

# Modicon LMC078

## Motion Controller

## Programmierhandbuch

03/2018



EIO0000001911.04

[www.schneider-electric.com](http://www.schneider-electric.com)

**Schneider**  
Electric

---

Die Informationen in der vorliegenden Dokumentation enthalten allgemeine Beschreibungen und/oder technische Leistungsmerkmale der hier erwähnten Produkte. Diese Dokumentation dient keinesfalls als Ersatz für die Ermittlung der Eignung oder Verlässlichkeit dieser Produkte für bestimmte Verwendungsbereiche des Benutzers und darf nicht zu diesem Zweck verwendet werden. Jeder Benutzer oder Integrator ist verpflichtet, angemessene und vollständige Risikoanalysen, Bewertungen und Tests der Produkte im Hinblick auf deren jeweils spezifischen Verwendungszweck vorzunehmen. Weder Schneider Electric noch deren Tochtergesellschaften oder verbundene Unternehmen sind für einen Missbrauch der Informationen in der vorliegenden Dokumentation verantwortlich oder können diesbezüglich haftbar gemacht werden. Verbesserungs- und Änderungsvorschläge sowie Hinweise auf angetroffene Fehler werden jederzeit gern entgegengenommen.

Sie erklären, dass Sie ohne schriftliche Genehmigung von Schneider Electric dieses Dokument weder ganz noch teilweise auf beliebigen Medien reproduzieren werden, ausgenommen zur Verwendung für persönliche nichtkommerzielle Zwecke. Darüber hinaus erklären Sie, dass Sie keine Hypertext-Links zu diesem Dokument oder seinem Inhalt einrichten werden. Schneider Electric gewährt keine Berechtigung oder Lizenz für die persönliche und nichtkommerzielle Verwendung dieses Dokument oder seines Inhalts, ausgenommen die nichtexklusive Lizenz zur Nutzung als Referenz. Das Handbuch wird hierfür „wie besehen“ bereitgestellt, die Nutzung erfolgt auf eigene Gefahr. Alle weiteren Rechte sind vorbehalten.

Bei der Montage und Verwendung dieses Produkts sind alle zutreffenden staatlichen, landesspezifischen, regionalen und lokalen Sicherheitsbestimmungen zu beachten. Aus Sicherheitsgründen und um die Übereinstimmung mit dokumentierten Systemdaten besser zu gewährleisten, sollten Reparaturen an Komponenten nur vom Hersteller vorgenommen werden.

Beim Einsatz von Geräten für Anwendungen mit technischen Sicherheitsanforderungen sind die relevanten Anweisungen zu beachten.

Die Verwendung anderer Software als der Schneider Electric-eigenen bzw. einer von Schneider Electric genehmigten Software in Verbindung mit den Hardwareprodukten von Schneider Electric kann Körperverletzung, Schäden oder einen fehlerhaften Betrieb zur Folge haben.

Die Nichtbeachtung dieser Informationen kann Verletzungen oder Materialschäden zur Folge haben!

© 2018 Schneider Electric. Alle Rechte vorbehalten.



	<b>Sicherheitshinweise</b> .....	<b>7</b>
	<b>Über dieses Buch</b> .....	<b>9</b>
<b>Kapitel 1</b>	<b>Informationen zum Modicon LMC078 Motion Controller</b> .....	<b>15</b>
	Informationen zum Modicon LMC078 Motion Controller .....	<b>16</b>
	Architektur für verteilte E/A .....	<b>18</b>
<b>Kapitel 2</b>	<b>Konfigurieren der Steuerung</b> .....	<b>21</b>
	Konfiguration der Steuerung .....	<b>21</b>
<b>Kapitel 3</b>	<b>Bibliotheken</b> .....	<b>25</b>
	Bibliotheken .....	<b>25</b>
<b>Kapitel 4</b>	<b>Unterstützte Standarddatentypen</b> .....	<b>27</b>
	Unterstützte Standarddatentypen .....	<b>28</b>
	Parametertypen .....	<b>29</b>
<b>Kapitel 5</b>	<b>Speicherorganisation</b> .....	<b>31</b>
	Organisation des Steuerungsspeichers .....	<b>32</b>
	Aufbau des RAM-Speichers .....	<b>33</b>
	Flash-Speicheranordnung .....	<b>35</b>
	USB-Speicherstick .....	<b>36</b>
<b>Kapitel 6</b>	<b>Tasks</b> .....	<b>37</b>
	Maximale Anzahl an Tasks .....	<b>38</b>
	Konfigurationsfenster der Tasks .....	<b>39</b>
	Tasktypen .....	<b>41</b>
	Bewegungstask .....	<b>44</b>
	System- und Task-Watchdogs .....	<b>47</b>
	Taskprioritäten .....	<b>48</b>
	Standard-Taskkonfiguration .....	<b>51</b>
<b>Kapitel 7</b>	<b>Steuerungszustände und Verhalten</b> .....	<b>53</b>
7.1	Diagramm der Steuerungszustände .....	<b>54</b>
	Diagramm der Steuerungszustände .....	<b>54</b>
7.2	Beschreibung der Steuerungszustände .....	<b>58</b>
	Beschreibung der Steuerungszustände .....	<b>58</b>
7.3	Zustandsübergänge und Systemereignisse .....	<b>62</b>
	Steuerungszustände und Ausgangsverhalten .....	<b>63</b>
	Befehlen von Zustandswechseln .....	<b>66</b>
	Fehlererkennung, Fehlertypen und Fehlerhandhabung .....	<b>71</b>
	Remanente Variablen .....	<b>72</b>

<b>Kapitel 8</b>	<b>Geräte-Editor der Steuerung</b> .....	<b>75</b>
	Steuerungsparameter .....	76
	Konfigurationsparameter .....	78
	Steuerungsauswahl .....	88
	SPS-Einstellungen .....	91
<b>Kapitel 9</b>	<b>Konfiguration integrierter Ein- und Ausgänge</b> .....	<b>93</b>
	Integrierte E/A-Konfiguration .....	94
	Master-Geber-Eingangskonfiguration .....	102
<b>Kapitel 10</b>	<b>Kommunikationsmodule</b> .....	<b>111</b>
10.1	Konfiguration des PROFIBUS DP-Slave-Moduls .....	112
	Hinzufügen eines PROFIBUS DP-Slave-Moduls .....	113
	Konfiguration des PROFIBUS DP-Slave-Moduls .....	115
	Azyklischer Datenaustausch .....	120
10.2	Konfiguration des EtherNet/IP-Adapters .....	123
	Konfiguration des EtherNet/IP-Adapters .....	124
	Zyklischer Datenaustausch .....	128
	Azyklischer Datenaustausch .....	129
10.3	Konfiguration eines Ethernet/IP-Scanners .....	134
	Beschreibung .....	135
	Unterstützte Geräte .....	136
	Konfiguration des EtherNet/IP Scanners .....	138
	EtherNet/IP Scanner-E/A-Abbild .....	140
	Status und Diagnose des EtherNet/IP Scanners .....	141
	Deklaration des Zielgeräts .....	143
	Zieleinstellungen .....	145
	Verbindungskonfiguration .....	147
	Austausch von Geräten über Benutzerparameter .....	165
	EtherNet/IP-E/A-Abbild .....	169
<b>Kapitel 11</b>	<b>Konfiguration der Ethernet-Verbindung</b> .....	<b>171</b>
11.1	Ethernet-Dienste .....	172
	Beschreibung .....	173
	Konfiguration der IP-Adresse .....	175
	Modbus TCP Client/Server .....	180
	FTP-Server .....	182
	FTP-Client .....	185
	LMC078 Motion Controller als IOScanner-Slave-Gerät in Modbus TCP-Netzwerken .....	186

11.2	Firewallkonfiguration .....	191
	Einführung .....	192
	Verhalten der Firewall .....	194
	Skriptbefehle für die Firewall .....	195
	Skriptdateien .....	199
<b>Kapitel 12</b>	<b>CANopen-Konfiguration .....</b>	<b>201</b>
	CANopen-Schnittstellenkonfiguration .....	202
	Konfiguration des CANopen-Masters .....	203
	Konfiguration eines CANopen-Slaves .....	205
<b>Kapitel 13</b>	<b>SercosKonfiguration .....</b>	<b>207</b>
	Überblick über den Sercos-Standard .....	208
	Sercos Schnittstellenkonfiguration .....	211
	Sercos-Geräte .....	215
	Editor für die Geräteadressierung .....	216
	Lexium LXM32S-Antriebskonfiguration .....	220
	TM5NS31 Sercos Interface Module .....	223
	Sercos Fehlercodes .....	224
<b>Kapitel 14</b>	<b>Konfiguration der seriellen Leitung .....</b>	<b>229</b>
	Konfiguration der seriellen Leitung .....	230
	ASCII-Manager .....	232
	SoMachine-Netzwerkmanager .....	234
	Modbus Serial-E/A-Scanner .....	235
	Hinzufügen eines Geräts auf dem Modbus Serial-E/A-Scanner .....	237
	Modbus-Manager .....	244
	Hinzufügen eines Modems zu einem Manager .....	249
<b>Kapitel 15</b>	<b>Anschließen von Modicon LMC078 Motion Controller an einen PC .....</b>	<b>251</b>
	Verbindung der Steuerung mit einem PC .....	251
<b>Kapitel 16</b>	<b>Firmwareaktualisierung .....</b>	<b>255</b>
	Aktualisieren von Modicon LMC078 Motion Controller-Firmware .....	255
<b>Anhang</b>	.....	<b>259</b>
<b>Anhang A</b>	<b>Vorgehensweise zur Änderung der IP-Adresse der Steuerung .....</b>	<b>261</b>
	changeIPAddress: Ändern der IP-Adresse der Steuerung .....	261
<b>Anhang B</b>	<b>Diagnosemeldungen .....</b>	<b>265</b>
	Meldungslogger .....	266
	Diagnosemeldungen .....	272

---

<b>Anhang C</b>	<b>LMC078 Sercos3-Bibliothek</b> . . . . .	<b>281</b>
C.1	Datentypen . . . . .	<b>282</b>
	ST_SercosConfiguration-Datentyp . . . . .	<b>283</b>
	ST_SercosConfigurationDevice-Datentyp . . . . .	<b>284</b>
	ET_Sercos3CmdType-Datentyp . . . . .	<b>286</b>
	ET_Sercos3IDNTType-Datentyp . . . . .	<b>287</b>
C.2	Sercos-Funktionen . . . . .	<b>288</b>
	FC_SercosGetConfiguration-Funktion . . . . .	<b>289</b>
	FC_SercosReadServiceData-Funktion . . . . .	<b>290</b>
	FC_SercosReadServiceDataByTopAddr-Funktion . . . . .	<b>293</b>
	FC_SercosScanConfiguration-Funktion . . . . .	<b>296</b>
	FC_SercosWriteServiceData-Funktion . . . . .	<b>298</b>
	FC_SercosWriteServiceDataByTopAddr-Funktion . . . . .	<b>300</b>
C.3	Asynchrone Sercos-Funktionsbausteine . . . . .	<b>302</b>
	FB_SercosReadServiceDataAsync : Asynchrones Lesen von Daten über die Sercos-Schnittstelle . . . . .	<b>303</b>
	FB_SercosWriteServiceDataAsync : Asynchrones Schreiben von Daten über die Sercos-Schnittstelle . . . . .	<b>305</b>
	FB_SercosProcedureCommandAsync: Asynchrones Senden von Befehlen über die Sercos-Schnittstelle . . . . .	<b>307</b>
<b>Anhang D</b>	<b>Funktionen zum Abrufen/Einrichten der Konfiguration der seriellen Leitung in einem Anwenderprogramm</b> . . . . .	<b>309</b>
	GetSerialConf: Abrufen der Konfiguration der seriellen Leitung . . . . .	<b>310</b>
	SetSerialConf: Ändern der Konfiguration der seriellen Leitung . . . . .	<b>311</b>
	SERIAL_CONF: Struktur des Datentyps für die serielle Leitungskonfiguration . . . . .	<b>313</b>
<b>Anhang E</b>	<b>SPS-Leistung</b> . . . . .	<b>315</b>
	Verarbeitungsleistung . . . . .	<b>315</b>
<b>Glossar</b>	. . . . .	<b>317</b>
<b>Index</b>	. . . . .	<b>329</b>



## Wichtige Informationen

### HINWEISE

Lesen Sie sich diese Anweisungen sorgfältig durch und machen Sie sich vor Installation, Betrieb, Bedienung und Wartung mit dem Gerät vertraut. Die nachstehend aufgeführten Warnhinweise sind in der gesamten Dokumentation sowie auf dem Gerät selbst zu finden und weisen auf potenzielle Risiken und Gefahren oder bestimmte Informationen hin, die eine Vorgehensweise verdeutlichen oder vereinfachen.



Wird dieses Symbol zusätzlich zu einem Sicherheitshinweis des Typs „Gefahr“ oder „Warnung“ angezeigt, bedeutet das, dass die Gefahr eines elektrischen Schlags besteht und die Nichtbeachtung der Anweisungen unweigerlich Verletzung zur Folge hat.



