ch Analoge Eingabe, Klasse 3	16	12
1746-NIO4I - Analog Comb 2 in & 2 Stromausgang	2	2
1746-NIO4V - Analog Comb 2 in & 2 Spannungsausgang	2	2
1746-NO4I - 4 Ch Analoger Stromausgang	0	4
1746-NO4V - 4 Ch Analoger Spannungsausgang	0	4
1746-NR4 - 4 Ch Rtd/Widerstand-Eingangsmodul	8	8
1746-NT4 - 4 Ch Thermoelement-Eingangsmodul	8	8
1746-NT8 Analog 8 Ch - Thermoelementeingang	8	8
1746-O*16 - Jedes einzelne 16-pt-Ausgabemodul	0	1
1746-O*32 - Jedes einzelne 32-pt-Ausgabemodul	0	2
1746-O*8 - Jedes einzelne 8-pt-Ausgabemodul	0	1
1746-OA16 - 16 Output (Triac) 100/240 VAC	0	1
1746-OA8 - 8 Output (Triac) 100/240 VAC	0	1
1746-OAP12 - 12 Output [Triac] 120/240 VDC	0	1
1746-OB16 - 16 Output [Trans] (Quelle) 10/50 VDC	0	1
1746-OB16E - 16 Output [Trans] (Quelle) Geschützt	0	1
1746-OB32 - 32 Output [Trans] (Quelle) 10/50 VDC	0	2

Modultyp	Eingabewörter	Ausgabewörter
1746-OB32E - 32 Output [Trans] (Quelle) 10/50 VDC	0	2
1746-OB6EI - 6 Output [Trans] (Quelle) 24 VDC	0	1
1746-OB8 - 8 Output [Trans] (Quelle) 10/50 VDC	0	1
1746-OBP16 - 16 Output [Trans 1 amp] (SRC) 24 VDC	0	1
1746-OBP8 - 8 Output [Trans 2 amp] (Quelle) 24 VDC	0	1
1746-OG16 - 16 Output [TLL] (SENKE) 5 VDC	0	1
1746-OV16 - 16 Output [Trans] (Senke) 10/50 VDC	0	1
1746-OV32 - 32 Output [Trans] (Senke) 10/50 VDC	0	2
1746-OV8 - 8 Output [Trans] (Senke) 10/50 VDC	0	1
1746-OVP16 - 16 Output [Trans 1 amp] (Senke) 24VDC3	0	1
1746-OW16 - 16 Output [Relais] VAC/VDC	0	1
1746-OW4 - 4 Output [Relais] VAC/VDC	0	1
1746-OW8 - 8 Output [Relais] VAC/VDC	0	1
1746-OX8 - 8 Output [Isoliertes Relais] VAC/VDC	0	1
1747-DCM - Direktkommunikationsmodul (1/2 Rack)	4	4
1747-DCM - Direktkommunikationsmodul (1/4 Rack)	2	2
1747-DCM - Direktkommunikationsmodul (3/4 Rack)	6	6
1747-DCM - Direktkommunikationsmodul (Volles Rack)	8	8
1747-DSN - Verteilter E/A-Scanner - 30 Blöcke	32	32
1747-DSN - Verteilter E/A-Scanner - 7 Blöcke	8	8
1747-KE - Schnittstellenmodul, Reihe A	1	0
1747-KE - Schnittstellenmodul - Reihe B	8	8
1747-MNET - MNET-Netzwerk-Kommunikationsmodul	0	0
1746-QS - Synchronisiertes Achsensteuerungsmodul	32	32
1747-QV - Open Loop-Geschwindigkeitsregelung	8	8
1747-RCIF - Robot-Steuerung-Schnittstellenmodul	32	32
1747-SCNR - ControlNet-SLC-Scanner	32	32
1747-SDN - DeviceNet-Scanner-Modul	32	32
1747-SN - Remote-E/A-Scanner	32	32
AMCI-1561 AMCI Series 1561 Resolver Module	8	8

Geräteigenschaften - 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.

Datentypbeschreibung

Datentyp	Beschreibung
Boolean	Einzelnes Bit
Byte	8-Bit-Wert ohne Vorzeichen Bit 0 ist das Low-Bit Bit 7 ist das High-Bit
Char	8-Bit-Wert mit Vorzeichen Bit 0 ist das Low-Bit Bit 6 ist das High-Bit Bit 7 ist das Vorzeichen-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
Long	32-Bit-Wert mit Vorzeichen
BCD	BCD mit zwei Byte gepackt, vier Dezimalstellen
LBCD	BCD mit vier Byte gepackt, acht Dezimalstellen
Float	32-Bit-IEEE-Gleitkommazahl
String	Mit Null beendetes Zeichen-Array

● **Hinweis:** Die Datentypen "DWord", "Long" und "LBCD" sind für keines der SPS-Modelle nativ.

Beim Referenzieren einer 16-Bit-Position als 32-Bit-Wert handelt es sich bei der referenzierten Position um das Low-Wort und bei der nachfolgenden Position um das High-Wort. Beispiel: Wenn N7:10 als DWord-Datentyp ausgewählt wird, wäre N7:10 das Low-Wort und N7:11 das High-Wort.

Adressbeschreibungen

Adressspezifikationen sind je nach Modell unterschiedlich. Wählen Sie eine Verknüpfung aus der folgenden Tabelle aus, um bestimmte Adressinformationen für das entsprechende Modell abzurufen.

Modell	Output	Input	Status	Binary	Timer	Counter	Control	Integer	Float	ASCII	String	BCD	Long	PID	Message	Block Transfer	Function
Micrologix	X	X	X	X	X	X	X	X	X		X		X	X	X		X
Micro800	X	X	X	X	X	X	X	X	X		X		X	X	X		X
SLC500*	X	X	X	X	X	X	X	X									
SLC5/01	X	X	X	X	X	X	X	X									
SLC5/02	X	X	X	X	X	X	X	X									
SLC5/03	X	X	X	X	X	X	X	X	X	X	X						
SLC5/04	X	X	X	X	X	X	X	X	X	X	X						
SLC5/05	X	X	X	X	X	X	X	X	X	X	X						
PLC5	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	

* Fester E/A-Prozessor

Ausgabedateien

Die Syntax für den Zugriff auf Daten in einer Ausgabedatei ist je nach SPS-Modell unterschiedlich. Arrays werden für Ausgabedateien nicht unterstützt. Die Standard-Datentypen werden **fett** dargestellt.

Syntax für PLC-5

Syntax	Datentyp	Zugriff
O:<Wort>	Short, Word , BCD	Lesen/Schreiben
O:<Wort>/<Bit>	Boolean	Lesen/Schreiben
O/Bit	Boolean	Lesen/Schreiben

● **Hinweis:** Wort- und Bit-Adressinformationen liegen für PLC-5-Modelle im Oktalformat vor. Dies entspricht der Konvention der Programmiersoftware.

Micrologix-Syntax

Syntax	Datentyp	Zugriff
O:<Wort>	Short, Word , BCD	Lesen/Schreiben
O:<Wort>/<Bit>	Boolean	Lesen/Schreiben
O/Bit	Boolean	Lesen/Schreiben

MicroLogix-Modelle verfügen über zwei E/A-Typen: eingebettete E/A und Erweiterungs-E/A (nicht zutreffend für MicroLogix 1000). Eingebettete E/A befindet sich bei der CPU-Basiseinheit, während Erweiterungs-E/A an die CPU-Basiseinheit angeschlossen ist. In der Tabelle unten werden die E/A-Funktionen jedes MicroLogix-Modells aufgelistet.

MicroLogix-Modell	Eingebettete E/A	Erweiterungs-E/A
1000	Steckplatz 0	k.A.
1100	Steckplatz 0	Steckplätze 1-4

MicroLogix-Modell	Eingebettete E/A	Erweiterungs-E/A
1200	Steckplatz 0	Steckplätze 1-6
1400	Steckplatz 0	Steckplätze 1-7
1500	Steckplatz 0	Steckplätze 1-16

Die Adresssyntax für MicroLogix-E/A referenziert ein nullbasiertes Wort-Offset und keinen Steckplatz. Berechnungen müssen erfolgen, um den Wort-Offset gegenüber einem bestimmten Steckplatz zu bestimmen. Dies erfordert Kenntnisse über die Module und deren jeweilige Größe in Wörtern. In der Tabelle unten wird die Größe einiger verfügbarer Module angegeben. Es wird jedoch empfohlen, in der Micrologix-Dokumentation und im Controller-Projekt nachzusehen, um die tatsächliche Wortgröße eines Moduls zu bestimmen. Anweisungen und Beispiele zur Berechnung des Wort-Offsets folgen in der Tabelle unten.

Micrologix - Eingebettete E/A-Wortgrößen

MicroLogix-Modell	# Eingabewörter	# Ausgabewörter
1000	2	1
1100	6	4
1200	4	4
1400	8	6
1500	4	4

Micrologix - Erweiterungs-E/A-Wortgrößen

Module	# Eingabewörter	# Ausgabewörter
1769-HSC	35	34
1769-IA8I	1	0
1769-IA16	1	0
1769-IF4	6	0
1769-IF4XOF2	8	2
1769-IF8	12	1
1769-IM12	1	0
1769-IQ16	1	0
1769-IQ6XOW4	1	1
1769-IQ16F	1	0
1769-IQ32	2	0
1769-IR6	8	0
1769-IT6	8	0
1769-OA8	0	1
1769-OA16	0	1
1769-OB8	0	1
1769-OB16	0	1
1769-OB16P	0	1
1769-OB32	0	2
1769-OF2	2	2
1769-OF8C	11	9
1769-OF8V	11	9
1769-OV16	0	1
1769-OW8	0	1
1769-OW16	0	1
1769-OW8I	0	1

Module	# Eingabewörter	# Ausgabewörter
1769-SDN	66	2
1769-SM1	12	12
1769-SM2	7	7
1769-ASCII	108	108
1762-IA8	1	0
1762-IF2OF2	6	2
1762-IF4	7	0
1762-IQ8	1	0
1762-IQ8OW6	1	1
1762-IQ16	1	0
1762-OA8	0	1
1762-OB8	0	1
1762-OB16	0	1
1762-OW8	0	1
1762-OW16	0	1
1762-IT4	6	0
1762-IR4	6	0
1762-OF4	2	4
1762-OX6I	0	1

Berechnung

Ausgabewort-Offset für Steckplatz $x = \#$ Ausgabewörter in Steckplatz 0 durch Steckplatz $(x-1)$.