Dies ist ein allgemeines Warnsymbol. Es macht Sie auf mögliche Verletzungsgefahren aufmerksam. Beachten Sie alle unter diesem Symbol aufgeführten Hinweise, um Verletzungen oder Unfälle mit Todesfälle zu vermeiden.

## **GEFAHR**

**GEFAHR** macht auf eine gefährliche Situation aufmerksam, die, wenn sie nicht vermieden wird, Tod oder schwere Verletzungen **zur Folge hat**.

## **WARNUNG**

**WARNUNG** macht auf eine gefährliche Situation aufmerksam, die, wenn sie nicht vermieden wird, Tod oder schwere Verletzungen **zur Folge haben kann**.

## **VORSICHT**

**VORSICHT** macht auf eine gefährliche Situation aufmerksam, die, wenn sie nicht vermieden wird, leichte Verletzungen **zur Folge haben kann**.

## **HINWEIS**

**HINWEIS** gibt Auskunft über Vorgehensweisen, bei denen keine Verletzungen drohen.

---

## **BITTE BEACHTEN**

Elektrische Geräte dürfen nur von Fachpersonal installiert, betrieben, bedient und gewartet werden. Schneider Electric haftet nicht für Schäden, die durch die Verwendung dieses Materials entstehen.

Als qualifiziertes Fachpersonal gelten Mitarbeiter, die über Fähigkeiten und Kenntnisse hinsichtlich der Konstruktion und des Betriebs elektrischer Geräte und deren Installation verfügen und eine Schulung zur Erkennung und Vermeidung möglicher Gefahren absolviert haben.

---

# Über dieses Buch

---



## Auf einen Blick

### Ziel dieses Dokuments

Das vorliegende Dokument soll Sie bei der Programmierung und Bedienung des Modicon LMC078 Motion Controller unterstützen.

**HINWEIS:** Lesen Sie sich dieses Dokument sowie alle zugehörigen Dokumente (*siehe Seite 9*) gründlich durch, bevor Sie den Modicon LMC078 Motion Controller installieren, verwenden, warten oder außer Betrieb nehmen.

Benutzer des Modicon LMC078 Motion Controller sollten das gesamte Dokument durcharbeiten, um sicherzugehen, dass alle Funktionen verstanden werden.

**HINWEIS:** Für dieses Dokument ist *Echtzeit* als Verarbeitung definiert, die Daten in derselben Geschwindigkeit aktualisieren wie empfangen kann.

### Gültigkeitsbereich

Dieses Dokument wurde für das TM3TI4D-Add-on für die SoMachine-Version V4.3. aktualisiert.

### Weiterführende Dokumentation


Titel der Dokumentation	Referenz-Nummer
SoMachine – Programmierhandbuch	<a href="#">EIO0000000067 (ENG)</a> <a href="#">EIO0000000069 (FRE)</a> <a href="#">EIO0000000068 (GER)</a> <a href="#">EIO0000000071 (SPA)</a> <a href="#">EIO0000000070 (ITA)</a> <a href="#">EIO0000000072 (CHS)</a>
Modicon LMC078 Motion Controller – Hardwarehandbuch	<a href="#">EIO0000001925 (ENG)</a> <a href="#">EIO0000001926 (FRE)</a> <a href="#">EIO0000001927 (GER)</a> <a href="#">EIO0000001928 (SPA)</a> <a href="#">EIO0000001929 (ITA)</a> <a href="#">EIO0000001930 (CHS)</a> <a href="#">EIO0000001932 (TUR)</a>

Titel der Dokumentation	Referenz-Nummer
Modicon LMC078 Motion Controller Systemfunktionen und Variablen – PLCSystem-Bibliothekshandbuch	<a href="#">EIO0000001917 (ENG)</a> <a href="#">EIO0000001918 (FRE)</a> <a href="#">EIO0000001919 (GER)</a> <a href="#">EIO0000001920 (SPA)</a> <a href="#">EIO0000001921 (ITA)</a> <a href="#">EIO0000001922 (CHS)</a> <a href="#">EIO0000001924 (TUR)</a>
SoMachine - Motion Control - Bibliothekshandbuch	<a href="#">EIO0000002221 (ENG)</a> <a href="#">EIO0000002222 (GER)</a> <a href="#">EIO0000002223 (CHS)</a>
Modicon Flexibles TM5/TM7-System - Planungs- und Installationshandbuch	<a href="#">EIO0000000426 (ENG)</a> <a href="#">EIO0000000427 (FRE)</a> <a href="#">EIO0000000428 (GER)</a> <a href="#">EIO0000000429 (SPA)</a> <a href="#">EIO0000000430 (ITA)</a> <a href="#">EIO0000000431 (CHS)</a>
Modicon TM5 Konfiguration der Erweiterungsmodule - Programmierhandbuch	<a href="#">EIO0000000420 (ENG)</a> <a href="#">EIO0000000421 (FRE)</a> <a href="#">EIO0000000422 (GER)</a> <a href="#">EIO0000000423 (SPA)</a> <a href="#">EIO0000000424 (ITA)</a> <a href="#">EIO0000000425 (CHS)</a>
Modicon TM7 Konfiguration von Erweiterungsbausteinen - Programmierhandbuch	<a href="#">EIO0000000880 (ENG)</a> <a href="#">EIO0000000881 (FRE)</a> <a href="#">EIO0000000882 (GER)</a> <a href="#">EIO0000000883 (SPA)</a> <a href="#">EIO0000000884 (ITA)</a> <a href="#">EIO0000000885 (CHS)</a>
SoMachine Modbus- und ASCII-Lese-/Schreibfunktionen – PLCCommunication-Bibliothekshandbuch	<a href="#">EIO0000000361 (ENG)</a> <a href="#">EIO0000000362 (FRE)</a> <a href="#">EIO0000000363 (GER)</a> <a href="#">EIO0000000364 (SPA)</a> <a href="#">EIO0000000365 (ITA)</a> <a href="#">EIO0000000366 (CHS)</a>
SoMachine Modemfunktionen – Modem-Bibliothekshandbuch	<a href="#">EIO0000000552 (ENG)</a> <a href="#">EIO0000000491 (FRE)</a> <a href="#">EIO0000000492 (GER)</a> <a href="#">EIO0000000493 (SPA)</a> <a href="#">EIO0000000494 (ITA)</a> <a href="#">EIO0000000495 (CHS)</a>

Titel der Dokumentation	Referenz-Nummer
SoMachine Controller Assistant – Benutzerhandbuch	<a href="#">EIO0000001671 (ENG)</a> <a href="#">EIO0000001672 (FRE)</a> <a href="#">EIO0000001673 (GER)</a> <a href="#">EIO0000001675 (SPA)</a> <a href="#">EIO0000001674 (ITA)</a> <a href="#">EIO0000001676 (CHS)</a>

Diese technischen Veröffentlichungen sowie andere technische Informationen stehen auf unserer Website <https://www.schneider-electric.com/en/download> zum Download bereit.

## Produktbezogene Informationen

 <b>WARNUNG</b>
<p><b>STEUERUNGS AUSFALL</b></p> <ul style="list-style-type: none"> <li>● Bei der Konzeption von Steuerungsstrategien müssen mögliche Störungen auf den Steuerpfaden berücksichtigt werden, und bei bestimmten kritischen Steuerungsfunktionen ist dafür zu sorgen, dass während und nach einem Pfadfehler ein sicherer Zustand erreicht wird. Beispiele kritischer Steuerungsfunktionen sind die Notabschaltung (Not-Aus) und der Nachlauf-Stopp, Stromausfall und Neustart.</li> <li>● Für kritische Steuerungsfunktionen müssen separate oder redundante Steuerpfade bereitgestellt werden.</li> <li>● Systemsteuerungspfade können Kommunikationsverbindungen umfassen. Dabei müssen die Auswirkungen unerwarteter Sendeverzögerungen und Verbindungsstörungen berücksichtigt werden.</li> <li>● Sämtliche Unfallverhütungsvorschriften und lokale Sicherheitsrichtlinien sind zu beachten.<sup>1</sup></li> <li>● Jede Implementierung des Geräts muss individuell und sorgfältig auf einen einwandfreien Betrieb geprüft werden, bevor das Gerät an Ort und Stelle in Betrieb gesetzt wird.</li> </ul> <p><b>Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.</b></p>

<sup>1</sup> Weitere Informationen finden Sie in den aktuellen Versionen von NEMA ICS 1.1 „Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control“ sowie von NEMA ICS 7.1, „Safety Standards for Construction and Guide for Selection, Installation, and Operation of Adjustable-Speed Drive Systems“ oder den entsprechenden, vor Ort geltenden Vorschriften.

## **WARNUNG**

### **UNBEABSICHTIGTER GERÄTEBETRIEB**

- Verwenden Sie mit diesem Gerät nur von Schneider Electric genehmigte Software.
- Aktualisieren Sie Ihr Anwendungsprogramm jedes Mal, wenn Sie die physische Hardwarekonfiguration ändern.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

### Terminologie gemäß den geltenden Standards

Die technischen Begriffe, Terminologien, Symbole und zugehörigen Beschreibungen, die in diesem Handbuch oder auf dem Produkt selbst verwendet werden, werden im Allgemeinen von den Begriffen oder Definitionen internationaler Standards abgeleitet.

Im Bereich der funktionalen Sicherheitssysteme, Antriebe und allgemeinen Automatisierungssysteme betrifft das unter anderem Begriffe wie *Sicherheit*, *Sicherheitsfunktion*, *Sicherer Zustand*, *Fehler*, *Fehlerreset/Zurücksetzen bei Fehler*, *Ausfall*, *Störung*, *Warnung/Warmmeldung*, *Fehlermeldung*, *gefährlich/gefahrbringend* usw.

Nachstehend einige der geltenden Standards:

<b>Norm</b>	<b>Beschreibung</b>
EN 61131-2:2007	Speicherprogrammierbare Steuerungen, Teil 2: Betriebsmittelanforderungen und Prüfungen.
ISO 13849-1:2008	Sicherheit von Maschinen: Sicherheitsbezogene Teile von Steuerungen. Allgemeine Gestaltungsleitsätze
EN 61496-1:2013	Sicherheit von Maschinen: Berührungslos wirkende Schutzeinrichtungen. Teil 1: Allgemeine Anforderungen und Prüfungen
ISO 12100:2010	Sicherheit von Maschinen – Allgemeine Gestaltungsleitsätze – Risikobeurteilung und Risikominderung
EN 60204-1:2006	Sicherheit von Maschinen – Elektrische Ausrüstungen von Maschinen – Teil 1: Allgemeine Anforderungen
EN 1088:2008 ISO 14119:2013	Sicherheit von Maschinen – Verriegelungseinrichtungen in Verbindung mit trennenden Schutzeinrichtungen – Leitsätze für Gestaltung und Auswahl
ISO 13850:2006	Sicherheit von Maschinen – Not-Halt – Gestaltungsleitsätze
EN/IEC 62061:2005	Sicherheit von Maschinen – Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbar elektronischer Steuerungssysteme
IEC 61508-1:2010	Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme: Allgemeine Anforderungen

Norm	Beschreibung
IEC 61508-2:2010	Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme: Anforderungen an sicherheitsbezogene elektrische/elektronische/programmierbare elektronische Systeme
IEC 61508-3:2010	Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme: Anforderungen an Software
IEC 61784-3:2008	Industrielle Kommunikationsnetze – Profile – Teil 3: Funktional sichere Übertragung bei Feldbussen
2006/42/EC	Maschinenrichtlinie
2014/30/EU	EMV-Richtlinie (Elektromagnetische Verträglichkeit)
2014/35/EU	Niederspannungsrichtlinie

Darüber hinaus wurden einige der in diesem Dokument verwendeten Begriffe unter Umständen auch anderen Normen entnommen, u. a.:

Norm	Beschreibung
Normenreihe IEC 60034	Rotierende elektrische Geräte
Normenreihe IEC 61800	Elektrische Leistungsantriebssysteme mit einstellbarer Drehzahl
Normenreihe IEC 61158	Industrielle Kommunikationsnetze – Feldbus für industrielle Steuerungssysteme

Bei einer Verwendung des Begriffs *Betriebsumgebung/Betriebsbereich* in Verbindung mit der Beschreibung bestimmter Gefahren und Risiken entspricht der Begriff der Definition von *Gefahrenbereich* oder *Gefahrenzone* in der *Maschinenrichtlinie (2006/42/EC)* der Norm *ISO 12100:2010*.

**HINWEIS:** Die vorherig erwähnten Standards können auf die spezifischen Produkte in der vorliegenden Dokumentation zutreffen oder nicht. Für weitere Informationen hinsichtlich individueller Standards, die auf hier beschriebene Produkte zutreffen, siehe die Eigenschaftstabellen der hier erwähnten Produkte.



---

# Kapitel 1

## Informationen zum Modicon LMC078 Motion Controller

---

### Einführung

Dieses Kapitel enthält Informationen über den Modicon LMC078 Motion Controller sowie Geräte, die mithilfe von SoMachine konfiguriert und programmiert werden können.

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Informationen zum Modicon LMC078 Motion Controller	16
Architektur für verteilte E/A	18

## Informationen zum Modicon LMC078 Motion Controller

### Überblick

Der Modicon LMC078 Motion Controller (LMC078CECS20T) von Schneider Electric ist eine Steuerung mit verschiedenen leistungsstarken Funktionen. Er ermöglicht die Steuerung einer breiten Palette von Anwendungen.