Hinweise:

1. Die Eingebettete E/A muss beim Offset auf Erweiterungs-E/A berücksichtigt werden.
2. Die Anzahl der Eingabewörter ist kein Faktor bei der Berechnung des Ausgabewort-Offsets.

E/A-Beispiel

Annahme

Steckplatz 0 = MicroLogix 1500 LRP-Reihe C = 4 Ausgabewörter

Steckplatz 1 = 1769-OF2 = 2 Ausgabewörter

Steckplatz 2 = 1769-OW8 = 1 Ausgabewort

Steckplatz 3 = 1769-IA16 = 0 Ausgabewort

Steckplatz 4 = 1769-OF8V = 9 Ausgabewort

Bit 5 von Steckplatz 4 = $4 + 2 + 1 = 7$ Wörter = O:7/5

SLC 500-Syntax

Der Standard-Datentyp wird **fett** dargestellt.

Syntax	Datentyp	Zugriff
O:<Steckplatz>	Short, Word , BCD	Schreibgeschützt
O:<Steckplatz>.<Wort>	Short, Word , BCD	Schreibgeschützt
O:<Steckplatz>/<Bit>	Boolean	Schreibgeschützt
O:<Steckplatz>.<Wort>/<Bit>	Boolean	Schreibgeschützt

Bereiche

SPS-Modell	Min. Steckplatz	Max. Steckplatz	Max. Wort
Micrologix	k.A.	k.A.	2047

SPS-Modell	Min. Steckplatz	Max. Steckplatz	Max. Wort
SLC500 offen	k.A.	k.A.	1
SLC 5/01	1	30	*
SLC 5/02	1	30	*
SLC 5/03	1	30	*
SLC 5/04	1	30	*
SLC 5/05	1	30	*
PLC-5-Familie	k.A.	k.A.	277**

*Die Anzahl der für jedes E/A-Modul verfügbaren Eingabe- oder Ausgabewörter ist unter [Richtlinien für modulare E/A-Auswahl](#) zu finden. Weitere Hilfe zur Steckplatzkonfiguration finden Sie unter [Geräte-Setup](#).

**Oktal.

Beispiele

Micrologix	Adresse
O:0	Wort 0
O/2	Bit 2
O:0/5	Bit 5

Feste E/A für SLC500	Adresse
O:0	Wort 0
O:1	Word 1
O/16	Bit 16
O:1/0	Bit 0 Wort 1 (identisch mit O/16)

PLC5	Adresse*
O:0	Wort 0
O:37	Wort 31 (37 oktal = 31 dezimal)
O/42	Bit 34 (42 oktal = 34 dezimal)
O:2/2	Bit 2 Wort 2 (identisch mit O/42)

*Adressen werden im Oktalformat angezeigt.

Modulare E/A für SLC500	Adresse
O:1	Wort 0 Steckplatz 1
O:1.0	Wort 0 Steckplatz 1 (identisch mit O:1)
O:12	Wort 0 Steckplatz 12
O:12.2	Wort 2 Steckplatz 12
O:4.0/0	Bit 0 Wort 0 Steckplatz 4
O:4/0	Bit 0 Steckplatz 4 (identisch mit O:4.0/0)
O:4.2/0	Bit 0 Wort 2 Steckplatz 4
O:4/32	Bit 32 Steckplatz 4 (identisch mit O:4.2/0)

Eingabedateien

Die Syntax für den Zugriff auf Daten in einer Eingabedatei ist je nach SPS-Modell unterschiedlich. Arrays werden für Eingabedateien nicht unterstützt. Die Standard-Datentypen werden **fett** dargestellt.

Syntax für PLC-5

Syntax	Datentyp	Zugriff
I:<Wort>	Short, Word , BCD	Lesen/Schreiben

Syntax	Datentyp	Zugriff
I:<Wort>/<Bit>	Boolean	Lesen/Schreiben
I/Bit	Boolean	Lesen/Schreiben

● **Hinweis:** Wort- und Bit-Adressinformationen liegen für PLC-5-Modelle im Oktalformat vor. Dies entspricht der Konvention der Programmiersoftware.

Micrologix-Syntax

Syntax	Datentyp	Zugriff
I:<Wort>	Short, Word, BCD	Lesen/Schreiben
I:<Wort>/<Bit>	Boolean	Lesen/Schreiben
I/Bit	Boolean	Lesen/Schreiben

MicroLogix-Modelle verfügen über zwei E/A-Typen: eingebettete E/A und Erweiterungs-E/A (nicht zutreffend für MicroLogix 1000). Eingebettete E/A befindet sich bei der CPU-Basiseinheit, während Erweiterungs-E/A an die CPU-Basiseinheit angeschlossen ist. In der Tabelle unten werden die E/A-Funktionen jedes MicroLogix-Modells aufgelistet.

MicroLogix-Modell	Eingebettete E/A	Erweiterungs-E/A
1000	Steckplatz 0	k.A.
1100	Steckplatz 0	Steckplätze 1-4
1200	Steckplatz 0	Steckplätze 1-6
1400	Steckplatz 0	Steckplätze 1-7
1500	Steckplatz 0	Steckplätze 1-16

Die Adresssyntax für MicroLogix-E/A referenziert ein nullbasiertes Wort-Offset und keinen Steckplatz. Berechnungen müssen erfolgen, um den Wort-Offset gegenüber einem bestimmten Steckplatz zu bestimmen. Dies erfordert Kenntnisse über die Module und deren jeweilige Größe in Wörtern. In der Tabelle unten wird die Größe einiger verfügbarer Module angegeben. Es wird jedoch empfohlen, in der Micrologix-Dokumentation und im Controller-Projekt nachzusehen, um die tatsächliche Wortgröße eines Moduls zu bestimmen. Anweisungen und Beispiele zur Berechnung des Wort-Offsets folgen in der Tabelle unten.

Micrologix - Eingebettete E/A-Wortgrößen

MicroLogix-Modell	# Eingabewörter	# Ausgabewörter
1000	2	1
1100	6	4
1200	4	4
1400	8	6
1500	4	4

Micrologix - Erweiterungs-E/A-Wortgrößen

Module	# Eingabewörter	# Ausgabewörter
1769-HSC	35	34
1769-IA8I	1	0
1769-IA16	1	0
1769-IF4	6	0
1769-IF4XOF2	8	2
1769-IF8	12	1
1769-IM12	1	0
1769-IQ16	1	0

Module	# Eingabewörter	# Ausgabewörter
1769-IQ6XOW4	1	1
1769-IQ16F	1	0
1769-IQ32	2	0
1769-IR6	8	0
1769-IT6	8	0
1769-OA8	0	1
1769-OA16	0	1
1769-OB8	0	1
1769-OB16	0	1
1769-OB16P	0	1
1769-OB32	0	2
1769-OF2	2	2
1769-OF8C	11	9
1769-OF8V	11	9
1769-OV16	0	1
1769-OW8	0	1
1769-OW16	0	1
1769-OW8I	0	1
1769-SDN	66	2
1769-SM1	12	12
1769-SM2	7	7
1769-ASCII	108	108
1762-IA8	1	0
1762-IF2OF2	6	2
1762-IF4	7	0
1762-IQ8	1	0
1762-IQ8OW6	1	1
1762-IQ16	1	0
1762-OA8	0	1
1762-OB8	0	1
1762-OB16	0	1
1762-OW8	0	1
1762-OW16	0	1
1762-IT4	6	0
1762-IR4	6	0
1762-OF4	2	4
1762-OX6I	0	1

Berechnung

Eingabewort-Offset für Steckplatz x = # Eingabewörter in Steckplatz 0 durch Steckplatz (x-1).

● Hinweise:

1. Die Eingebettete E/A muss beim Offset auf Erweiterungs-E/A berücksichtigt werden.
2. Die Anzahl der Ausgabewörter ist kein Faktor bei der Berechnung des Eingabewort-Offsets.

E/A-Beispiel

Annahme

Steckplatz 0 = Micrologix 1500 LRP-Reihe C = 4 Eingabewörter
 Steckplatz 1 = 1769-OF2 = 2 Eingabewörter
 Steckplatz 2 = 1769-OW8 = 0 Eingabewörter
 Steckplatz 3 = 1769-IA16 = 1 Eingabewörter
 Steckplatz 4 = 1769-OF8V = 11 Eingabewörter
 Bit 5 von Steckplatz 3 = 4 + 2 = 6 Wörter = I:6/5

SLC 500-Syntax

Der Standard-Datentyp wird **fett** dargestellt.

Syntax	Datentyp	Zugriff
I:<Steckplatz>	Short, Word , BCD	Schreibgeschützt
I:<Steckplatz>.<Wort>	Short, Word , BCD	Schreibgeschützt
I:<Steckplatz>/<Bit>	Boolean	Schreibgeschützt
I:<Steckplatz>.<Wort>/<Bit>	Boolean	Schreibgeschützt

Bereiche

SPS-Modell	Min. Steckplatz	Max. Steckplatz	Max. Wort
Micrologix	k.A.	k.A.	2047
SLC500 offen	k.A.	k.A.	1
SLC 5/01	1	30	*
SLC 5/02	1	30	*
SLC 5/03	1	30	*
SLC 5/04	1	30	*
SLC 5/05	1	30	*
PLC-5-Familie	k.A.	k.A.	277

*Die Anzahl der für jedes E/A-Modul verfügbaren Eingabe- oder Ausgabewörter ist unter **Richtlinien für modulare E/A-Auswahl** zu finden. Weitere Informationen dazu finden Sie unter [Geräte-Setup](#).

**Oktal.

Beispiele

Micrologix	Adresse
I:0	Wort 0
I/2	Bit 2
I:1/5	Bit 5 Wort 1

Feste E/A für SLC500	Adresse
I:0	Wort 0
I:1	Wort 1
I/16	Bit 16
I:1/0	Bit 0 Wort 1 (identisch mit I/16)

PLC5	Adresse
I:0	Wort 0
I:10	Word 8 (10 oktal = 8 dezimal)
I/20	Bit 16 (20 oktal = 16 dezimal)
I:1/0	Bit 0 Wort 1 (identisch mit I/20)

*Adressen werden im Oktalformat angezeigt.