Der Modicon LMC078 Motion Controller implementiert und zentralisiert die Steuerungs- und Bewegungsfunktionen. Ein Modicon LMC078 Motion Controller synchronisiert, koordiniert und erstellt die Bewegungsfunktionen einer Maschine für maximal 24 Achsen (synchronisiert in nur 4 ms).

Dank der SoMachine-Softwareplattform mit integrierten Automatisierungsfunktionen und einer ergonomischen Schnittstelle für die Achsenkonfiguration steht mit dieser Steuerung eine optimierte Lösung zur Achsenpositionierung bereit. In Verbindung mit Lexium 32S-Servoantrieben können Sie Ihre Anwendungen damit auf einfache Weise entwickeln und in Betrieb nehmen.

Weitere Informationen zu Lexium 32S-Servoantrieben finden Sie im LXM32S Produkthanduch.

Die Softwarekonfiguration wird im SoMachine Programmierhandbuch sowie im LMC078 Motion Controller Programmierhandbuch (*siehe Seite 9*) beschrieben.

### Hauptmerkmale

Die SoMachine-Software unterstützt zur Verwendung mit diesen Steuerungen die folgenden Programmiersprachen gemäß IEC61131-3:

- IL (Instruction List): Anweisungsliste (auch AWL)
- LD (Ladder): Kontaktplan (auch KOP)
- ST: Strukturierter Text
- FBD: Funktionsbausteindiagramm
- SFC (Sequential Function Chart): Ablaufsteuerung

Die Software SoMachine kann ebenfalls zur Programmierung der Steuerung in der Programmiersprache CFC (Continuous Function Chart / Freigrafischer Funktionsplaneditor) verwendet werden.

Der LMC078 Motion Controller unterstützt folgende Feldbusse:

- Mit integrierten Kommunikationsschnittstellen:
  - CANopen-Master/Slave
  - Sercos III
  - Ethernet TCP/IP
  - Serielle Leitung
- Mit optionalen Kommunikationsmodulen:
  - EtherNet/IP-Adapter/Scanner
  - PROFIBUS DP-Slave

Der LMC078 Motion Controller unterstützt folgende E/A-Typen:

- Master-Gebereingang
- Integrierte E/A
  - Digitale E/A
  - Erweiterte Digitaleingänge (Messtaster- und Interrupt-Eingänge)
- Verteilte E/A auf CANopen- und Sercos-Feldbussen (TM5/TM7-Module)

## Leistung

Der LMC078 Motion Controller weist folgende Leistung auf:

- Bis zu 8 Achsen mit einer minimalen Synchronisierungszeit von 1 ms
- Bis zu 16 Achsen mit einer minimalen Synchronisierungszeit von 2 ms
- Bis zu 24 Achsen mit einer minimalen Synchronisierungszeit von 4 ms (verfügbar ab der Produkt-Hardware-Version RS02).
- Minimale Task-Zykluszeit (nicht für Bewegung): 250 µs

Zum Anzeigen der Hardwareversion:

1. Zeigen Sie die Konfigurationsparameter (*siehe Seite 78*) der Steuerung an.
2. Überprüfen Sie, dass die ersten 2 Zeichen des `HW_Code`-Parameters „0“ bzw. „2“ sind.

oder:

1. Ziehen Sie die LC-Anzeige der Steuerung zurate.
2. Verwenden Sie die Menü-Schaltflächen, um das `HwCode`-Menüelement anzuzeigen.
3. Überprüfen Sie, dass die ersten 2 Zeichen des `HwCode`-Parameters „0“ bzw. „2“ sind.

Beispiel eines `HW_Code`- oder `HwCode`-Parameters für Hardwareversion RS02:

0224013000000000

## Architektur für verteilte E/A

### Einführung

Der LMC078 Motion Controller bietet die Möglichkeit zur Erstellung von Inseln mit verteilten E/A über:

- Sercos-Feldbus mit TM5-Feldbusschnittstelle (TM5NS31)
- CANopen-Feldbus mit TM5-Feldbusschnittstelle (TM5NC31) oder TM7-Feldbusschnittstelle (TM7NCOM•••)

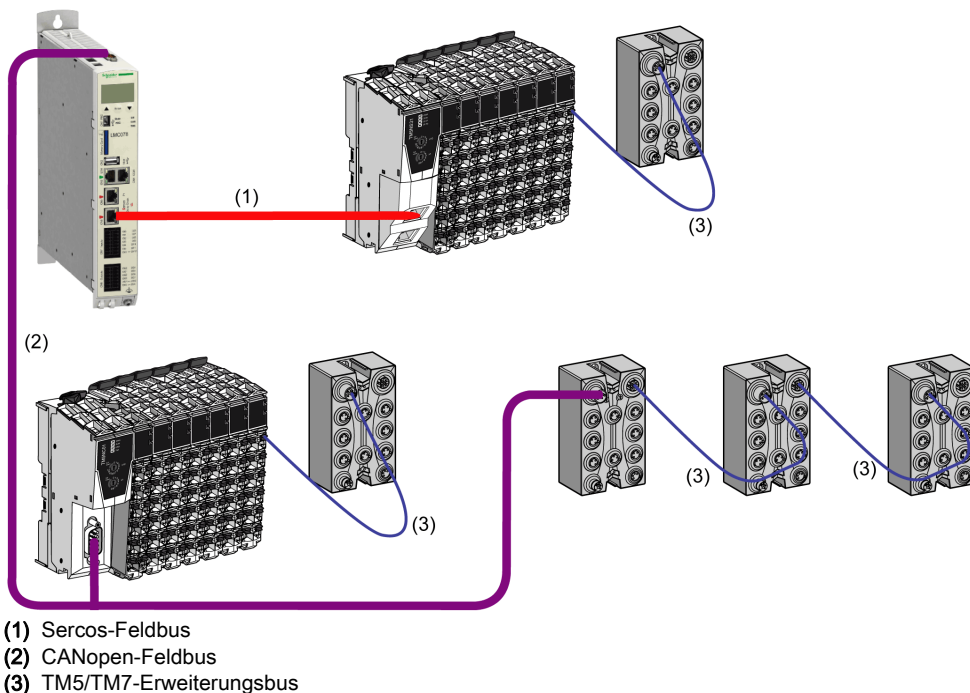
### Verteilte LMC078 Motion Controller-Architektur

Eine optimierte dezentrale Konfiguration und Flexibilität werden durch den Verband folgender Komponenten erreicht:

- LMC078 Motion Controller
- TM5- und/oder TM7-Feldbusschnittstelle
- TM5- und/oder TM7-Erweiterungsmodule

Die Architektur der LMC078 Motion Controller-Konfiguration ist von den Anforderungen der jeweiligen Anwendung abhängig.

Diese Abbildung veranschaulicht eine verteilte Konfiguration auf Sercos- und CANopen-Feldbussen:



Weitere Informationen zu TM5- und TM7-Erweiterungsbussen finden Sie unter TM5/TM7-Architektur für verteilte E/A (*siehe Modicon TM5 / TM7 Flexible System, System Planning and Installation Guide*).



---

# Kapitel 2

## Konfigurieren der Steuerung

---

### Konfiguration der Steuerung

#### Einführung

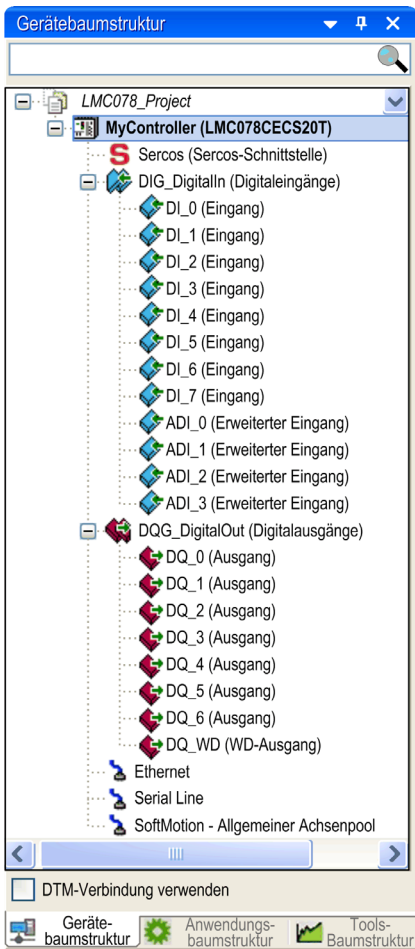
Legen Sie in der SoMachine-Software zuerst ein neues Projekt an oder öffnen Sie ein bereits vorhandenes Projekt.

Weitere Informationen zu den folgenden Themen sind dem *SoMachine-Programmierhandbuch* zu entnehmen:

- Hinzufügen einer Steuerung zu einem Projekt
- Ersetzen einer vorhandenen Steuerung
- Konvertieren einer Steuerung in ein anderes, aber kompatibles Gerät

### Gerätebaumstruktur

Die **Gerätebaumstruktur** zeigt eine strukturierte Ansicht der aktuellen Hardwarekonfiguration. Wenn Sie in Ihrem Projekt eine Steuerung hinzufügen, werden je nach den von der Steuerung bereitgestellten Funktionen in der **Gerätebaumstruktur** mehrere Knoten hinzugefügt.



Element	Wird verwendet zur Konfiguration von...
Sercos	Integrierte Sercos III-Schnittstelle
DIG_DigitalIn	Integrierte Digitaleingänge des Motion Controller
DQG_DigitalOut	Integrierte Digitalausgänge des Motion Controller

Element	Wird verwendet zur Konfiguration von...
Ethernet	Integrierte Ethernet- und SL-Kommunikationsschnittstellen (Serielle Leitung)
Serielle Leitung	
SoftMotion – Allgemeiner Achsenpool	SoftMotionLight-Geräte (Virtuelle Achsenkonfiguration)

### Anwendungsbaumstruktur

In der **Anwendungsbaumstruktur** können Sie projektspezifische Anwendungen sowie globale Anwendungen, POUs und Tasks verwalten.

### Tools-Baumstruktur

In der **Tools-Baumstruktur** können Sie folgende Aktionen durchführen:

- Konfigurieren des HMI-Teils eines Projekts
- Verwalten von Bibliotheken
- Zugreifen auf das Tool **Geräteadressierung** (*siehe Seite 216*).
- Zugreifen auf das Tool **Meldungslogger** (*siehe Seite 266*).
- Hinzufügen von CNC-Programmen



---

# Kapitel 3

## Bibliotheken

---

### Bibliotheken

#### Einführung

Bibliotheken stellen Funktionen, Funktionsbausteine, Datentypen und globale Variablen zur Verfügung, die zur Entwicklung Ihres Projekts genutzt werden können.

Der **Bibliotheksverwalter** von SoMachine zeigt Informationen zu den in Ihrem Projekt enthaltenen Bibliotheken an und ermöglicht die Installation neuer Bibliotheken. Weitere Informationen zum **Bibliotheksverwalter** SoMachine finden Sie im - Programmierhandbuch.

#### Modicon LMC078 Motion Controller

Wenn Sie einen Modicon LMC078 Motion Controller für Ihre Anwendung auswählen, lädt SoMachine die folgenden Bibliotheken:

Name der Bibliothek	Beschreibung
SystemConfiguration	Der Inhalt dieser Bibliothek wird von SoMachine nur verwendet, um die Instanzen der Antriebsfunktionsbausteine zu erstellen.
LMC078 PLCSystem <i>(siehe Modicon LMC078 Motion Controller, Systemfunktionen und - Variablen, PLCSystem Bibliothekshandbuch)</i>	Enthält Funktionen und Variablen, um Informationen abzurufen und Befehle an das Steuerungssystem zu senden.
SystemConfigurationIlf	Enthält Schnittstellen, mit denen die verschiedenen Eigenschaften der Systemobjekte (Steuerung, Antrieb und Spannungsversorgung) verwaltet werden.
IoStandard	<b>CmpIoMgr</b> -Konfigurationstypen, <b>ConfigAccess</b> , Parameter und Hilfsfunktionen: Verwaltet die E/A in der Anwendung.
Standard	Enthält alle Funktionen und Funktionsbausteine, die erforderlich sind, um Standard-POUs IEC61131-3-konform für ein IEC-Programmiersystem zu verwenden. Standard-POUs mit dem Projekt verknüpfen (standard.library).
SM3_Basic	Enthält Funktionen für die grundlegende Verwaltung von SoftMotion. Weitere Informationen finden Sie in der Online-Hilfe zu SoMachine im Kapitel <i>CoDeSys-Bibliotheken/SoftMotion-Bibliotheken</i> .
SM3_CNC	Enthält Funktionen für die CNC-Verwaltung von SoftMotion. Weitere Informationen finden Sie in der Online-Hilfe zu SoMachine im Kapitel <i>CoDeSys-Bibliotheken/SoftMotion-Bibliotheken</i> .

Name der Bibliothek	Beschreibung
LMC078 Sercos3 ( <i>siehe Seite 207</i> )	Enthält Funktionen und Variablen, um Informationen abzurufen und über die Sercos-Schnittstelle Befehle an das Steuerungssystem zu senden.
Util	Enthält Funktionen für analoge Monitore, BCD-Konvertierungen, Bit/Byte-Funktionen, Steuerungsdatentypen, Funktionsmanipulatoren, mathematische Symbole usw.
CAA Device Diagnosis	Diese Bibliothek enthält Funktionen und Schnittstellen für die Implementierung einer einfachen, aber leistungsstarken Diagnosefunktion. Die Bibliothek definiert Methoden, mit denen Sie auf die erforderlichen Informationen zu Geräten und Feldbussen zugreifen können.
CDS_MemMan	Speicherverwaltungsbibliothek.

---

# Kapitel 4

## Unterstützte Standarddatentypen

---

### Einführung

Dieses Kapitel führt die verschiedenen IEC-Datentypen an, die von der Steuerung unterstützt werden.

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Unterstützte Standarddatentypen	28
Parametertypen	29

## Unterstützte Standarddatentypen

### Unterstützte Standarddatentypen

Die Steuerung unterstützt die folgenden IEC-Datentypen:

Datentyp	Unterer Grenzwert	Oberer Grenzwert	Informationsinhalt
BOOL	FALSE	TRUE	1 Bit
BYTE	0	255	8 Bit
WORD	0	65.535	16 Bit
DWORD	0	4.294.967.295	32 Bit
LWORD	0	$2^{64}-1$	64 Bit
SINT	-128	127	8 Bit
USINT	0	255	8 Bit
INT	-32.768	32.767	16 Bit
UINT	0	65.535	16 Bit
DINT	-2.147.483.648	2.147.483.647	32 Bit
UDINT	0	4.294.967.295	32 Bit
LINT	$-2^{63}$	$2^{63}-1$	64 Bit
ULINT	0	$2^{64}-1$	64 Bit
REAL	1,175494351e-38	3,402823466e+38	32 Bit
STRING	1 Zeichen	255 Zeichen	1 Zeichen = 1 Byte
WSTRING	1 Zeichen	255 Zeichen	1 Zeichen = 1 Wort
TIME	-	-	32 Bit

Weitere Informationen zu ARRAY, LTIME, DATE, TIME, DATE\_AND\_TIME und TIME\_OF\_DAY erhalten Sie im SoMachine Programmierhandbuch.

## Parametertypen

### Parametertypen

Diese Tabelle enthält eine Beschreibung der Parametertypen der Steuerung:

Typ <sup>(1)</sup>	Online bearbeitbar	Offline bearbeitbar	Schriftfarbe <sup>(2)</sup>	Eigenschaften	Gelesener Wert	Geschriebener Wert
ER	Ja	Ja	Schwarz	Eingang mit Benutzerinitialisierung. Transfer nur nach Neustart.	Schneller Speicherzugriff.	Nicht verwendet.
ED	Ja	Ja	Schwarz	Eingang mit Benutzerinitialisierung. Transfer direkt nach der Änderung.	Schneller Speicherzugriff.	Schneller Speicherzugriff.
ED	Ja	Nein	Grau	Eingang mit automatischer Initialisierung auf den Standardwert. Transfer direkt nach der Änderung. Kann nur online durch SoMachine geändert werden.	Schneller Speicherzugriff.	Schneller Speicherzugriff.
EF	Ja	Ja	Schwarz	Eingang mit Benutzerinitialisierung. Transfer direkt nach der Änderung.	Schneller Speicherzugriff.	Funktionszugriff (interne Berechnung erforderlich).
EF	Ja	Nein	Grau	Eingang mit automatischer Initialisierung. Transfer direkt nach der Änderung. Kann nur online durch SoMachine geändert werden.	Schneller Speicherzugriff.	Funktionszugriff (interne Berechnung erforderlich).
ES	Ja	Nein	Grau	Eingang. Transfer direkt nach der Änderung. Kann nur online durch SoMachine geändert werden.	Schneller Speicherzugriff.	Kommunikation via Sercos. Verzögerung des Aufrufers (üblich sind 10 bis 100 ms).
AK	Nein	Nein	Grau	Ausgang. Konstante.	Schneller Speicherzugriff.	Nicht möglich.

Typ (1)	Online bearbeitbar	Offline bearbeitbar	Schriftfarbe (2)	Eigenschaften	Gelesener Wert	Geschriebener Wert
AD	Nein	Nein	Grau	Ausgang. Dynamischer Wert.	Schneller Speicherzugriff.	Nicht möglich.
AF	Nein	Nein	Grau	Ausgang. Dynamischer Wert.	Funktionszugriff (interne Berechnung erforderlich).	Nicht möglich.
AS	Nein	Nein	Grau	Ausgang. Dynamischer Wert.	Kommunikation via Sercos. Verzögerung des Aufrufers (üblich sind 10 bis 100 ms).	Nicht möglich.

(1) Der Parametertyp wird in der Spalte **Beschreibung** der Steuerungsfenster **Konfiguration** angezeigt (Steuerungsparameter (*siehe Seite 78*), integrierte E/A-Parameter (*siehe Seite 94*), Geberparameter (*siehe Seite 103*), Sercos-Parameter (*siehe Seite 211*)).

(2) Die Schriftfarbe ist die Farbe des Parameters, der unter **Konfiguration** angezeigt wird. Wenn der Parameter in einer schwarzen Schrift angezeigt wird, ist er offline bearbeitbar.

### Sercos-Reset-Parameter

Sercos-Reset-Parameter werden nicht direkt nach der Eingabe akzeptiert, sondern erst nach dem nächsten Sercos-Start (Phase 0 -> Phase 4).

Diese Tabelle führt die Sercos-Reset-Parameter der Steuerung auf:

Parameter	Gruppe	Akzeptanz	Parametertyp
WorkingMode	Identifikation	Phase 2 -> Phase 3	EF
IdentificationMode	Identifikation	Phase 2 -> Phase 3	EF
ConfiguredTopologyAddress	Identifikation	Phase 2 -> Phase 3	EF
ConfiguredApplicationType	Identifikation	Phase 2 -> Phase 3	EF
ConfiguredSercosAddress	Identifikation	Phase 2 -> Phase 3	EF
ConfiguredSerialNumber	Identifikation	Phase 2 -> Phase 3	EF

---

# Kapitel 5

## Speicherorganisation

---

### Einführung

Dieses Kapitel beschreibt die Speicherbelegungspläne und -größen für die verschiedenen Speicherbereiche im Modicon LMC078 Motion Controller. Diese Speicherbereiche dienen zur Speicherung von Benutzerprogrammlogik, Daten und Programmierbibliotheken.

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Organisation des Steuerungsspeichers	32
Aufbau des RAM-Speichers	33
Flash-Speicheranordnung	35
USB-Speicherstick	36

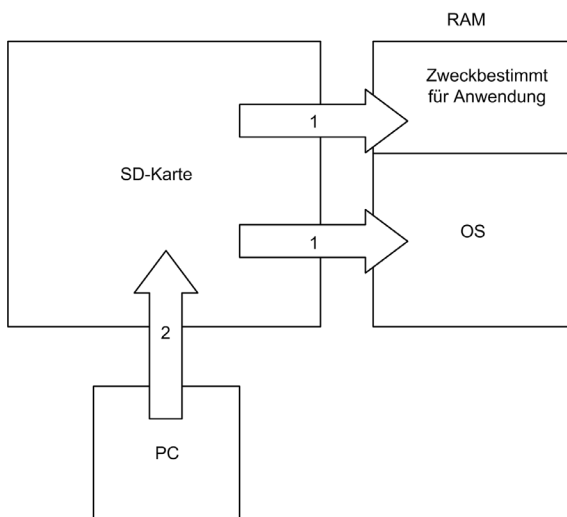
## Organisation des Steuerungsspeichers

### Einführung

Der Speicher der Steuerung setzt sich aus drei Arten von physischem Speicher zusammen:

- Der SD-Karten (*siehe Seite 35*)-Speicher enthält Dateien (Anwendung, Konfiguration und Betriebssystem).
- Der RAM (Random Access Memory) (*siehe Seite 33*) wird zur Ausführung von Anwendungen und Betriebssystem verwendet.
- Das NVRAM enthält Retain- und Persistent-Variablen.

### Dateiübertragungen im Speicher



Element	Steuerungszustand	Dateiübertragungseignisse	Verbindung	Beschreibung
1	–	Automatisch bei Start und Neustart gestartet.	Intern	Dateiübertragung von SD-Karte auf RAM. Der Inhalt des RAM-Speichers wird überschrieben.
2	Alle Zustände	Vom Benutzer gestartet	Ethernet- oder USB-Programmierport	Dateien können wie folgt übertragen werden: <ul style="list-style-type: none"> <li>• FTP-Server (<i>siehe Seite 182</i>)</li> <li>• SoMachine</li> </ul>

**HINWEIS:** Dateien auf der SD-Karte können gelesen, geschrieben oder gelöscht werden, abhängig vom Zustand der Steuerung. Die Änderung der Dateien auf der SD-Karte darf sich nicht auf die ausgeführte Anwendung auswirken. Alle an Dateien auf der SD-Karte vorgenommenen Änderungen werden beim nächsten Neustart berücksichtigt.

## Aufbau des RAM-Speichers

### Einführung

Dieser Abschnitt beschreibt die Größe des RAM (Random Access Memory) der verschiedenen Bereiche des Modicon LMC078 Motion Controller.

### Speicherzuordnung

Das RAM (512 MB) besteht aus zwei Bereichen:

- Betriebssystem-Speicher
- Reservierter Anwendungsspeicher

Das NVRAM (128 KB) besteht aus zwei Bereichen:

- Retain-Variablen
- Persistent-Variablen

Das NVRAM mit Persistent- und Retain-Variablen wird bei Stromausfällen oder bei ausgeschalteter Steuerung durch eine interne Batterie beibehalten.

Das Deklarieren von Variablen als **Persistent** erhöht die Zykluszeit der Steuerung um ungefähr 0,2 ms per 1000 Variablen.

Persistente Variablen werden in NVRAM gespeichert und bei Stromausfällen oder bei ausgeschalteter Steuerung durch eine interne Batterie beibehalten.

Konfigurieren Sie die minimale Anzahl an persistenten Variablen für Ihre Anwendung, um eine Beeinträchtigung der Steuerungsleistung zu vermeiden.

Diese Tabelle beschreibt den reservierten Anwendungsspeicher:

Bereich	Element	Größe (Byte)
Systembereich	Eingang (%I)	Minimum: 65536
	Ausgang (%Q)	Minimum: 65536
	Speicher (%M)	Minimum: 65536
Benutzerbereich	Symbole	Minimum: 1048576
	Variablen	
	Anwendung	
	Bibliotheken	

**HINWEIS:** Die definierten Größen werden während des Starts automatisch zugewiesen. Es ist auch eine dynamische Speicherzuweisung möglich.

Diese Tabelle enthält eine Beschreibung des NVRAM-Speichers:

Bereich	Größe (Byte)
Retain-Variablen	1000 bis 84501
Retain- und Persistent-Variablen	1000 bis 84501

**HINWEIS:** Um die Speichernutzung der einzelnen Bereiche zu überprüfen, klicken Sie mit der rechten Maustaste auf den Steuerungsknoten in der **Gerätebaumstruktur**, und klicken Sie auf **Gerätespeicher-Info**.

### Systemvariablen

Weitere Informationen zu Systemvariablen finden Sie im *LMC078 PLCSystem-Bibliothekshandbuch*.

### Speicheradressierung

Diese Tabelle beschreibt die Speicheradressierung für die Adressgrößen Double word (%MD), Word (%MW), Byte (%MB) und Bit (%MX):

Doppelwörter	Worte	Byte	Bit
%MD0	%MW0	%MB0	%MX0.7...%MX0.0
		%MB1	%MX1.7...%MX1.0
	%MW1	%MB2	%MX2.7...%MX2.0
		%MB3	%MX3.7...%MX3.0
%MD1	%MW2	%MB4	%MX4.7...%MX4.0
		%MB5	%MX5.7...%MX5.0
	%MW3	%MB6	%MX6.7...%MX6.0
		%MB7	%MX7.7...%MX7.0
%MD2	%MW4	%MB8	%MX8.7...%MX8.0
		...	...
	...	...	...
		...	...

Beispiel für Überlappung von Speicherbereichen:

%MD0 enthält %MB0...%MB3, %MW0 enthält %MB0 und %MB1, %MW1 enthält %MB2 und %MB3.

## Flash-Speicheranordnung

### Einführung

Die SD-Karte enthält das von der Steuerung verwendete Dateisystem.

Sie können die SD-Karte auch als Massenspeicher für Ihre Dateien verwenden.