Modulare E/A für SLC500	Adresse
I:1	Wort 0 Steckplatz 1
I:1.0	Wort 0 Steckplatz 1 (identisch mit I:1)
I:12	Wort 0 Steckplatz 12
I:12.2	Wort 2 Steckplatz 12
I:4.0/0	Bit 0 Wort 0 Steckplatz 4
I:4/0	Bit 0 Steckplatz 4 (identisch mit I:4.0/0)
I:4.2/0	Bit 0 Wort 2 Steckplatz 4
I:4/32	Bit 32 Steckplatz 4 (identisch mit I:4.2/0)

Statusdateien

Geben Sie für den Zugriff auf Statusdateien ein Wort und ein optionales Bit im Wort an. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
S:<Wort>	Short, Word , BCD, DWord, Long, LBCD	Lesen/Schreiben
S:<Wort> [Zeilen][Spalten]	Short, Word , BCD, DWord, Long, LBCD*	Lesen/Schreiben
S:<Wort> [Spalten]	Short, Word , BCD, DWord, Long, LBCD*	Lesen/Schreiben
S:<Wort>/<Bit>	Boolean	Lesen/Schreiben
S/Bit	Boolean	Lesen/Schreiben

*Array-Typen.

Die Anzahl der Array-Elemente (in Byte) darf die angegebene Blockanforderungsgröße nicht überschreiten. Dies bedeutet, dass die Array-Größe 16 Wortwerte bei einer Blockanforderungsgröße von 32 Byte nicht überschreiten darf.

Bereiche

SPS-Modell	Max. Wort
Micrologix	96
Alle SLC	96
PLC-5	127

Die maximale Wortposition ist eine weniger, wenn als 32-Bit-Datentyp (Long, DWord oder Long-BCD) zugegriffen wird.

Beispiele

Beispiel	Description
S:0	Wort 0.
S/26	Bit 26.
S:4/15	Bit 15 Wort 4.
S:10 [16]	Array mit 16 Elementen, beginnend bei Wort 10.
S:0 [4] [8]	Array mit 4 von 8 Elementen, beginnend bei Wort 0.

Binäre Dateien

Geben Sie für den Zugriff auf binäre Dateien eine Dateinummer, ein Wort und ein optionales Bit im Wort an. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
B<Datei>:<Wort>	Short, Word , BCD,	Lesen/Schreiben

Syntax	Datentyp	Zugriff
	DWord, Long, LBCD	
B<Datei>:<Wort> [Zeilen][Spalten]	Short, Word , BCD, DWord, Long, LBCD*	Lesen/Schreiben
B<Datei>:<Wort> [Spalten]	Short, Word , BCD, DWord, Long, LBCD*	Lesen/Schreiben
B<Datei>:<Wort>/<Bit>	Boolean	Lesen/Schreiben
B<Datei>/Bit	Boolean	Lesen/Schreiben

*Array-Typen.

Die Anzahl der Array-Elemente (in Byte) darf die angegebene Blockanforderungsgröße nicht überschreiten. Dies bedeutet, dass die Array-Größe 16 Word-Werte bei einer Blockanforderungsgröße von 32 Byte nicht überschreiten darf.

Bereiche

SPS-Modell	Dateinummer	Max. Wort
Micrologix	3-255	255
Alle SLC	3-255	255
PLC-5	3-999	1999

Die maximale Wortposition ist eine weniger, wenn als 32-Bit-Datentyp (Long, DWord oder Long-BCD) zugegriffen wird.

Beispiele

Beispiel	Description
B3:0	Wort 0.
B3/26	Bit 26.
B12:4/15	Bit 15 Wort 4.
B3:10 [20]	Array mit 20 Elementen, beginnend bei Wort 10.
B15:0 [6][6]	Array mit 6 mal 6 Elementen, beginnend bei Wort 0.

Zeitgeberdateien

Bei Zeitgeberdateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen einer Dateinummer, eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
T<Datei>:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Informationen zur Verwendung jedes Felds finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
ACC	Short , Word	Lesen/Schreiben
PRE	Short , Word	Lesen/Schreiben
DN	Boolean	Schreibgeschützt
TT	Boolean	Schreibgeschützt
EN	Boolean	Schreibgeschützt

Bereiche

SPS-Modell	Dateinummer	Max. Element
Micrologix	3-255	255
Alle SLC	3-255	255
PLC-5	3-999	1999

Beispiele

Beispiel	Description
T4:0.ACC	Akkumulator von Zeitgeber 0, Datei 4.
T4:10.DN	Abschluss-Bit von Zeitgeber 10, Datei 4.
T15:0.PRE	Voreinstellung von Zeitgeber 0, Datei 15.

Zählerdateien

Bei Zählerdateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen einer Dateinummer, eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
C<Datei>:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
ACC	Word, Short	Lesen/Schreiben
PRE	Word, Short	Lesen/Schreiben
UA	Boolean	Schreibgeschützt
UN	Boolean	Schreibgeschützt
OV	Boolean	Schreibgeschützt
DN	Boolean	Schreibgeschützt
CD	Boolean	Schreibgeschützt
CU	Boolean	Schreibgeschützt

Bereiche

SPS-Modell	Dateinummer	Max. Element
Micrologix	3-255	255
Alle SLC	3-255	255
PLC-5	3-999	1999

Beispiele

Beispiel	Description
C5:0.ACC	Akkumulator von Zähler 0, Datei 5.
C5:10.DN	Abschluss-Bit von Zähler 10, Datei 5.
C15:0.PRE	Voreinstellung von Zähler 0, Datei 15.

Steuerdateien

Bei Steuerdateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen einer Dateinummer, eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
R<Datei>:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
LEN	Word, Short	Lesen/Schreiben
POS	Word, Short	Lesen/Schreiben
FD	Boolean	Schreibgeschützt
IN	Boolean	Schreibgeschützt
UL	Boolean	Schreibgeschützt
ER	Boolean	Schreibgeschützt
EM	Boolean	Schreibgeschützt
DN	Boolean	Schreibgeschützt
EU	Boolean	Schreibgeschützt
EN	Boolean	Schreibgeschützt

Bereiche

SPS-Modell	Dateinummer	Max. Element
Micrologix	3-255	255
Alle SLC	3-255	255
PLC-5	3-999	1999

Beispiele

Beispiel	Description
R6:0.LEN	Längenfeld von Steuerung 0, Datei 6.
R6:10.DN	Abschluss-Bit von Steuerung 10, Datei 6.
R15:18.POS	Positionsfeld von Steuerung 18, Datei 15.

Ganzzahldateien

Geben Sie für den Zugriff auf Ganzzahldateien eine Dateinummer, ein Wort und ein optionales Bit im Wort an. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
N<Datei>:<Wort>	Short, Word , BCD, DWord, Long, LBCD	Lesen/Schreiben
N<Datei>:<Wort> [rows][cols]	Short, Word , BCD, DWord, Long, LBCD*	Lesen/Schreiben
N<Datei>:<Wort> [cols]	Short, Word , BCD, DWord, Long, LBCD*	Lesen/Schreiben
N<Datei>:<Wort>/<Bit>	Boolean	Lesen/Schreiben
N<Datei>/Bit	Boolean	Lesen/Schreiben

*Array-Typen.

Die Anzahl der Array-Elemente (in Byte) darf die angegebene Blockanforderungsgröße nicht überschreiten. Dies bedeutet, dass die Array-Größe 16 Wortwerte bei einer Blockanforderungsgröße von 32 Byte nicht überschreiten darf.

Bereiche

SPS-Modell	Dateinummer	Max. Wort
Micrologix	3-255	255
Alle SLC	3-255	255
PLC-5	3-999	1999

Die maximale Wortposition ist eine weniger, wenn als 32-Bit-Datentyp (Long, DWord oder Long-BCD) zugegriffen wird.

Beispiele

Beispiel	Beschreibung
N7:0	Wort 0.
N7/26	Bit 26.
N12:4/15	Bit 15 Wort 4.
N7:10 [8]	Array mit 8 Elementen, beginnend bei Wort 10.
N15:0 [4] [5]	Array mit 4 von 5 Elementen, beginnend bei Wort 0.

Float-Dateien

Geben Sie für den Zugriff auf Float-Dateien eine Dateinummer und ein Element an. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
F<Datei>:<Element>	Float	Lesen/Schreiben
F<Datei>:<Element> [rows][cols]	Float*	Lesen/Schreiben
F<Datei>:<Element> [cols]	Float*	Lesen/Schreiben

*Array-Typ.

Die Anzahl der Array-Elemente (in Byte) darf die angegebene Blockanforderungsgröße nicht überschreiten. Dies bedeutet, dass die Array-Größe 8 Float-Werten bei einer Blockanforderungsgröße von 32 Byte nicht überschreiten darf.

Bereiche

SPS-Modell	Dateinummer	Max. Wort
Micrologix	3-255	255
Alle SLC	3-255	255
PLC-5	3-999	1999

Beispiele

Beispiel	Beschreibung
F8:0	Float (Gleitkommazahl) 0.
F8:10 [16]	Array mit 16 Elementen, beginnend bei Wort 10.
F15:0 [4] [4]	Array mit 16 Elementen, beginnend bei Wort 0.

ASCII-Dateien

Geben Sie für den Zugriff auf Daten in einer ASCII-Datei eine Dateinummer und Zeichenposition an. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
A<Datei>:<Char>	Char, Byte*	Lesen/Schreiben
A<Datei>:<Char> [Zeilen][Spalten]	Char, Byte*	Lesen/Schreiben
A<Datei>:<Char> [Spalten]	Char, Byte*	Lesen/Schreiben

Syntax	Datentyp	Zugriff
A<Datei>:<Wort-Offset>/Länge	String**	Lesen/Schreiben

*Die Anzahl der Array-Elemente darf die angegebene Blockanforderungsgröße nicht überschreiten. Intern packt die SPS zwei Zeichen pro Wort in die Datei, wobei das High-Byte das erste Zeichen und das Low-Byte das zweite Zeichen enthält. Die SPS-Programmiersoftware ermöglicht Zugriff auf der Wortebene oder der Zwei-Zeichen-Ebene. DF1-Treiber von Allen-Bradley ermöglicht Zugriff auf die Zeichenebene.

Wenn Sie die Programmiersoftware verwenden, würde bei **A10:0 = AB** der Wert 'A' im High-Byte von A10:0 und 'B' im Low-Byte gespeichert. Unter Verwendung von DF1-Treiber von Allen-Bradley würden die beiden Zuweisungen **A10:0 = A** und **A10:1 = B** zur Folge haben, dass dieselben Daten im SPS-Speicher gespeichert werden.

**Das Referenzieren dieser Datei als Zeichenfolgendaten ermöglicht Zugriff auf Daten an Wortgrenzen wie die Programmiersoftware. Die Länge kann aus bis zu 236 Zeichen bestehen. Wenn eine an das Gerät gesendete Zeichenfolge kürzer ist als die durch die Adresse angegebene Länge, beendet der Treiber die Zeichenfolge mit Null, bevor sie an den Controller gesendet wird.

Bereiche

SPS-Modell	Dateinummer	Max. Zeichen
Micrologix	3-255	511
Alle SLC	3-255	511
PLC-5	3-999	1999

● *Nicht alle Micrologix- und SLC500-SPS-Geräte unterstützen ASCII-Dateitypen. Weitere Informationen dazu finden Sie in der SPS-Dokumentation.*

Beispiele

Beispiel	Description
A9:0	Zeichen 0 (High-Byte von Wort 0).
A27:10 [80]	Char-Array aus 80 Zeichen, das bei Zeichen 10 beginnt.
A15:0 [4] [16]	Char-Array aus 4 mal 16 Zeichen, das bei Zeichen 0 beginnt.
A62:0/32	Char-Array aus 32 Zeichenfolgen, das bei Wort-Offset 0 beginnt.

String-Dateien

Geben Sie für den Zugriff auf Daten in einer String-Datei eine Dateinummer und ein Element an. Zeichenfolgen sind mit Null beendete Arrays mit 82 Zeichen. Der Treiber platziert das Nullabschlusszeichen basierend auf der von der SPS zurückgegebenen Zeichenfolgenlänge. Die Standard-Datentypen werden **fett** dargestellt.

● **Hinweis:** Arrays werden für String-Dateien nicht unterstützt.

Syntax	Datentyp	Zugriff
ST<Datei>:<Element><Feld>	String	Lesen/Schreiben

Bereiche

SPS-Modell	Dateinummer	Max. Wort
Micrologix	3-255	255
Alle SLC	3-255	255
PLC-5	3-999	999

Beispiele

Beispiel	Description
ST9:0	Zeichenfolge 0.
ST18:10	Zeichenfolge 10.

BCD-Dateien

Geben Sie für den Zugriff auf BCD-Dateien eine Dateinummer und ein Wort an. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
D<Datei>:<Wort>	BCD , LBCD	Lesen/Schreiben
D<Datei>:<Wort> [Zeilen][Spalten]	BCD , LBCD*	Lesen/Schreiben
D<Datei>:<Wort> [Spalten]	BCD , LBCD*	Lesen/Schreiben

*Array-Typen.

Die Anzahl der Array-Elemente (in Byte) darf die angegebene Blockanforderungsgröße nicht überschreiten. Dies bedeutet, dass die Array-Größe 16 BCD-Werte bei einer Blockanforderungsgröße von 32 Byte nicht überschreiten darf.

Bereiche

SPS-Modell	Dateinummer	Max. Wort
Micrologix	k.A.	k.A.
Alle SLC	k.A.	k.A.
PLC-5	3-999	999

Beispiele

Beispiel	Beschreibung
D9:0	Wort 0.
D27:10 [16]	Array mit 16 Elementen, beginnend bei Wort 10.
D15:0 [4] [8]	Array mit 32 Elementen, beginnend bei Wort 0.

Long-Dateien

Geben Sie für den Zugriff auf Long-Dateien eine Dateinummer und ein DWord an. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
L<Datei>:<DWord>	Long, DWord , LBCD	Lesen/Schreiben
L<Datei>:<DWord> [Zeilen][Spalten]	Long, DWord , LBCD*	Lesen/Schreiben
L<Datei>:<DWord> [Spalten]	Long, DWord , LBCD*	Lesen/Schreiben

*Array-Typen.

Die Anzahl der Array-Elemente darf 16 nicht überschreiten.

Bereiche

SPS-Modell	Dateinummer	Max. Wort
Micrologix	3-255	255
Alle SLC	k.A.	k.A.
PLC5	k.A.	k.A.

Beispiele

Beispiel	Description
L9:0	Wort 0.
L9:10 [8]	Array mit 8 Elementen, beginnend bei Wort 10.
L15:0 [4] [5]	Array mit 4 von 5 Elementen, beginnend bei Wort 0.

MicroLogix-PID-Dateien

Bei PID-Dateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen einer Dateinummer, eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
PD<Datei>:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
SPS	Word , Short	Lesen/Schreiben
KC	Word , Short	Lesen/Schreiben
TI	Word , Short	Lesen/Schreiben
TD	Word , Short	Lesen/Schreiben
MAXS	Word , Short	Lesen/Schreiben
MINS	Word , Short	Lesen/Schreiben
ZCD	Word , Short	Lesen/Schreiben
CVH	Word , Short	Lesen/Schreiben
CVL	Word , Short	Lesen/Schreiben
LUT	Word , Short	Lesen/Schreiben
SPV	Word , Short	Lesen/Schreiben
CVP	Word , Short	Lesen/Schreiben
TM	Boolean	Lesen/Schreiben
AM	Boolean	Lesen/Schreiben
CM	Boolean	Lesen/Schreiben
OL	Boolean	Lesen/Schreiben
RG	Boolean	Lesen/Schreiben
SC	Boolean	Lesen/Schreiben
TF	Boolean	Lesen/Schreiben
DA	Boolean	Lesen/Schreiben
DB	Boolean	Lesen/Schreiben
UL	Boolean	Lesen/Schreiben
LL	Boolean	Lesen/Schreiben
SP	Boolean	Lesen/Schreiben
PV	Boolean	Lesen/Schreiben
DN	Boolean	Lesen/Schreiben
EN	Boolean	Lesen/Schreiben

Bereiche

SPS-Modell	Dateinummer	Max. Element
Micrologix	3-255	255
Alle SLC	k.A.	k.A.

SPS-Modell	Dateinummer	Max. Element
PLC-5	*	*

• *Weitere Informationen dazu finden Sie unter [PLC5-PID-Dateien](#).

Beispiele

Beispiel	Description
PD14:0.KC	Proportionaler Zuwachs von PD 0 Datei 14.
PD18:6.EN	PID-Aktivierungs-Bit von PD 6 Datei 18.

PLC5-PID-Dateien

Bei PID-Dateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen einer Dateinummer, eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
PD<Datei>.<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
SP	Real	Lesen/Schreiben
KP	Real	Lesen/Schreiben
KI	Real	Lesen/Schreiben
KD	Real	Lesen/Schreiben
BIAS	Real	Lesen/Schreiben
MAXS	Real	Lesen/Schreiben
MINS	Real	Lesen/Schreiben
DB	Real	Lesen/Schreiben
SO	Real	Lesen/Schreiben
MAXO	Real	Lesen/Schreiben
MINO	Real	Lesen/Schreiben
UPD	Real	Lesen/Schreiben
PV	Real	Lesen/Schreiben
ERR	Real	Lesen/Schreiben
OUT	Real	Lesen/Schreiben
PVH	Real	Lesen/Schreiben
PVL	Real	Lesen/Schreiben
DVP	Real	Lesen/Schreiben
DVN	Real	Lesen/Schreiben
PVDB	Real	Lesen/Schreiben
DVDB	Real	Lesen/Schreiben
MAXI	Real	Lesen/Schreiben
MINI	Real	Lesen/Schreiben
TIE	Real	Lesen/Schreiben
FILE	Word, Short	Lesen/Schreiben
ELEM	Word, Short	Lesen/Schreiben
EN	Boolean	Lesen/Schreiben

Elementfeld	Datentyp	Zugriff
CT	Boolean	Lesen/Schreiben
CL	Boolean	Lesen/Schreiben
PVT	Boolean	Lesen/Schreiben
DO	Boolean	Lesen/Schreiben
SWM	Boolean	Lesen/Schreiben
CA	Boolean	Lesen/Schreiben
MO	Boolean	Lesen/Schreiben
PE,	Boolean	Lesen/Schreiben
INI	Boolean	Lesen/Schreiben
SPOR	Boolean	Lesen/Schreiben
OLL	Boolean	Lesen/Schreiben
OLH	Boolean	Lesen/Schreiben
EWD	Boolean	Lesen/Schreiben
DVNA	Boolean	Lesen/Schreiben
DVHA	Boolean	Lesen/Schreiben
PVLA	Boolean	Lesen/Schreiben
PVHA	Boolean	Lesen/Schreiben

Bereiche

SPS-Modell	Dateinummer	Max. Element
Micrologix	*	*
Alle SLC	k.A.	k.A.
PLC-5	3-999	999

 *Weitere Informationen dazu finden Sie unter [Micrologix-PID-Dateien](#).

Beispiele

Beispiel	Description
PD14:0.SP	Setpoint-Feld von PD 0 Datei 14.
PD18:6.EN	Statusaktivierungs-Bit von PD 6 Datei 18.

MicroLogix-Meldungsdateien

Bei Meldungsdateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen einer Dateinummer, eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
MG<Datei>:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
IA	Word , Short	Lesen/Schreiben
RBL	Word , Short	Lesen/Schreiben
LBN	Word , Short	Lesen/Schreiben
RBN	Word , Short	Lesen/Schreiben
CHN	Word , Short	Lesen/Schreiben

Elementfeld	Datentyp	Zugriff
NOD	Word, Short	Lesen/Schreiben
MTO	Word, Short	Lesen/Schreiben
NB	Word, Short	Lesen/Schreiben
TFT	Word, Short	Lesen/Schreiben
TFN	Word, Short	Lesen/Schreiben
ELE	Word, Short	Lesen/Schreiben
SEL	Word, Short	Lesen/Schreiben
TO	Boolean	Lesen/Schreiben
CO	Boolean	Lesen/Schreiben
EN	Boolean	Lesen/Schreiben
RN	Boolean	Lesen/Schreiben
EW	Boolean	Lesen/Schreiben
ER	Boolean	Lesen/Schreiben
DN	Boolean	Lesen/Schreiben
ST	Boolean	Lesen/Schreiben
BK	Boolean	Lesen/Schreiben

Die folgenden Dateinummern und maximalen Elemente sind für jedes Modell zulässig.