### Dateiorganisation

In der folgenden Tabelle wird der Dateiaufbau auf der SD-Karte beschrieben:

Verzeichnis	Datei	Inhalt
\	Application.app Application.crc	Anwendung
\ESystem\	bootc4.sys	Boot loader
	sysc3.sys	Kernel und Firmware für VxWorks
	sysc3.cfg	Lzs2-Komponentenkonfiguration
\ESystem\FBUSFW\	NETX100-BSL.bin	NetX Boot Loader
	cifXrcX.nxf	Grundlegende Firmware für NetX
	DPS_XC0.nxo	Firmware für NetX für PROFIBUS DP Slave
	DPS_XC2.nxo	
	nx100eis.nxo	Firmware für NetX für den EtherNet/IP-Adapter
	nx100eim.nxo	Firmware für NetX für den EtherNet/IP-Scanner
	nx100ecs.nxo	Firmware für NetX für EtherCAT Slave
\ESystem\FirmwareDatabase\ D3\	TM5NS31_V245.fw	Firmware für das TM5NS31 Sercos-Schnittstellenmodul
\ESystem\Languages\	english.xml	LCD-Sprache
\romfs\	Prsnlty.ini	EtherBrick Konfiguration

**HINWEIS:** Mit den Funktionen der **CAA File**-Bibliothek können Sie auf die Dateien der SD-Karte zugreifen. Weitere Informationen zu den Funktionsbausteinen dieser Bibliothek finden Sie im Kapitel *CoDeSys-Bibliotheken* in der Online-Hilfe von SoMachine.

## USB-Speicherstick

### Einführung

Der USB-Speicherstick wird als Massenspeicher für Ihre Dateien verwendet. Der Zugriff erfolgt über den FTP oder die Anwendung.

**HINWEIS:** Mit den Funktionen der **CAA File**-Bibliothek können Sie auf den USB-Speicherstick zugreifen. Weitere Informationen zu den Funktionsbausteinen dieser Bibliothek finden Sie im Kapitel *CoDeSys-Bibliotheken* in der Online-Hilfe von SoMachine.

---

# Kapitel 6

## Tasks

---

### Einführung

Über den Knoten **Taskkonfiguration** in der SoMachine-**Anwendungsbaumstruktur** können Sie eine oder mehrere Tasks zur Steuerung der Ausführung eines Anwendungsprogramms definieren.

Es sind folgende Tasktypen verfügbar:

- Zyklisch
- Ereignis
- Externes Ereignis
- Status

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Maximale Anzahl an Tasks	38
Konfigurationsfenster der Tasks	39
Tasktypen	41
Bewegungstask	44
System- und Task-Watchdogs	47
Taskprioritäten	48
Standard-Taskkonfiguration	51

## Maximale Anzahl an Tasks

### Maximale Anzahl an Tasks

Der Modicon LMC078 Motion Controller unterstützt bis zu 1.000 anwendungsprogrammspezifische Tasks.

## Konfigurationsfenster der Tasks

### Beschreibung des Fensters

Im folgenden Fenster können die Tasks konfiguriert werden. Doppelklicken Sie in der **Anwendungsbaumstruktur** auf die Task, die Sie konfigurieren möchten, um dieses Fenster zu öffnen.

Jeder Konfigurationstask hat eigene, von anderen Tasks unabhängige Parameter.

Das Fenster **Konfiguration** besteht aus 4 Bereichen:

The screenshot shows a configuration window titled 'MAST x' with a 'Konfiguration' tab. The window is divided into four main sections:

- Priorität (0..31):** A text input field containing the value '1'.
- Typ:** A dropdown menu set to 'Zyklisch' and an 'Intervall (z.B. #200 ms):' field containing '#20ms'.
- Watchdog:** A section with a checked 'Aktivieren' checkbox, a 'Zeit (z.B. #200 ms):' field containing '100', and an 'Empfindlichkeit:' field containing '1'. A unit dropdown is set to 'ms'.
- POU and Kommentar:** A table with two columns: 'POU' and 'Kommentar'. The table is currently empty.

At the bottom, there is a toolbar with icons for 'Aufruf hinzufügen', 'Aufruf entfernen', 'Aufruf ändern', 'Nach oben', 'Nach unten', and 'POU öffnen'.

In der folgenden Tabelle werden die Felder des Fensters **Konfiguration** beschrieben:

Feldname	Definition
<b>Priorität</b>	<p>Sie können die Priorität einer Task mit einer Nummer von 0 bis 31 konfigurieren (0 entspricht dabei der höchsten, 31 der niedrigsten Priorität).            Zu einem Zeitpunkt kann jeweils nur eine Task ausgeführt werden. Die Priorität bestimmt, wann die Task ausgeführt wird:</p> <ul style="list-style-type: none"> <li>● Eine Task mit höherer Priorität erhält Vorrang vor einer Task mit geringerer Priorität.</li> <li>● Tasks mit derselben Priorität werden abwechselnd ausgeführt (mit einem Zeitanteil von jeweils 2 ms)</li> </ul> <p><b>HINWEIS:</b> Vermeiden Sie das Zuweisen von Tasks mit dergleichen Priorität. Wenn noch andere Tasks vorhanden sind, die versuchen, Tasks mit der gleichen Priorität zuvorzukommen, kann dies zu unvorhersehbaren Ergebnissen führen. Wichtige Informationen zur Sicherheit finden Sie unter Taskprioritäten (<i>siehe Seite 48</i>).</p>
<b>Typ</b>	<p>Die folgenden Tasktypen sind verfügbar:</p> <ul style="list-style-type: none"> <li>● Zyklisch</li> <li>● Ereignis (<i>siehe Seite 42</i>)</li> <li>● Extern (<i>siehe Seite 42</i>)</li> <li>● Status (<i>siehe Seite 42</i>)</li> </ul>
<b>Watchdog</b>	<p>Für die Konfiguration des Watchdogs (<i>siehe Seite 47</i>) müssen Sie 2 Parameter definieren:</p> <ul style="list-style-type: none"> <li>● <b>Zeit:</b> Geben Sie das Timeout ein, nach dessen Ablauf der Watchdog ausgeführt werden soll.</li> <li>● <b>Empfindlichkeit:</b> Definiert, wie oft der Watchdog-Timer ablaufen muss, bevor die Steuerung die Programmausführung beendet und sich in einen HALT-Zustand begibt.</li> </ul>
<b>POUs</b>	<p>Die Liste der von der Task gesteuerten POUs (<i>siehe SoMachine, Programmierhandbuch</i>) (Programming Organization Unit) wird im Fenster der Taskkonfiguration definiert.</p> <ul style="list-style-type: none"> <li>● Um eine mit der Task verknüpfte POU hinzuzufügen, verwenden Sie den Befehl <b>Aufruf hinzufügen</b> und wählen Sie die POU in der <b>Eingabehilfe</b> aus.</li> <li>● Verwenden Sie den Befehl <b>Aufruf löschen</b>, um eine POU aus der Liste zu entfernen.</li> <li>● Um die aktuell in der Liste ausgewählte POU durch eine andere zu ersetzen, verwenden Sie den Befehl <b>Aufruf ändern</b>.</li> <li>● <b>POUs</b> werden in der Reihenfolge ausgeführt, in der sie in der Liste angezeigt werden. Zum Verschieben der <b>POUs</b> innerhalb der Liste wählen Sie eine <b>POU</b> aus und verwenden Sie die Befehle <b>Nach oben</b> oder <b>Nach unten</b>.</li> </ul> <p><b>HINWEIS:</b> Sie können eine beliebige Anzahl von POUs erstellen. Wenn eine Anwendung anstelle einer großen POU über mehrere kleine POUs verfügt, kann dadurch die Aktualisierungszeit der Variablen im Online-Modus verbessert werden.</p>

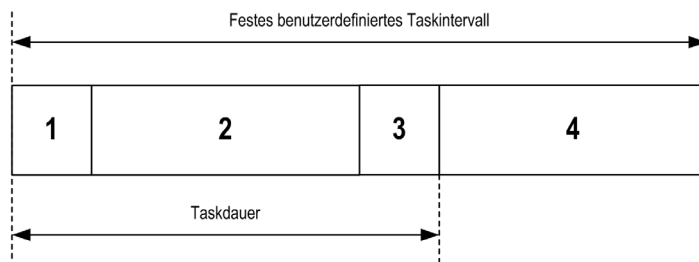
## Tasktypen

### Einführung

Im folgenden Abschnitt werden die verschiedenen, für ein Programm verfügbaren Tasktypen sowie die Eigenschaften dieser Tasktypen beschrieben.

### Zyklische Task

Einer zyklischen Task wird über die Einstellung "Intervall" im Bereich "Typ" auf der Unterregisterkarte "Konfiguration" eine feste Dauer zugewiesen. Die Ausführung einer zyklischen Task verläuft wie folgt:



1. **Eingänge lesen:** Die physischen Eingangszustände werden an die Eingangsspeichervariablen  $\%I$  geschrieben, und andere Systemvorgänge werden ausgeführt.
2. **Taskverarbeitung:** Der in der Task definierte Benutzercode (POU usw.) wird verarbeitet. Die Ausgangsspeichervariablen  $\%Q$  werden gemäß den Anweisungen im Anwendungsprogramm aktualisiert. Die Ausgangswerte verteilter E/A-Module werden während dieses Vorgangs noch nicht an die physischen Ausgänge geschrieben. Die integrierten Ausgangswerte werden sofort an die physischen Ausgänge geschrieben.
3. **Ausgänge verteilter E/A-Module schreiben:** Die Ausgangsspeichervariable  $\%Q$  wird gemäß jeglicher definierten Ausgangsforcierung angepasst, das Schreiben der physischen Ausgänge hängt jedoch vom Typ des verwendeten Ausgangs und der verwendeten Anweisungen ab. Weitere Informationen zum Definieren der Buszyklus-Task finden Sie im SoMachine - Programmierhandbuch sowie unter Modicon LMC078 Motion Controller-Einstellungen (*siehe Seite 91*).
4. **Verbleibende Intervalldauer:** Die Firmware der Steuerung führt Systemverarbeitung und andere Tasks mit geringer Priorität aus.

**HINWEIS:** Wenn der für eine zyklische Task definierte Zeitraum zu kurz ist, wird die Task wiederholt, ohne zuvor andere Tasks mit einer niedrigeren Priorität oder andere Systemverarbeitungen durchzuführen. Dies wirkt sich auf die Ausführung aller Tasks aus.


**HINWEIS:** Sie können den Intervall einer zyklischen Task über die Funktionen **GetCurrentTaskCycle** und **SetCurrentTaskCycle** je Anwendung abrufen und festlegen. (Weitere Informationen finden Sie im Toolbox\_Advance-Bibliothekshandbuch.)

Die Mindestzykluszeit für zyklische Tasks beträgt 250  $\mu$ s. Die konfigurierte Zykluszeit muss ein Vielfaches von 250  $\mu$ s sein (500  $\mu$ s, 750  $\mu$ s, 1 ms usw.).

### Ereignistask

Diese Art von Task ist ereignisgesteuert und wird durch eine Programmvariable eingeleitet. Die Task startet an der steigenden Flanke der booleschen Variable, die mit dem Trigger-Ereignis verknüpft ist, es sei denn, eine Task mit einer höheren Priorität kommt ihr zuvor. In diesem Fall wird die Ereignistask entsprechend den Vorgaben durch die Taskprioritätszuweisungen gestartet.

Wenn Sie z. B. eine Variable namens `my_Var` definiert haben und diese einem Ereignis zuweisen möchten, gehen Sie wie folgt vor:

Schritt	Aktion
1	Doppelklicken Sie in der <b>Anwendungsbaumstruktur</b> auf die <b>TASK</b> .
2	Wählen Sie auf der Registerkarte <b>Konfiguration</b> aus der Liste <b>Typ</b> das Element <b>Ereignis</b> aus.
3	Klicken Sie rechts neben dem Feld <b>Ereignis</b> auf die Schaltfläche <b>Eingabehilfe</b>  . <b>Ergebnis:</b> Das Fenster <b>Eingabehilfe</b> wird angezeigt.
4	Navigieren Sie in der Baumstruktur des Dialogs <b>Eingabehilfe</b> zur Variablen <code>my_Var</code> , um diese zuzuweisen.

**HINWEIS:** Die maximale zulässige Frequenz zum ereignisgesteuerten Triggern einer Ereignistask ist 100 Hz.

### Externe Ereignistask

Dieser Tasktyp ist ereignisgesteuert und wird durch die Erkennung eines Hardware- oder Hardware-bezogenen Funktionsereignisses initiiert. Er beginnt, wenn das Ereignis eintritt, außer wenn eine Task mit höherer Priorität vorrangig ausgeführt wird. In diesem Fall wird die Ereignistask entsprechend den Vorgaben durch die Taskprioritätszuweisungen gestartet.

So könnte z. B. eine externe Ereignistask einer steigenden Flanke an einem erweiterten Eingang (DI8 bis DI11) zugeordnet werden. Um das Ereignis **INIRQ1** mit einer externen Ereignistask zu verknüpfen, wählen Sie es in der Dropdownliste **Externes Ereignis** auf der Registerkarte **Konfiguration** aus.


Es sind bis zu 6 Ereignistypen verfügbar, die einer externen Ereignistask zugewiesen werden können:

- **INIRQx**: Steigende Flanke an einem erweiterten Eingang
- **RTP\_READ**: RTP-Prozess (Real Time Process) nach dem Echtzeit-Lesen von Daten
- **RTP\_MENC**: RTP-Prozess (Real Time Process) nach dem Master-Geber
- **RTP\_LENC**: RTP-Prozess (Real Time Process) nach dem Logikgeber
- **RTP\_AXIS**: RTP-Prozess (Real Time Process) nach der Verarbeitung der Funktionsbausteine **RefValues**
- **MDT\_WRITE\_ACCESS**: Auslösen der Bewegungstask (Schreibzugriff auf MDT (Sercos Master Data Telegram (*siehe Seite 209*)))

### Statustask

Diese Art von Task ist ereignisgesteuert und wird durch eine Programmvariable eingeleitet. Sie startet, wenn die dem auslösenden Ereignis zugeordnete boolesche Variable den Wert TRUE aufweist, vorausgesetzt, eine Task mit höherer Priorität kommt ihr nicht zuvor. In diesem Fall wird die Statustask entsprechend den Vorgaben durch die Taskprioritätszuweisungen gestartet.