Bereiche

SPS-Modell	Dateinummer	Max. Element
Micrologix	3-255	255
Alle SLC	k.A.	k.A.
PLC5	*	*

• *Weitere Informationen dazu finden Sie unter [PLC5-Meldung](#).*

Beispiele

Beispiel	Description
MG14:0.TO	Ignorieren bei Timeout-Bit von MG 0 Datei 14.
MG18:6.CO	Fortsetzungs-Bit von MG 6 Datei 18.

PLC5-Meldungsdateien

Bei Meldungsdateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen einer Dateinummer, eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
MG<Datei>:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
ERR	Short, Word	Lesen/Schreiben
RLEN	Short, Word	Lesen/Schreiben
DLEN	Short, Word	Lesen/Schreiben
EN	Boolean	Lesen/Schreiben

Elementfeld	Datentyp	Zugriff
ST	Boolean	Lesen/Schreiben
DN	Boolean	Lesen/Schreiben
ER	Boolean	Lesen/Schreiben
CO	Boolean	Lesen/Schreiben
EW	Boolean	Lesen/Schreiben
NR	Boolean	Lesen/Schreiben
TO	Boolean	Lesen/Schreiben

Bereiche

SPS-Modell	Dateinummer	Max. Element
Micrologix	*	*
Alle SLC	k.A.	k.A.
PLC-5	3-999	999

• *Weitere Informationen dazu finden Sie unter [Micrologix-Meldungsdateien](#).

Beispiele

Beispiel	Description
MG14:0.RLEN	Angefordertes Längenfeld von MG 0 Datei 14.
MG18:6.CO	Fortsetzungs-Bit von MG 6 Datei 18.

Blocktransferdateien

Bei Blocktransferdateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen einer Dateinummer, eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
BT<Datei>:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
RLEN	Word , Short	Lesen/Schreiben
DLEN	Word , Short	Lesen/Schreiben
FILE	Word , Short	Lesen/Schreiben
ELEM	Word , Short	Lesen/Schreiben
RW	Boolean	Lesen/Schreiben
ST	Boolean	Lesen/Schreiben
DN	Boolean	Lesen/Schreiben
ER	Boolean	Lesen/Schreiben
CO	Boolean	Lesen/Schreiben
EW	Boolean	Lesen/Schreiben
NR	Boolean	Lesen/Schreiben
TO	Boolean	Lesen/Schreiben

Bereiche

SPS-Modell	Dateinummer	Max. Element
Micrologix	k.A.	k.A.
Alle SLC	k.A.	k.A.
PLC-5	3-999	1999

Beispiele

Beispiel	Description
BT14:0.RLEN	Angefordertes Längenfeld von BT 0, Datei 14.
BT18:6.CO	Fortsetzungs-Bit von BT 6, Datei 18.

HSC-Datei (High Speed Counter, Hochgeschwindigkeitszähler)

Bei HSC-Dateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

• *Siehe auch:* [Funktionsdateioptionen](#)

Syntax	Datentyp	Zugriff
HSC:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Standardtyp	Zugriff
ACC	DWord , Long	Schreibgeschützt
HIP	DWord , Long	Lesen/Schreiben
LOP	DWord , Long	Lesen/Schreiben
OVF	DWord , Long	Lesen/Schreiben
UNF	DWord , Long	Lesen/Schreiben
PFN	Word , Short	Schreibgeschützt
ER	Word , Short	Schreibgeschützt
MOD	Word , Short	Schreibgeschützt
OMB	Word , Short	Schreibgeschützt
HPO	Word , Short	Lesen/Schreiben
LPO	Word , Short	Lesen/Schreiben
UIX	Boolean	Schreibgeschützt
UIP	Boolean	Schreibgeschützt
AS	Boolean	Schreibgeschützt
ED	Boolean	Schreibgeschützt
SP	Boolean	Schreibgeschützt
LPR	Boolean	Schreibgeschützt
HPR	Boolean	Schreibgeschützt
DIR	Boolean	Schreibgeschützt
CD	Boolean	Schreibgeschützt
CU	Boolean	Schreibgeschützt
UIE	Boolean	Lesen/Schreiben
UIL	Boolean	Lesen/Schreiben
FE	Boolean	Lesen/Schreiben
CE	Boolean	Lesen/Schreiben

Elementfeld	Standardtyp	Zugriff
LPM	Boolean	Lesen/Schreiben
HPM	Boolean	Lesen/Schreiben
UFM	Boolean	Lesen/Schreiben
OFM	Boolean	Lesen/Schreiben
LPI	Boolean	Lesen/Schreiben
HPI	Boolean	Lesen/Schreiben
UFI	Boolean	Lesen/Schreiben
OFI	Boolean	Lesen/Schreiben
UF	Boolean	Lesen/Schreiben
OF	Boolean	Lesen/Schreiben
MD	Boolean	Lesen/Schreiben

Bereiche

SPS-Modell	Dateinummer	Max. Element
Micrologix	k.A.	254
Alle SLC	k.A.	k.A.
PLC5	k.A.	k.A.

Beispiele

Beispiel	Description
HSC:0.OMB	Einstellung der Ausgabemaske für Hochgeschwindigkeitszähler 0.
HSC:1.ED	Hinweis auf festgestellten Fehler für Hochgeschwindigkeitszähler 1.

RTC-Datei (Real Time Clock, Echtzeituhr)

Bei RTC-Dateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

• *Siehe auch:* [Funktionsdateioptionen](#)

Syntax	Datentyp	Zugriff
RTC:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
YR	Word , Short	Lesen/Schreiben
MON	Word , Short	Lesen/Schreiben
DAY	Word , Short	Lesen/Schreiben
HR	Word , Short	Lesen/Schreiben
MIN	Word , Short	Lesen/Schreiben
SEC	Word , Short	Lesen/Schreiben
DOW	Word , Short	Lesen/Schreiben
DS	Boolean	Schreibgeschützt
BL	Boolean	Schreibgeschützt
_SET (for Blockschreibvorgänge)	Boolean	Lesen/Schreiben

Bereiche

SPS-Modell	Dateinummer	Max. Element
Micrologix	k.A.	254
Alle SLC	k.A.	k.A.
PLC5	k.A.	k.A.

Beispiele

Beispiel	Description
RTC:0.YR	Jahreseinstellung für Echtzeituhr 0.
RTC:0.BL	Hinweis auf niedrigen Batteriestand für Echtzeituhr 0.

Kommunikationsstatusdatei für Kanal 0 (CS0, Channel 0 Communication Status File)

Geben Sie für den Zugriff auf die Kommunikationsstatusdatei für Kanal 0 ein Wort und optional ein Bit im Wort an. Die Standard-Datentypen werden **fett** dargestellt.

• *Siehe auch:* [Funktionsdateioptionen](#)

Syntax	Datentyp	Zugriff
CS0:<Wort>	Short, Word , BCD, DWord, Long, LBCD	Hängt von <Wort> und <Bit> ab
CS0:<Wort>/<Bit>	Boolean	Hängt von <Wort> und <Bit> ab
CS0/Bit	Boolean	Hängt von <Wort> und <Bit> ab

Bereiche

SPS-Modell	Dateinummer	Max. Element
Micrologix	k.A.	254
Alle SLC	k.A.	k.A.
PLC5	k.A.	k.A.

Beispiele

Beispiel	Beschreibung
CS0:0	Wort 0.
CS0:4/2	Bit 2, Wort 4 = MCP.

• *Weitere Informationen zu Bedeutungen von CS0-Wörtern/Bit finden Sie in der Rockwell-Dokumentation.*

Kommunikationsstatusdatei für Kanal 1 (CS1, Channel 1 Communication Status File)

Geben Sie für den Zugriff auf die Kommunikationsstatusdatei für Kanal 1 ein Wort und optional ein Bit im Wort an. Die Standard-Datentypen werden **fett** dargestellt.

• *Siehe auch:* [Funktionsdateioptionen](#)

Syntax	Datentyp	Zugriff
CS1:<Wort>	Short, Word , BCD, DWord, Long, LBCD	Hängt von <Wort> und <Bit> ab
CS1:<Wort>/<Bit>	Boolean	Hängt von <Wort> und <Bit> ab
CS1/Bit	Boolean	Hängt von <Wort> und <Bit> ab

Bereiche

SPS-Modell	Dateinummer	Max. Element
Micrologix	k.A.	254
Alle SLC	k.A.	k.A.
PLC5	k.A.	k.A.

Beispiele

Beispiel	Description
CS1:0	Wort 0
CS1:4/2	Bit 2, Wort 4 = MCP

• Weitere Informationen zu Bedeutungen von CS1-Wörtern/Bit finden Sie in der Rockwell-Dokumentation.

IOS-Datei (I/O Module Status, E/A-Modulstatus)

Geben Sie für den Zugriff auf die E/A-Modulstatus-Datei ein Wort und optional ein Bit im Wort an. Der Standard-Datentyp für jede Syntax wird **fett** dargestellt.

• *Siehe auch:* [Funktionsdateioptionen](#)

Syntax	Datentyp	Zugriff
IOS:<Wort>	Short, Word , BCD, DWord, Long, LBCD	Hängt von <Wort> und <Bit> ab
IOS:<Wort>/<Bit>	Boolean	Hängt von <Wort> und <Bit> ab
IOS/Bit	Boolean	Hängt von <Wort> und <Bit> ab

Bereiche

SPS-Modell	Dateinummer	Max. Element
Micrologix	k.A.	254
Alle SLC	k.A.	k.A.
PLC5	k.A.	k.A.

Beispiele

Beispiel	Description
IOS:0	Wort 0.
IOS:4/2	Bit 2, Wort 4.

• Eine Auflistung von 1769 Erweiterungs-E/A-Statuscodes finden Sie in der Bedienungsanleitung.

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).

Block kann nicht auf Gerät gelesen werden. | Blockanfang = '<Adresse>', Status = <Code>, erweiterter Status = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

Knoten wurde nicht gefunden, oder ein doppelter Knoten wurde festgestellt.

Mögliche Lösung:

Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Die Codes werden im Hexadezimalformat angezeigt.

● Hinweis:

Diese Fehlermeldung gilt für Fehler von lokalen Knoten. Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Der Treiber versucht diese Datenblöcke in regelmäßigen Abständen erneut zu lesen. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem bestimmten Grund nicht erkennen kann.

Block kann nicht auf Gerät gelesen werden. Block deaktiviert. | Blockanfang = '<Adresse>', Status = <Code>, erweiterter Status = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Die im Block angeforderte Adresse ist in der SPS nicht vorhanden.
2. Prozessor ist im Programmmodus.

Mögliche Lösung:

Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Die Codes werden im Hexadezimalformat angezeigt.

● Hinweis:

Diese Fehlermeldung gilt für Fehler vom Remote-Knoten. Statuscodefehler im High-Nibble des Statuscodes weisen auf von der SPS festgestellte Fehler hin. Diese Fehler werden generiert, wenn der vom Treiber angefragte Datenblock in der SPS nicht verfügbar ist. Der Treiber fragt nicht erneut nach diesen Blöcken, nachdem dieser Fehlertyp empfangen wurde. Dieser Fehlertyp kann generiert werden, wenn die Adresse nicht in der SPS vorhanden ist.

Block kann nicht auf Gerät gelesen werden. | Blockanfang = '<Adresse>', Status = <Code>, erweiterter Status = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

Die im Block angeforderte Adresse ist in der SPS nicht vorhanden.

Mögliche Lösung:

Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Die Codes werden im Hexadezimalformat angezeigt.

Block kann nicht auf Gerät gelesen werden. Block deaktiviert. | Blockanfang = '<Adresse>', Status = <Code>, erweiterter Status = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Die im Block angeforderte Adresse ist in der SPS nicht vorhanden.
2. Knoten wurde nicht gefunden, oder ein doppelter Knoten wurde festgestellt.

Mögliche Lösung:

Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Die Codes werden im Hexadezimalformat angezeigt.

● Hinweis:

Diese Fehlermeldung gilt für Fehler von lokalen Knoten. Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Der Treiber versucht diese Datenblöcke in regelmäßigen Abständen erneut zu lesen. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem bestimmten Grund nicht erkennen kann.

Funktionsdatei auf Gerät kann nicht gelesen werden. | Funktionsdatei = '<Funktionsdateielement>', Status = <Code>, erweiterter Status = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

Knoten wurde nicht gefunden, oder ein doppelter Knoten wurde festgestellt.

Mögliche Lösung:

Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Die Codes werden im Hexadezimalformat angezeigt.

● Hinweis:

Diese Fehlermeldung gilt für Fehler von lokalen Knoten. Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Der Treiber versucht diese Funktionsdatei in regelmäßigen Abständen zu lesen. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem bestimmten Grund nicht erkennen kann.

Funktionsdatei auf Gerät kann nicht gelesen werden. Block deaktiviert. | Funktionsdatei = '<Funktionsdateielement>', Status = <Status>, erweiterter Status = <Status>.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Die im Block angeforderte Adresse der Funktionsdatei ist in der SPS nicht vorhanden.
2. Prozessor ist im Programmmodus.

Mögliche Lösung:

Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Die Codes werden im Hexadezimalformat angezeigt.

● Hinweis:

Diese Fehlermeldung gilt für Fehler von Remote-Knoten. Statuscodefehler im High-Nibble des Statuscodes weisen auf von der SPS festgestellte Fehler hin. Diese Fehler werden generiert, wenn die vom Treiber angeforderte Funktionsdatei in der SPS nicht verfügbar ist. Der Treiber fordert diese Funktionsdatei nicht erneut an, nachdem dieser Fehlertyp empfangen wurde. Dieser Fehlertyp kann generiert werden, wenn die Adresse der Funktionsdatei nicht in der SPS vorhanden ist.

Block kann nicht auf Gerät gelesen werden. Framing-Fehler. | Blockanfang = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Unerwarteter Frame empfangen.
2. Größe stimmt nicht überein.

Mögliche Lösung:

Das Gerät hat eine Leseantwort zurückgegeben, die ungültig ist oder eine unerwartete Größe aufweist.

Funktionsdatei auf Gerät kann nicht gelesen werden. Framing-Fehler. | Funktionsdatei = '<Funktionsdateielement>'.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Unerwarteter Frame empfangen.
2. Größe stimmt nicht überein.

Mögliche Lösung:

Das Gerät hat eine Funktionsdatei zurückgegeben, die ungültig ist oder eine unerwartete Größe aufweist.

Block kann nicht auf Gerät gelesen werden. Prüfsummenfehler. | Blockanfang = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Es besteht ein Problem mit den Verbindungskabeln zwischen den Geräten, das Störungen und Prüfsummenfehler verursacht.

Mögliche Lösung:

Überprüfen Sie die Verkabelung zwischen dem Host-PC und dem Gerät.

Funktionsdatei auf Gerät kann nicht gelesen werden. Prüfsummenfehler. | Funktionsdatei = '<Funktionsdateielement>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Es besteht ein Problem mit den Verbindungskabeln zwischen den Geräten, das Störungen und Prüfsummenfehler verursacht.

Mögliche Lösung:

Überprüfen Sie die Verkabelung zwischen dem Host-PC und dem Gerät.

Block kann nicht auf Gerät gelesen werden. Slave-Senke/Quelle ist voll. | Blockanfang = '<Startadresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Das Slave-Gerät kann keine weiteren Anforderungen vom Master annehmen. Die Geschwindigkeit für die Datenanforderung vom Client ist möglicherweise zu hoch.

Mögliche Lösung:

Der Treiber ruft den Slave immer wieder automatisch ab, damit die Slave-Quelle geleert und Platz für Anforderungen in der zuvor vollen Senke freigegeben wird. Falls dieser Fehler zu häufig auftritt, verringern Sie das Aktualisierungsintervall auf den verdächtigen Tags.

Funktionsdatei auf Gerät kann nicht gelesen werden. Slave-Senke/Quelle voll. | Funktionsdatei = '<Funktionsdateielement>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Das Slave-Gerät kann keine weiteren Anforderungen vom Master annehmen. Die Geschwindigkeit für die Datenanforderung vom Client ist möglicherweise zu hoch.

Mögliche Lösung:

Der Treiber ruft den Slave immer wieder automatisch ab, damit die Slave-Quelle geleert und Platz für Anforderungen in der zuvor vollen Senke freigegeben wird. Falls dieser Fehler zu häufig auftritt, verringern Sie das Aktualisierungsintervall auf den verdächtigen Funktionsdatei-Tags.

Block kann nicht auf Gerät gelesen werden. Slave-Quelle leer. | Blockanfang = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Das Slave-Gerät hat keine Antwort auf die Datenanforderung vorbereitet. Möglicherweise ist der Wert für die Slave-Abrufverzögerung zu niedrig eingestellt.

Mögliche Lösung:

Der Treiber ruft den Slave immer wieder auf eine Abrufantwort automatisch ab. Falls dieser Fehler häufig auftritt, erhöhen Sie die Slave-Abrufverzögerung im Kanal.

Funktionsdatei auf Gerät kann nicht gelesen werden. Slave-Quelle leer. | Funktionsdatei = '<Funktionsdateielement>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Das Slave-Gerät hat keine Antwort auf die Anforderung des Funktionsdateielements vorbereitet. Möglicherweise ist der Wert für die Slave-Abrufverzögerung zu niedrig eingestellt.

Mögliche Lösung:

Der Treiber ruft den Slave immer wieder auf eine Abrufantwort automatisch ab. Falls dieser Fehler häufig auftritt, erhöhen Sie die Slave-Abrufverzögerung im Kanal.

Fehler beim Schreiben in Adresse. | Tag-Adresse = '<Adresse>', Status = <Status>, erweiterter Status = <Status>.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Knoten wurde nicht gefunden, oder ein doppelter Knoten wurde festgestellt.
2. Die im Block angeforderte Adresse ist in der SPS nicht vorhanden.
3. Prozessor ist im Programmmodus.

Mögliche Lösung:

Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Die Codes werden im Hexadezimalformat angezeigt.

● Hinweis:

1. Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem bestimmten Grund nicht erkennen kann.
2. Statuscodefehler im High-Nibble weisen auf von der SPS festgestellte Fehler hin. Diese Fehler werden generiert, wenn der vom Treiber angeforderte Datenblock in der SPS nicht verfügbar ist.

Fehler beim Schreiben in Adresse. Framing-Fehler. | Tag-Adresse = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Unerwarteter Frame empfangen.
2. Größe stimmt nicht überein.

Mögliche Lösung:

Das Gerät hat eine Schreibantwort zurückgegeben, die ungültig ist oder eine unerwartete Größe aufweist.

Prüfsummenfehler beim Schreiben in Adresse. | Tag-Adresse = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Es besteht ein Problem mit den Verbindungskabeln zwischen den Geräten, das Störungen und Prüfsummenfehler verursacht.

Mögliche Lösung:

Überprüfen Sie die Verkabelung zwischen dem Host-PC und dem Gerät, und beheben Sie gegebenenfalls Fehler.

Fehler beim Schreiben in Adresse. Slave-Senke/Quelle voll. | Tag-Adresse = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Das Slave-Gerät kann keine weiteren Anforderungen vom Master annehmen. Die Geschwindigkeit für die Datenanforderung vom Client ist möglicherweise zu hoch.

Mögliche Lösung:

Der Treiber ruft den Slave immer wieder automatisch ab, damit die Slave-Quelle geleert und Platz für Anforderungen in der zuvor vollen Senke freigegeben wird. Falls dieser Fehler häufig auftritt, verringern Sie das Aktualisierungsintervall auf den verdächtigen Tags, jedoch nicht unbedingt auf dem Tag, in das geschrieben wird.

Fehler beim Schreiben in Adresse. Slave-Quelle leer. | Tag-Adresse = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Das Slave-Gerät hat keine Antwort auf die Schreibenanforderung vorbereitet. Möglicherweise ist der Wert für die Slave-Abfrageverzögerung zu niedrig.

Mögliche Lösung:

Der Treiber ruft den Slave immer wieder auf eine Abfrageantwort automatisch ab. Falls dieser Fehler häufig auftritt, erhöhen Sie die Slave-Abfrageverzögerung im Kanal.

Timeout von Gerät beim Schreiben in Adresse. | Tag-Adresse = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Das Gerät antwortet nicht.

Mögliche Lösung:

Überprüfen Sie die Verkabelung zwischen dem Host-PC und dem Gerät. Vergewissern Sie sich, dass das Gerät eingeschaltet ist und ordnungsgemäß funktioniert.

Block kann nicht auf Gerät gelesen werden. | Blockanfang = '<Adresse>', Status = <Code>, erweiterter Status = <Code>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Knoten wurde nicht gefunden, oder doppelter Knoten wurde festgestellt.