Wenn Sie z. B. eine Variable namens `my_Var` definiert haben und diese einer Statustask zuweisen möchten, gehen Sie wie folgt vor:

Schritt	Aktion
1	Doppelklicken Sie in der <b>Anwendungsbaumstruktur</b> auf die <b>TASK</b> .
2	Wählen Sie <b>Status</b> aus der Liste <b>Typ</b> auf der Registerkarte <b>Konfiguration</b> aus.
3	Klicken Sie rechts neben dem Feld <b>Ereignis</b> auf die Schaltfläche <b>Eingabehilfe</b>  . <b>Ergebnis:</b> Das Fenster <b>Eingabehilfe</b> wird angezeigt.
4	Navigieren Sie in der Baumstruktur des Dialogs <b>Eingabehilfe</b> zur Variablen <code>my_Var</code> , um diese zuzuweisen.

## Bewegungstask

### Einführung

In diesem Abschnitt werden die Eigenschaften der Bewegungstask beschrieben. Die Bewegungstask wird automatisch mit dem Ereignisnamen **Extern** von **MDT\_WRITE\_ACCESS** erstellt. Dieser Mechanismus ermöglicht eine Synchronisierung der Bewegungstask mit dem Buszyklus des Sercos-Bus.

Der Parameter **Priorität (0 bis 31)** wird ignoriert. Die Task wird mit der Priorität des Echtzeitprozesses ausgeführt (höher als die IEC-Taskpriorität 0).

Die **SR\_Motion** POU wird automatisch erstellt und der Bewegungstask hinzugefügt.

#### HINWEIS:

Eine angemessen definierte Zykluszeit erfüllt die beiden nachstehenden Anforderungen:

- Die in der Bewegungstask definierte Programmverarbeitung muss über die zur vollständigen Ausführung erforderliche Zeit verfügen. Testen Sie die für die Ausführung der Bewegungstask erforderliche Zeit unter verschiedenen Betriebsbedingungen, um einen angemessenen Wert zu bestimmen.
- Die Sercos **Zykluszeit** (*siehe Seite 211*) muss ausreichen, um den physischen Austausch aller Daten zwischen der Steuerung und allen konfigurierten Geräten zu ermöglichen.

Wenn die **Zykluszeit** nicht ausreicht, kann dies eine System-Watchdog-Ausnahme oder sogar einen Synchronisierungsverlust für die gesteuerten Geräte zur Folge haben. So kann eine unzureichende **Zykluszeit** u. U. den Verlust des Sercos-Masters für alle gesteuerten Geräte verursachen. In diesem Fall wechseln alle Geräte, die den Verlust des Sercos-Masters erkennen, in den jeweils programmierten Fehlerausweichzustand. Stellen Sie grundsätzlich sicher, dass der für die **Zykluszeit** eingegebene Wert die Ausführung der Bewegungstask und einen vollständigen physischen Austausch aller PDO- und SDO-Daten ermöglicht, bevor Sie das System in Betrieb nehmen.

## WARNUNG

### UNBEABSICHTIGTER GERÄTEBETRIEB

- Berechnen Sie die zur Verarbeitung der Task und den physischen Datenaustausch mindestens erforderliche Zykluszeit.
- Definieren Sie einen Task-Watchdog (Software-Watchdog) für die Bewegungstask mit einer Watchdog-Zeit, die den Wert für die **Zykluszeit**, der für die Sercos-Schnittstelle festgelegt wurde, leicht übersteigt.
- Testen Sie Ihr Sercos-System besonders gründlich unter normalen und außergewöhnlichen Betriebsbedingungen, bevor Sie es in Betrieb nehmen.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

Diese Abbildung zeigt die Einstellungen für die Bewegungstask:

Konfiguration

Priorität (0..31): 0

Typ  
Extern Externes Ereignis: MDT\_WRITE\_ACCESS

Watchdog  
 Aktivieren  
Zeit (z.B. #200 ms): 750 µs  
Empfindlichkeit: 1

Aufruf hinzufügen Aufruf entfernen Aufruf ändern Nach oben Nach unten POU öffnen

POU	Kommentar
SR_Motion	

**HINWEIS:** Sie dürfen die Bewegungstask weder löschen noch deren Attribute Name, Typ oder Externes Ereignis ändern. Wenn Sie diese Attribute ändern, wird während des Erstellens der Applikation kein Fehler erkannt, aber es wird ein Fehler durch die Bewegungsbibliothek ausgegeben, sobald Sie versuchen, die Applikation zu verwenden.

## Programmieranforderungen für die Bewegungstask

Die Bewegungstask muss zur Verwaltung sämtlicher Aspekte der Programmierung in Zusammenhang mit dem Sercos-Bus und den angeschlossenen Bewegungsgeräten, wie z. B. Antriebssteuerungen, verwendet werden.

Diese umfasst folgende Elemente:

- Lokale Eingänge, die zur Erfassung von Bewegungsereignissen verwendet werden
- Gebereingänge, die zur Erfassung von Bewegungsereignissen verwendet werden
- Taskverarbeitung für alle Bewegungsfunktionen (Bewegungstask-POU usw.)
- Geberausgänge, die für eine Antwort auf Bewegungsereignisse konfiguriert wurden
- Lokale Ausgänge, die für eine Antwort auf Bewegungsereignisse konfiguriert wurden

### **WARNUNG**

#### **UNBEABSICHTIGTER GERÄTEBETRIEB**

Verwenden Sie die Bewegungstask zur Verwaltung aller bewegungsspezifischen Eingänge, Ausgänge, Task-Verarbeitungen und Sercos-Kommunikationen.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

## Merkmale von Bewegungstasks

Der Modicon LMC078 Motion Controller kann eine hohe Leistung erzielen. Die Steuerung kann Folgendes verwalten:

- 8 Achsen mit einer minimalen Synchronisierungszeit von 1 ms
- 16 Achsen mit einer minimalen Synchronisierungszeit von 2 ms
- 24 Achsen mit einer minimalen Synchronisierungszeit von 4 ms (verfügbar mit Hardwareversion  $\geq$  RS02). Weitere Informationen finden Sie unter Leistung (*siehe Seite 17*).

Im Folgenden ist die Untergruppe der Funktionen aufgeführt, bei deren Verwendung eine vergleichbare Leistung (sofern Sie eine effiziente Anwendung schreiben) erreicht werden kann:

- Virtuelle Achsen
- Relative und absolute Positionierung
- Geschwindigkeitssteuerung
- Nockenprofile
- Elektronisches Getriebe
- Linear- und Zirkularinterpolation Verwendung von G-Code

## System- und Task-Watchdogs

### Einführung

Für den Modicon LMC078 Motion Controller werden zwei verschiedene Watchdog-Funktionen implementiert:

- **System-Watchdog:** Dieser Watchdog wird in der SPS-Firmware definiert und von dieser verwaltet. Er kann vom Benutzer nicht konfiguriert werden.
- **Task-Watchdogs:** Diese Watchdogs sind optionale Watchdogs, die Sie für einzelne Tasks definieren können. Diese werden von Ihrem Anwendungsprogramm verwaltet und in SoMachine konfiguriert.

### System-Watchdog

Der System-Watchdog wird von der SPS-Firmware verwaltet und deshalb in der Online-Hilfe von SoMachine auch als Hardware-Watchdog bezeichnet. Wenn der System-Watchdog einen Schwellenwert überschreitet, tritt ein Fehler auf, der dann auf der Steuerung angezeigt wird.

Wenn der RTP-Prozess (Real Time Process) nicht während eines Intervalls von 100 ms ausgelöst wird, wird ein System-Watchdog erkannt. Die Steuerung wechselt in den Zustand HALT und muss neu gestartet werden, damit sie in den Zustand RUNNING zurückkehrt.

**HINWEIS:** Der System-Watchdog kann vom Benutzer nicht konfiguriert werden.

Der Digitalausgang 7 (DQ\_WD) kann als vom System-Watchdog gesteuerter Watchdog-Ausgang konfiguriert werden (Konfiguration des Watchdog-Ausgangs).

### Task-Watchdogs

SoMachine ermöglicht das Konfigurieren eines optionalen Task-Watchdogs für jede in der Anwendung definierte Task. (Task-Watchdogs werden in der Online-Hilfe zu SoMachine manchmal auch als Software-Watchdogs bezeichnet.) Wenn einer der definierten Task-Watchdogs seine Schwellenwertbedingung erreicht, tritt ein Anwendungsfehler auf, und die Steuerung begibt sich in den Zustand HALT (Diagnosemeldungen (*siehe Seite 272*)).

Wenn Sie einen Task-Watchdog definieren, sind folgende Optionen verfügbar:

- **Zeit:** Diese Option definiert die maximal zulässige Dauer für die Ausführung einer Task. Wenn ein Task länger braucht, als hier festgelegt wurde, meldet die Steuerung eine Task-Watchdog-Ausnahme.
- **Empfindlichkeit:** Dieses Feld definiert die Anzahl der Task-Watchdog-Ausnahmen, die auftreten müssen, bevor die Steuerung einen Anwendungsfehler erkennt.


Um auf die Konfiguration eines Task-Watchdogs zuzugreifen, doppelklicken Sie in der **Anwendungsbaumstruktur** auf die **Task**.

**HINWEIS:** Weitere Informationen zu Watchdogs finden Sie im SoMachine - Programmierhandbuch.

## Taskprioritäten

### Konfiguration der Taskpriorität

Sie können für jede Task eine Priorität zwischen 0 und 31 konfigurieren (0 ist die höchste und 31 die geringste Priorität). Jede Task muss einen eindeutigen Namen besitzen. Wenn Sie mehreren Tasks die gleiche Priorität zuweisen, ist die Ausführung dieser Tasks unbestimmt und unvorhersehbar, was zu unbeabsichtigten Ergebnissen führen kann.

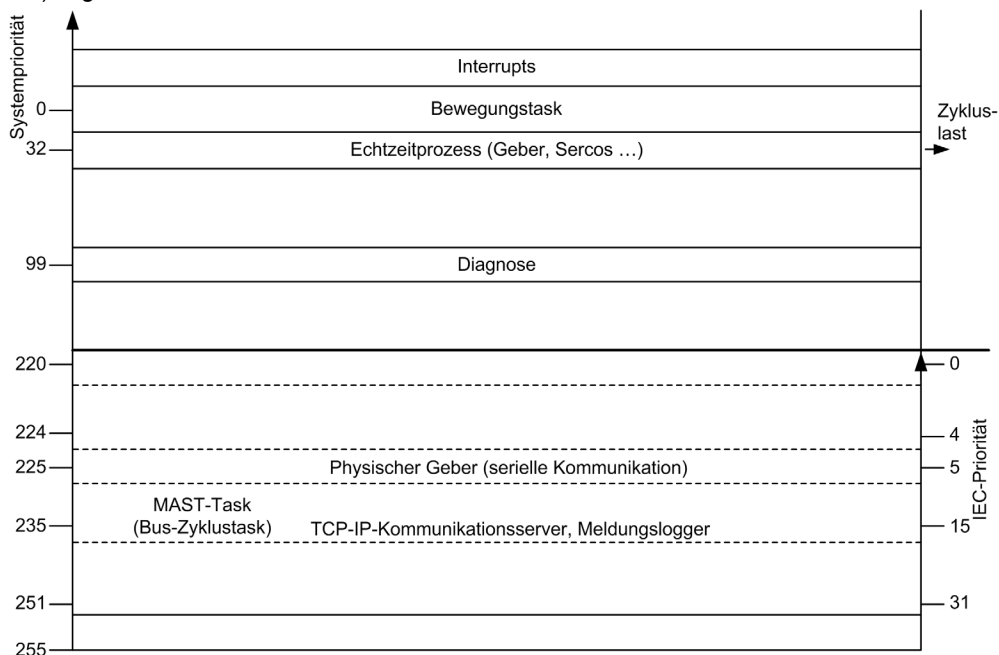
 **WARNUNG**

**UNBEABSICHTIGTER GERÄTEBETRIEB**

Eine Prioritätsstufe darf nicht zwei verschiedenen Tasks zugewiesen werden.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

Die IEC-spezifischen Prioritäten (0 bis 31) werden den systemspezifischen Prioritäten (220 bis 251) zugeordnet.