Mögliche Lösung:

Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Die Codes werden im Hexadezimalformat angezeigt.

● Hinweis:

Diese Fehlermeldung gilt für Fehler von lokalen Knoten. Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Der Treiber versucht diese Datenblöcke in regelmäßigen Abständen erneut zu lesen. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem bestimmten Grund nicht erkennen kann.

Block kann nicht auf Gerät gelesen werden. Block deaktiviert. | Blockanfang = '<Adresse>', Status = <Code>, erweiterter Status = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

Die im Block angeforderte Adresse ist in der SPS nicht vorhanden.

Mögliche Lösung:

Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Die Codes werden im Hexadezimalformat angezeigt.

● **Hinweis:**

Diese Fehlermeldung gilt für Fehler vom Remote-Knoten. Statuscodefehler im High-Nibble des Statuscodes weisen auf von der SPS festgestellte Fehler hin. Diese Fehler werden generiert, wenn der vom Treiber angefragte Datenblock in der SPS nicht verfügbar ist. Der Treiber fragt nicht erneut nach diesen Blöcken, nachdem dieser Fehlertyp empfangen wurde. Dieser Fehlertyp kann generiert werden, wenn die Adresse nicht in der SPS vorhanden ist.

Block kann nicht auf Gerät gelesen werden. Gerät hat mit NAK geantwortet. | Blockanfang = '<Adresse>'.

Fehlertyp:

Warnung

Funktionsdatei auf Gerät kann nicht gelesen werden. Gerät hat mit NAK geantwortet. | Funktionsdatei = '<Funktionsdateielement>'.

Fehlertyp:

Warnung

Block kann nicht auf Gerät gelesen werden. Speicherzuordnungsfehler. | Blockanfang = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Beim Schreiben in Speicher auf Server ist ein Fehler aufgetreten.

Mögliche Lösung:

Vergewissern Sie sich, dass das Gerät online ist, kommunizieren kann und über eine gültige/fehlerfreie Speicherzuordnung verfügt, bevor Sie den Vorgang erneut durchführen.

Funktionsdatei auf Gerät kann nicht gelesen werden. Speicherzuordnungsfehler. | Funktionsdatei = '<Funktionsdateielement>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Beim Zugriff auf die Speicherzuordnung im Gerät ist ein Fehler aufgetreten.

Mögliche Lösung:

Vergewissern Sie sich, dass das Gerät online ist, kommunizieren kann und über eine gültige/fehlerfreie Speicherzuordnung verfügt, bevor Sie den Vorgang erneut durchführen.

Funktionsdatei auf Gerät kann nicht gelesen werden. Gerät hat unerwarteterweise mit NAK geantwortet. Überprüfen Sie das Verknüpfungsprotokoll des Geräts. | Funktionsdatei = '<Funktionsdateielement>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Der Server hat eine Antwort gesendet, die ungültig ist oder eine unerwartete Größe aufweist. Die Verknüpfungseinstellungen für das Protokoll des Servers stimmen möglicherweise nicht mit der Gerätekonfiguration überein.

Mögliche Lösung:

Korrigieren Sie die Einstellungen für das Verknüpfungsprotokoll des Geräts, sodass sie den Einstellungen des Kanals entsprechen, und führen Sie den Vorgang erneut durch.

Block kann nicht auf Gerät gelesen werden. Gerät hat unerwarteterweise mit NAK geantwortet. Überprüfen Sie das Verknüpfungsprotokoll des Geräts. | Blockanfang = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Der Server hat eine Antwort gesendet, die ungültig ist oder eine unerwartete Größe aufweist. Die Verknüpfungseinstellungen für das Protokoll des Servers stimmen möglicherweise nicht mit der Gerätekonfiguration überein.

Mögliche Lösung:

Korrigieren Sie die Einstellungen für das Verknüpfungsprotokoll des Geräts, sodass sie den Einstellungen des Kanals entsprechen, und führen Sie den Vorgang erneut durch.

In Adresse auf Gerät kann nicht geschrieben werden. Paketlänge liegt außerhalb des zulässigen Bereichs. | Tag-Adresse = '<Adresse>', Paketlängenbereich = <Min.> bis <Max.> (Byte).

Fehlertyp:

Informationen

Mögliche Ursache:

Es besteht ein Fehler bei der Paketlänge.

Mögliche Lösung:

Korrigieren Sie die Paketlänge, sodass sie im unterstützten Bereich liegt, und führen Sie den Vorgang erneut durch.

In Adresse auf Gerät kann nicht geschrieben werden. TNS liegt außerhalb des zulässigen Bereichs. | Tag-Adresse = '<Adresse>', TNS-Bereich = <Min.> bis <Max.>.

Fehlertyp:

Informationen

Mögliche Ursache:

Eine ungültige Adresse wurde angegeben.

Mögliche Lösung:

Vergewissern Sie sich, dass die Adresse im unterstützten Bereich liegt, und führen Sie den Vorgang erneut durch.

Fehlermaskendefinitionen

B = Hardwareunterbrechung festgestellt

F = Framing-Fehler

E = E/A-Fehler

O = Zeichenpufferüberlauf

R = RX-Pufferüberlauf

P = Erhaltener Byte-Paritätsfehler

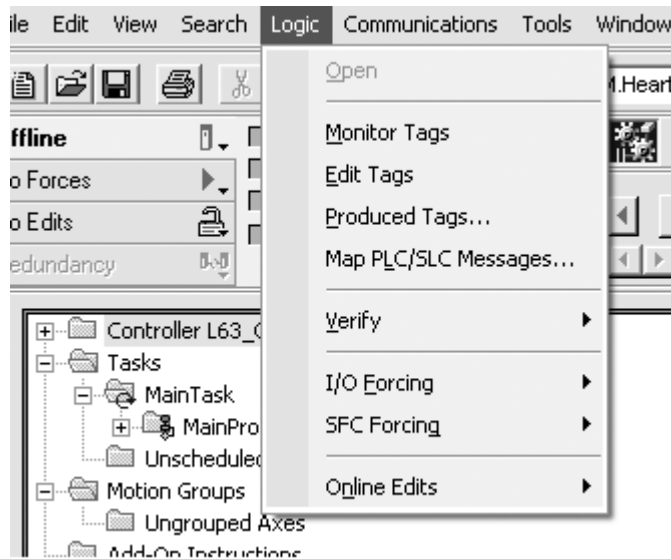
T = TX-Puffer voll

Appendix: Communicating with RSLogix5000 Family Controllers

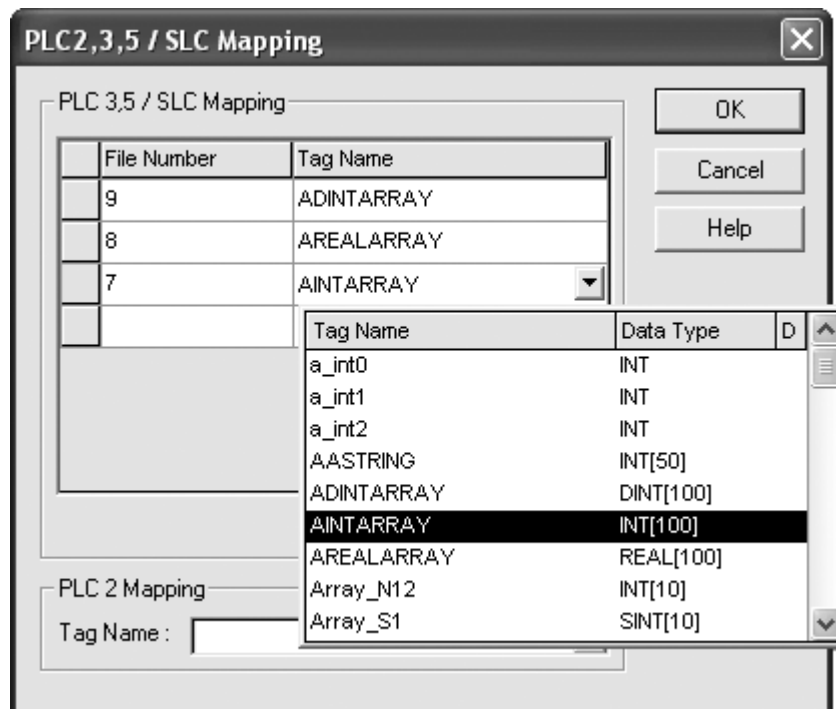
The DF1-Treiber von Allen-Bradley can be used to connect to an RSLogix5000 controller's serial port. Users must configure the RSLogix5000 Controller Tags to be mapped to the PLC/SLC data tables. For information on enabling communications, refer to the instructions below.

● **Note:** A NULL Modem cable is required for this connection.

1. To start, open an RSLogix5000 development software application (such as RSLogix5000 or Studio 5000 Logix Designer). Then, select **Logic | Map PLC/SLC Messages**.

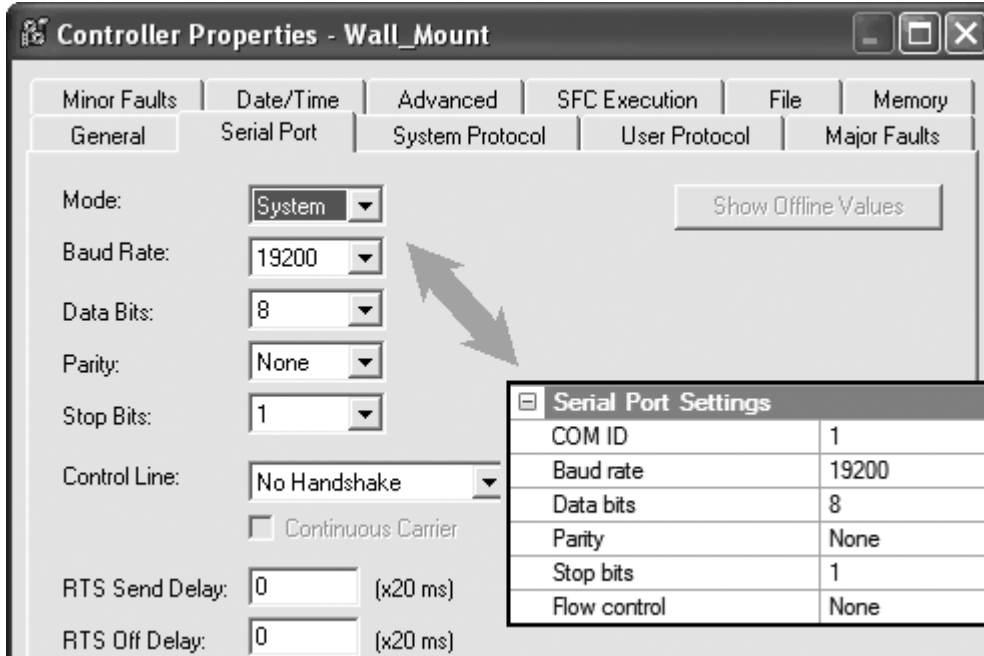


2. Next, map the tags to the different data files as shown in the image below.



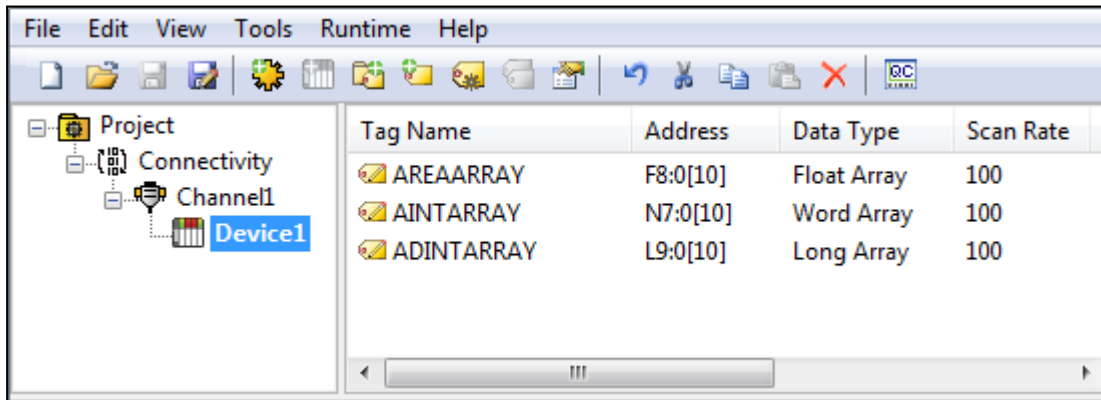
- 3. Ensure that the DF1-Treiber von Allen-Bradley and controller have matching settings for the Serial Port and System Protocol (such as the baud rate, data bits, and parity properties shown in the images below).

● **Note:** The Link Protocol should be set to **Full Duplex**.



- 4. Next, create the desired tags within the DF1-Treiber von Allen-Bradley.

● **Note:** The MicroLogix model is the only model that supports "L" data types, which are 32-bit data types equivalent to the controller's DINT data type.



Index

A

Adressbeschreibungen 30
Alle Datenanfragen im Scan-Intervall 19
Anfangsaktualisierungen aus Cache 19
Anforderungs-Timeout 21
Anforderungsgröße 22
Anfragen verwerfen, wenn herabgestuft 22
Antworten von anderen Stationen ignorieren 14
ASCII-Dateien 41
Ausgabedateien 30
Avtron 23

B

BCD 29
BCD-Dateien 43
Binäre Dateien 37
Block kann nicht auf Gerät gelesen werden. | Blockanfang = '<Adresse>', Status = <Code>, erweiterter Status = <Code>. 53, 58
Block kann nicht auf Gerät gelesen werden. Block deaktiviert. | Blockanfang = '<Adresse>', Status = <Code>, erweiterter Status = <Code>. 53-54, 59
Block kann nicht auf Gerät gelesen werden. Framing-Fehler. | Blockanfang = '<Adresse>'. 55
Block kann nicht auf Gerät gelesen werden. Gerät hat mit NAK geantwortet. | Blockanfang = '<Adresse>'. 59
Block kann nicht auf Gerät gelesen werden. Gerät hat unerwarteterweise mit NAK geantwortet. Überprüfen Sie das Verknüpfungsprotokoll des Geräts. | Blockanfang = '<Adresse>'. 60
Block kann nicht auf Gerät gelesen werden. Prüfsummenfehler. | Blockanfang = '<Adresse>'. 55
Block kann nicht auf Gerät gelesen werden. Slave-Quelle leer. | Blockanfang = '<Adresse>'. 56
Block kann nicht auf Gerät gelesen werden. Slave-Senke/Quelle ist voll. | Blockanfang = '<Startadresse>'. 56
Block kann nicht auf Gerät gelesen werden. Speicherzuordnungsfehler. | Blockanfang = '<Adresse>'. 59
Blocktransferdateien 48
Boolean 29
Byte 29

C

Char 29
Communicating with RSLogix5000 Family Controllers 62

D

Datenanfrage nicht schneller als Scan-Intervall 19
Datensammlung 18
Datentypbeschreibung 29
Durch Tag angegebenes Scan-Intervall berücksichtigen 19
DWord 29

E

Eingabedateien 33
Ereignisprotokollmeldungen 53

F

Fehler beim Schreiben in Adresse. | Tag-Adresse = '<Adresse>', Status = <Status>, erweiterter Status = <Status>. 57
Fehler beim Schreiben in Adresse. Framing-Fehler. | Tag-Adresse = '<Adresse>'. 57
Fehler beim Schreiben in Adresse. Slave-Quelle leer. | Tag-Adresse = '<Adresse>'. 58
Fehler beim Schreiben in Adresse. Slave-Senke/Quelle voll. | Tag-Adresse = '<Adresse>'. 58
Fehlermaskendefinitionen 61
Fehlerprüfung 22
Float 22, 29
Float-Dateien 41
Framing 61
Funkmodem 14
Funktionsdatei auf Gerät kann nicht gelesen werden. | Funktionsdatei = '<Funktionsdateielement>', Status = <Code>, erweiterter Status = <Code>. 54
Funktionsdatei auf Gerät kann nicht gelesen werden. Block deaktiviert. | Funktionsdatei = '<Funktionsdateielement>', Status = <Status>, erweiterter Status = <Status>. 54
Funktionsdatei auf Gerät kann nicht gelesen werden. Framing-Fehler. | Funktionsdatei = '<Funktionsdateielement>'. 55
Funktionsdatei auf Gerät kann nicht gelesen werden. Gerät hat mit NAK geantwortet. | Funktionsdatei = '<Funktionsdateielement>'. 59
Funktionsdatei auf Gerät kann nicht gelesen werden. Gerät hat unerwarteterweise mit NAK geantwortet. Überprüfen Sie das Verknüpfungsprotokoll des Geräts. | Funktionsdatei = '<Funktionsdateielement>'. 60
Funktionsdatei auf Gerät kann nicht gelesen werden. Prüfsummenfehler. | Funktionsdatei = '<Funktionsdateielement>'. 55
Funktionsdatei auf Gerät kann nicht gelesen werden. Slave-Quelle leer. | Funktionsdatei = '<Funktionsdateielement>'. 56
Funktionsdatei auf Gerät kann nicht gelesen werden. Slave-Senke/Quelle voll. | Funktionsdatei = '<Funktionsdateielement>'. 56
Funktionsdatei auf Gerät kann nicht gelesen werden. Speicherzuordnungsfehler. | Funktionsdatei = '<Funktionsdateielement>'. 59

Funktionsdateioptionen 23

G

Ganzzahldateien 40

Geräte-Setup 17

Geräteeigenschaften - Automatische Herabstufung 21

Geräteeigenschaften - Ethernet-Kapselung 19

Geräteeigenschaften - Protokolleinstellungen 22

Gleitkommazahl-Wörter 23

Gleitkommazahlwörter tauschen 22

H

Halbduplex 14

Halbduplex-Master 14

Herabstufen bei Fehler 21

Herabstufungszeitraum 22

HSC-Datei (High Speed Counter) 49

I

ID 18

ID-Format 18

In Adresse auf Gerät kann nicht geschrieben werden. Paketlänge liegt außerhalb des zulässigen Bereichs. | Tag-Adresse = '<Adresse>', Paketlängenbereich = <Min.> bis <Max.> (Byte). 60

In Adresse auf Gerät kann nicht geschrieben werden. TNS liegt außerhalb des zulässigen Bereichs. | Tag-Adresse = '<Adresse>', TNS-Bereich = <Min.> bis <Max.>. 60

Inhalt der Hilfe 5

IOS-Datei (I/O Module Status, E/A-Modulstatus) 52

IP-Adresse 20

K

Kabelverbindungen 6

Kanal-Setup 7

Kanaleigenschaften - Verknüpfungseinstellungen 13

Kanalzuweisung 18

KF2/KF3-Halbduplex-Master 14-15

Kommunikations-Timeouts 20-21

Kommunikationsstatusdatei für Kanal 0 (Channel 0 Communication Status File) 51

Kommunikationsstatusdatei für Kanal 1 (Channel 1 Communication Status File) 51

L

LBCD 29
Long 29
Long-Dateien 43

M

MicroLogix-Meldungsdateien 46
MicroLogix-PID-Dateien 44
Modell 18

N

N-Datei-Gleitkommazahlzugriff 22
Nicht scannen, nur Abruf anfordern 19

P

Parität 61
PLC5-Meldungsdateien 47
PLC5-PID-Dateien 45
Port 20
Protokoll 20
Prüfsumme 22
Prüfsummenfehler beim Schreiben in Adresse. | Tag-Adresse = '<Adresse>'. 57

R

Redundanz 28
Richtlinien für modulare E/A-Auswahl 26
RTC-Datei (Real Time Clock, Echtzeituhr) 50

S

Scan-Modus 19
Short 29
Simuliert 18
Slave-Abrufverzögerung 14
Stations-ID 13
Statusdateien 37

Steckplatzkonfiguration 25
Steuerdateien 39
String-Dateien 42

T

Timeout bis zum Herabstufen 22
Timeout von Gerät beim Schreiben in Adresse. | Tag-Adresse = '<Adresse>'. 58
Treiber 18

U

Überlauf 61
Übersicht 5
Unterstützte Geräte 17
Unterstützte Protokolle 7

V

Verbindungs-Timeout 20
Verbindungsversuche 21
Verknüpfungsprotokolle 14
Versuche vor Timeout 21
Verzögerung zwischen Anfragen 21
Vollduplex 14
Vom Client angegebenes Scan-Intervall berücksichtigen 19

W

Word 29

Z

Zählerdateien 39
Zeichenfolge 29
Zeitgeberdateien 38