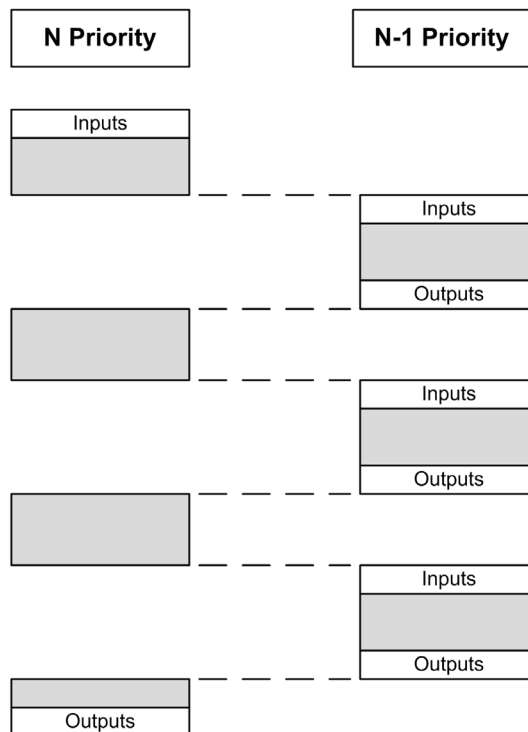
Die Bewegungstask (dem Ereignis **MDT\_WRITE\_ACCESS** zugeordnet) wird mit der Systempriorität 0 erstellt und über den RTP-Prozess (Real Time Process – Systempriorität 32) ausgelöst.

### Empfehlungen bzgl. der Task-Prioritäten

- Priorität 0 bis 24: Steuerungs-Tasks. Ordnen Sie diese Prioritäten den Tasks mit einer hohen Verfügbarkeitsanforderung zu.
- Priorität 25 bis 31: Hintergrund-Tasks. Weisen Sie diese Prioritäten Tasks zu, für die ein geringer Verfügbarkeitsbedarf besteht.

### Task-Preemption aufgrund von Taskprioritäten

Wenn ein Taskzyklus gestartet wird, kann dieser jegliche Tasks mit einer geringeren Priorität unterbrechen (Task-Preemption). Die unterbrochene Task wird wiederaufgenommen, wenn der Taskzyklus mit der höheren Priorität fertiggestellt wurde.



**HINWEIS:** Wenn ein Eingang für verschiedene Tasks verwendet wird, kann sich das Eingangsbild während des Zyklus einer Task mit einer niedrigeren Priorität ändern.

Um beim Multitasking ein ordnungsgemäßes Ausgangsverhalten gewährleisten zu können, wird eine Meldung angezeigt, wenn mehrere Ausgänge in einem Byte von verschiedenen Tasks verwendet werden.

Die integrierten Ausgänge werden sofort nach dem Schreiben aktualisiert, nicht erst am Ende des Taskzyklus.

Die integrierten Eingänge werden sofort nach der Zustandsänderung aktualisiert, nicht nur zu Beginn des Taskzyklus.

## **WARNUNG**

### **UNBEABSICHTIGTER GERÄTEBETRIEB**

Ordnen Sie die Eingänge so zu, dass die Eingangsbilder von den Tasks nicht auf unerwartete Weise geändert werden.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

## Standard-Taskkonfiguration

### Standard-Taskkonfiguration

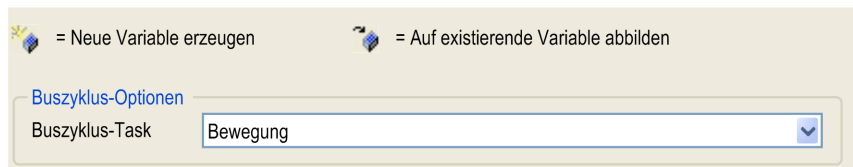
Für den Modicon LMC078 Motion Controller:

- Die MAST-Task wird automatisch als zyklische Task erstellt. Dabei wird die Task auf eine mittlere Priorität (15) und ein Intervall von 10 ms voreingestellt, und der Task-Watchdog-Dienst wird auf eine Dauer von 10 ms und eine Empfindlichkeit von 5 gesetzt. Informationen zu den Prioritätseinstellungen finden Sie unter Taskprioritäten (*siehe Seite 48*). Weitere Informationen zu Watchdogs finden Sie unter System- und Task-Watchdogs (*siehe Seite 47*).
- Eine Bewegungstask wird automatisch erstellt. Diese Task wird als externe Ereignistask deklariert und verringert die Anzahl der externen Ereignistasks, die Sie für andere Betriebsweisen konfigurieren können. Die Bewegungstask wird mit der Priorität des Echtzeitprozesses ausgeführt (der Prioritätsparameter wird ignoriert). Zwar ist diese Einstellung für zahlreiche Installationen geeignet, es liegt jedoch in Ihrer Verantwortung, die korrekte Taskpriorität für Ihr System zu überprüfen. Weitere Informationen finden Sie unter Taskprioritäten (*siehe Seite 48*).
- Eine `SR_Motion` POU wird automatisch erstellt und von der Bewegungstask aufgerufen.

**HINWEIS:** Sie dürfen die Bewegungstask weder löschen noch deren Attribute **Name**, **Typ** oder **Externes Ereignis** ändern. Wenn Sie diese Attribute ändern, erkennt SoMachine bei der Erstellung der Anwendung keinen Fehler, aber es wird ein Fehler von der Bewegungsbibliothek ausgegeben, sobald Sie versuchen, die Anwendung zu verwenden.

Die standardmäßige **Zykluszeit** ist 1 ms (Sercos-Schnittstellenkonfiguration (*siehe Seite 211*)).

Die **Buszyklus-Optionen** auf der Registerkarte **E/A-Abbild** aller Geräte, die von der Bewegungsanwendung gesteuert werden, müssen auf die Bewegungstask gesetzt werden:



**HINWEIS:** Sie können den gesamten Code in diese Task schreiben, wenn die Leistung der Steuerung, die Größe des Programms und die im Programm ausgeführten Funktionen dies zulassen.

Während der Entwicklung und Inbetriebnahme des Geräts müssen Sie die Ausführungszeit dieser Task überwachen. Die Parameter `AvailableLoad` (Steuerungsobjekt), `CycleLoad` (Sercos-Objekt) und `RTBWriteRes` (Sercos-Objekt) können zur Einschätzung der Last verwendet werden, die der Code im Echtzeitprozess für den Antrieb des Sercos-Bus benötigt. Wenn Sie die Ausführungszeit nicht überwachen, können die festgelegten Punkte für Antriebs- und Ausgangswerte der E/As verzögert werden. Die Diagnosemeldung `8507 SERCOS write cycle overflow` weist auf diese Situation hin.

### Weitere Tasks

Wenn Code in andere Tasks verschoben werden muss, sollten Sie weitere Tasks vom Typ **Zyklisch** hinzufügen und eine Priorität zwischen 16 und 31 auswählen.

---

# Kapitel 7

## Steuerungszustände und Verhalten

---

### Einführung

Dieses Kapitel enthält Informationen zu den Steuerungszuständen, Zustandsübergängen sowie den Verhalten in Reaktion auf Systemereignisse. Zunächst werden anhand eines detaillierten Diagramms die verschiedenen Steuerungszustände erläutert. Anschließend werden der Zusammenhang zwischen den Ausgangs- und den Steuerungszuständen sowie die Befehle und Ereignisse beschrieben, die Zustandsübergänge bewirken. Den Abschluss bilden Informationen zu remanenten Variablen sowie zu den Auswirkungen der Programmieroptionen der SoMachine-Tasks auf das Verhalten des Systems angegeben.

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Abschnitte:

Abschnitt	Thema	Seite
7.1	Diagramm der Steuerungszustände	54
7.2	Beschreibung der Steuerungszustände	58
7.3	Zustandsübergänge und Systemereignisse	62

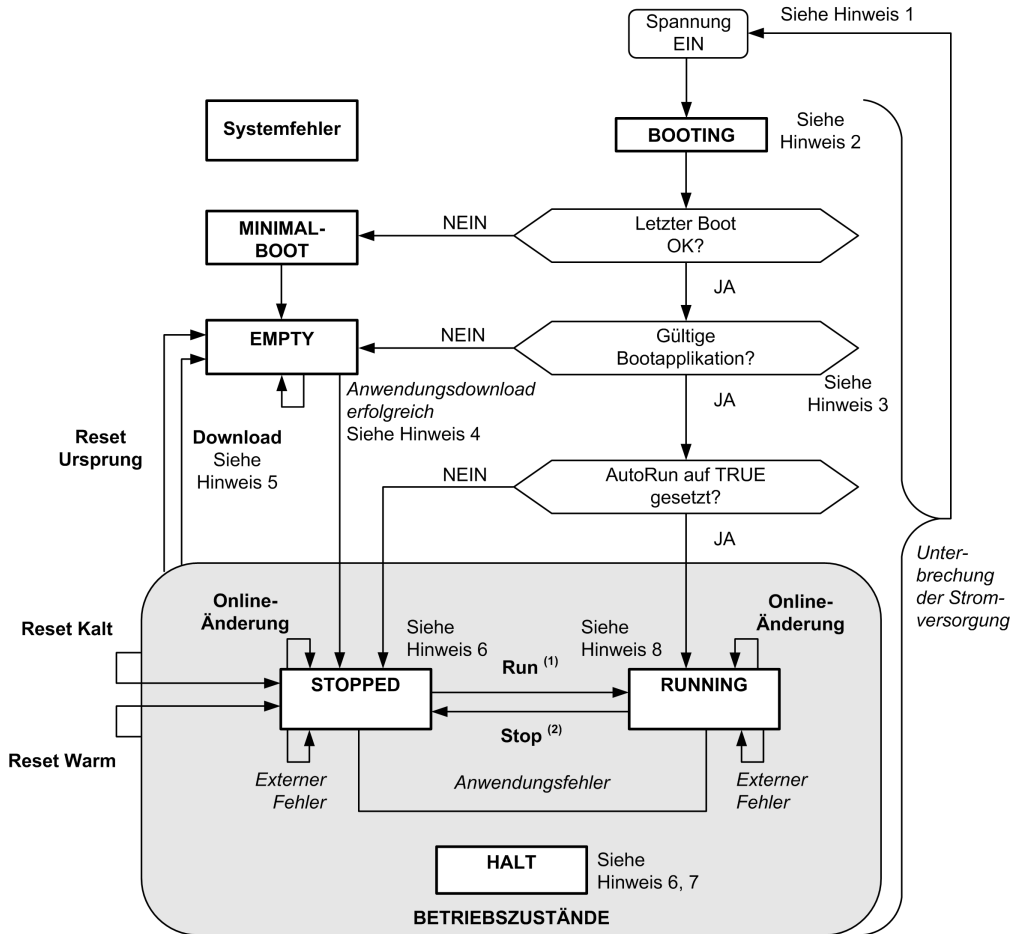
# Abschnitt 7.1

## Diagramm der Steuerungszustände

### Diagramm der Steuerungszustände

#### Diagramm der Steuerungszustände

Das folgende Diagramm illustriert die Betriebsmodi der Steuerung:



Legende:

- Die Steuerungszustände sind in **GROSSBUCHSTABEN UND FETTDRUCK** ausgewiesen.
- Benutzer- und Anwendungsbefehle sind in **Fettdruck** ausgewiesen.
- Systemereignisse sind in *Kursivschrift* ausgewiesen.
- Entscheidungen, Ergebnisse von Entscheidungen und allgemeine Informationen werden in normalem Text angegeben.

(1) Einzelheiten zum Übergang vom Zustand STOPPED in den Zustand RUNNING finden Sie unter Run-Befehl (*siehe Seite 66*).

(2) Einzelheiten zum Übergang vom Zustand RUNNING in den Zustand STOPPED finden Sie unter Stop-Befehl (*siehe Seite 66*).

### Hinweis 1

Durch Aus-/Einschalten (Spannungsunterbrechung gefolgt von Einschalten der Spannungszufuhr) werden sämtliche Einstellungen für die Ausgangsforcierung gelöscht. Weitere Einzelheiten finden Sie unter Steuerungszustände und Ausgangsverhalten (*siehe Seite 63*).

### Hinweis 2

Zwischen dem Eintritt in den Zustand BOOTING und der LED-Anzeige dieses Zustands besteht eine Verzögerung von 4 bis 5 Sekunden. Der Bootvorgang kann unter normalen Bedingungen bis zu 60 Sekunden in Anspruch nehmen. Die Ausgänge nehmen ihre Initialisierungszustände an.

### Hinweis 3

Die Anwendung wird in den RAM-Speicher geladen, sobald ihre Eigenschaft als gültige Bootapplikation überprüft wurde.

Während des Ladevorgangs für die Bootapplikation erfolgt eine Kontextprüfung, um zu bestätigen, dass die remanenten Variablen gültig sind. Wenn die Kontextprüfung nicht erfolgreich durchgeführt wurde, wird zwar die Bootapplikation geladen, die Steuerung geht jedoch in den Zustand STOPPED (*siehe Seite 68*) über.

### Hinweis 4:

Mit dem erfolgreichen Herunterladen einer Anwendung gehen folgende Ereignisse einher:

- Die Anwendung wird direkt in den RAM-Speicher geladen.
- Standardmäßig wird die Bootapplikation erstellt und auf der SD-Karte gespeichert.

## Hinweis 5

Standardmäßig begibt sich eine Steuerung nach dem Download eines Anwendungsprogramms in den Zustand STOPPED, unabhängig von der Einstellung von **AutoRun** oder dem letzten Zustand der Steuerung vor dem Download.

Diesbezüglich ist jedoch Folgendes zu beachten:

**Online Change:** Ein erfolgreich verlaufener Online Change (teilweiser Download), der eingeleitet wurde, während sich die Steuerung im Zustand RUNNING befand, versetzt die Steuerung erneut in den Zustand RUNNING. Vor Verwendung der Option **Mit Online Change einloggen** müssen Sie die Änderungen im Anwendungsprogramm in einer virtuellen bzw. außerhalb einer Produktionsumgebung testen, um sicherzustellen, dass die Steuerung mitsamt der zugeordneten Geräte die erwarteten Bedingungen im Zustand RUNNING erfüllen.

### **WARNUNG**

#### **UNBEABSICHTIGTER GERÄTEBETRIEB**

Vergewissern Sie sich immer, dass Online-Änderungen an einer Anwendung im RUNNING-Modus erwartungsgemäß funktionieren, bevor Sie sie in Steuerungen herunterladen.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

**HINWEIS:** Online-Änderungen an Ihrem Programm werden nicht automatisch an die Bootapplikation geschrieben und von der vorhandenen Bootapplikation beim nächsten Neustart überschrieben. Wenn Ihre Änderungen auch nach einem Neustart weiter bestehen sollen, müssen Sie die Bootapplikation manuell aktualisieren, indem Sie **Bootapplikation erzeugen** im **Online**-Menü auswählen (für diesen Vorgang muss sich die Steuerung im Zustand STOPPED befinden).

**Mehrfacher Download:** SoMachine enthält eine Funktion, mit der Sie eine Anwendung vollständig auf mehrere Ziele in einem Netzwerk oder auf einen Feldbus herunterladen können. Eine der Standardoptionen, die nach der Auswahl des Befehls **Mehrfach-Download...** zur Verfügung steht, lautet **Nach Download oder Online Change alle Applikationen starten**. Ist diese Option aktiviert, werden alle heruntergeladenen Ziele ungeachtet des letzten Steuerungszustand vor dem Mehrfach-Download im Zustand RUNNING neu gestartet. Deaktivieren Sie diese Option, wenn Sie nicht wünschen, dass sich alle Zielsteuerungen nach einem Neustart im Zustand RUNNING befinden. Vor dem Verwenden der Option **Mit Online Change einloggen** müssen Sie die Änderungen im Anwendungsprogramm in einer virtuellen bzw. außerhalb der Produktionsumgebung testen, um sicherzustellen, dass die Steuerung und die zugeordneten Geräte die erwarteten Bedingungen im Zustand RUNNING erfüllen.

## WARNUNG

### UNBEABSICHTIGTER GERÄTEBETRIEB

Vergewissern Sie sich immer, dass Ihr Anwendungsprogramm auf allen Zielsteuerungen und -geräten erwartungsgemäß funktioniert, bevor Sie den Befehl "**Mehrfacher Download...**" mit ausgewählter Option "**Nach Download oder Online Change alle Applikationen starten**" erteilen.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

**HINWEIS:** Im Gegensatz zu einem normalen Download wird die Option zum Erstellen einer Bootapplikation während des Mehrfach-Downloads von SoMachine nicht zur Verfügung gestellt. Sie können eine Bootapplikation jederzeit manuell erstellen, indem Sie **Bootapplikation erzeugen** im **Online-Menü** auf allen Zielsteuerungen auswählen (für diesen Vorgang muss sich die Steuerung im Zustand STOPPED befinden).

### Hinweis 6

Die SoMachine-Softwareplattform stellt zahlreiche leistungsstarke Optionen zur Verwaltung der Taskausführung und der Ausgangszustände bereit, wenn sich die Steuerung im Zustand STOPPED oder HALT befindet. Weitere Einzelheiten finden Sie unter Beschreibung der Steuerungszustände (*siehe Seite 58*).

### Hinweis 7:

Um den Zustand HALT zu beenden, müssen Sie einen der Reset-Befehle verwenden (Reset Warm, Reset Kalt, Reset Ursprung), eine Anwendung herunterladen oder die Steuerung aus- und wieder einschalten.

Bei Auftreten eines Ereignisses, bei dem keine Wiederherstellung möglich ist (Hardware-Watchdog oder interner Fehler), muss die Steuerung auf jeden Fall aus- und anschließend wieder eingeschaltet werden.

### Hinweis 8

Im Zustand RUNNING gibt es zwei Ausnahmbedingungen.

Hierbei handelt es sich um:

- RUNNING mit externem Fehler: Diese Ausnahmbedingung wird durch die Zustands-LED **STS** angezeigt, durch grünes Leuchten mit einmaligem roten Blinken. Sie können diesen Zustand beenden, indem Sie den externen Fehler löschen. Es sind keine Steuerungsbefehle erforderlich.
- RUNNING mit Haltepunkt: Diese Ausnahmbedingung wird durch die Zustands-LED **STS** angezeigt, durch dreimaliges grünes Blinken. Weitere Einzelheiten finden Sie unter Beschreibung der Steuerungszustände (*siehe Seite 58*).

## Abschnitt 7.2

### Beschreibung der Steuerungszustände

---

#### Beschreibung der Steuerungszustände

##### Einleitung

Dieser Abschnitt enthält eine detaillierte Beschreibung der Steuerungszustände.

### **WARNUNG**

#### **UNBEABSICHTIGTER BETRIEBSZUSTAND DES GERÄTS**

- Gehen Sie niemals davon aus, dass sich die Steuerung in einem bestimmten Steuerungszustand befindet, wenn Sie einen Zustandswechsel anfordern, die Steuerungsoptionen konfigurieren oder die physische Konfiguration der Steuerung und der damit verbundenen Geräte ändern.
- Ziehen Sie die konkreten Auswirkungen auf alle angeschlossenen Geräte in Betracht, bevor Sie irgendeinen dieser Vorgänge durchführen.
- Bevor Sie Änderungen an der Steuerung vornehmen, müssen Sie grundsätzlich den Steuerungszustand feststellen und dazu auf das Ausgangsforcing achten bzw. die Statusinformationen der Steuerung überprüfen (über SoMachine).<sup>(1)</sup>

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

<sup>(1)</sup> Die Steuerungszustände können mit der Funktion `FC_DiagMsgRead` der LMC078 PLCSystem-Bibliothek (siehe *Modicon LMC078 Motion Controller, Systemfunktionen und -Variablen, PLCSystem Bibliothekshandbuch*) oder in SoMachine mit dem Meldungslogger der Steuerung gelesen werden.

## Tabelle der Steuerungszustände

Die folgende Tabelle beschreibt die Zustände der Steuerung:

Steuerungszustand	Beschreibung	STS-LED
HOCHFahren	Die Steuerung führt die Boot-Firmware und ihre internen Selbsttests aus. Anschließend überprüft die Steuerung die Prüfsumme der Firmware und der Benutzeranwendungen. Sie führt weder die Anwendung aus noch kommuniziert sie.	Blinken Grün/Rot
BOOTING nach Feststellung eines <i>Systemfehlers</i>	Dieser Zustand entspricht dem normalen Zustand BOOTING, außer dass ein Flag gesetzt wird, um es so aussehen zu lassen, als wäre keine Bootapplikation vorhanden, und dass die LED-Anzeigen anders sind.	Schnelles Blinken Rot
MINIMAL BOOT	Auf der SD-Karte ist keine gültige Firmwaredatei vorhanden. Die Steuerung führt die Anwendung nicht aus. Siehe Aktualisieren der Modicon LMC078 Motion Controller-Firmware ( <i>siehe Seite 255</i> ).	Rotes Blinken
EMPTY	Es ist keine bzw. eine ungültige Anwendung vorhanden.	Einmaliges Blinken Grün
EMPTY nach Feststellung eines <i>Systemfehlers</i>	Dieser Zustand entspricht dem normalen Zustand EMPTY, außer dass ein Flag gesetzt wird, um es so aussehen zu lassen, als wäre keine Bootapplikation vorhanden (keine Anwendung geladen), und dass die LED-Anzeigen anders sind.	Schnelles Blinken Rot
RUNNING	Die Steuerung führt eine gültige Anwendung aus.	Grün
RUNNING mit Haltepunkt	Dieser Zustand ist mit dem Zustand RUNNING identisch, mit folgenden Ausnahmen: <ul style="list-style-type: none"> <li>• Der taskverarbeitende Teil des Programms wird erst fortgesetzt, wenn der Haltepunkt gelöscht wird.</li> <li>• Die LED-Anzeigen sind anders.</li> </ul> Weitere Informationen zur Verwaltung von Haltepunkten finden Sie in Online-Hilfe von SoMachine über Menübefehle.	Dreimaliges Blinken Grün
RUNNING mit Feststellung eines <i>externen Fehlers</i>	Dieser Zustand entspricht dem normalen Zustand RUNNING, mit dem Unterschied, dass die LED-Anzeigen anders sind.	Grün / Einmaliges Blinken Rot
STOPPED	Die Steuerung verfügt über eine gültige Anwendung, die gestoppt wurde. Eine Erläuterung des Verhaltens von Ausgängen und Feldbussen in diesem Zustand finden Sie unter Details zum Zustand STOPPED ( <i>siehe Seite 60</i> ).	Grünes Blinken
STOPPED mit Feststellung eines <i>externen Fehlers</i>	Dieser Zustand entspricht dem normalen Zustand STOPPED, mit dem Unterschied, dass die LED-Anzeigen anders sind.	Blinken Grün / Einmaliges Blinken Rot

Steuerungszustand	Beschreibung	STS-LED
HALT	<p>Die Steuerung stoppt die Ausführung der Anwendung, da ein Anwendungsfehler festgestellt wurde. Dieser Zustand entspricht dem Zustand STOPPED, wobei folgende Ausnahmen gelten:</p> <ul style="list-style-type: none"> <li>● Integrierte Ausgänge nehmen ihre Initialisierungswerte (<i>siehe Seite 63</i>) an.</li> <li>● Der CAN-Bus verhält sich so, als ob die Option <b>E/As aktualisieren im Stop</b> nicht ausgewählt wäre, wenn er durch die für den Anwendungsfehler verantwortliche Task verwaltet wird. Anderenfalls folgt das Verhalten des CAN-Bus der tatsächlichen Einstellung.</li> <li>● Die LED-Anzeigen sind anders.</li> </ul>	Einmaliges Blinken Rot

### Details zum Zustand STOPPED

Die folgenden Aussagen treffen für den Zustand STOPPED zu:

- Ethernet, SL (Modbus, ASCII usw.) und USB-Kommunikationsdienste bleiben funktionsfähig, die von diesen Diensten geschriebenen Befehle wirken sich weiterhin auf die Anwendung, den SPS-Zustand und die Speichervariablen aus.
- Alle Ausgänge nehmen anfangs ihren konfigurierten Standardwert an (**Werte beibehalten** oder **Alle Ausgänge auf Standardwert setzen**) oder den Zustand, der durch Ausgangsforcierung vorgegeben wird (falls verwendet). Der darauf folgende Zustand der Ausgänge hängt vom Wert der Einstellung **E/A im STOP-Zustand aktualisieren** und von Befehlen ab, die von Remotegeräten empfangen werden.

#### Task- und E/A-Verhalten bei aktivierter Option "E/A im STOP-Zustand aktualisieren"

Wenn die Einstellung **E/A im STOP-Zustand aktualisieren** ausgewählt wurde, gilt Folgendes:

- Der Vorgang "Ausgänge lesen" wird normal fortgesetzt. Die physischen Eingänge werden gelesen und dann in die Eingangsspeichervariablen %I geschrieben.
  - Der Taskverarbeitungsvorgang wird nicht ausgeführt.
  - Der Vorgang "Ausgänge schreiben" wird fortgesetzt. Die Ausgangsspeichervariablen %Q werden aktualisiert, um entweder der Konfiguration von **Werte beibehalten** oder der von **Alle Ausgänge auf Standardwert setzen** zu entsprechen, gemäß eventuell vorhandener Ausgangsforcierung angepasst und dann an die physischen Eingänge geschrieben.
- HINWEIS:** Von Ethernet-, serieller, USB- und CAN-Kommunikation empfangene Befehle können weiterhin an die Speichervariablen schreiben. Die Änderungen an den Ausgangsspeichervariablen %Q werden an die physischen Ausgänge geschrieben.

**CAN-Verhalten bei aktivierter Option "E/A im STOP-Zustand aktualisieren"**

Wenn die Einstellung **E/A im STOP-Zustand aktualisieren** ausgewählt wurde, gilt für CAN-Busse Folgendes:

- Der CAN-Bus bleibt voll betriebsfähig. Die Geräte auf dem CAN-Bus fahren mit der Suche nach einem funktionsfähigen CAN-Master fort.
- Es werden weiterhin TPDOs und RPDOs ausgetauscht.
- Optionale SDOs, sofern konfiguriert, werden weiterhin ausgetauscht.
- Die Heartbeat- und Node Guarding-Funktionen, sofern konfiguriert, sind weiterhin in Betrieb.
- Wenn das Feld **Verhalten der Ausgänge bei Stop** den Wert **Werte beibehalten** aufweist, werden die TPDOs weiterhin mit den letzten aktuellen Werten ausgegeben.
- Wenn das Feld **Verhalten der Ausgänge bei Stop** auf **Alle Ausgänge auf Standardwert setzen** eingestellt ist, werden die letzten aktuellen Werte auf die Standardwerte aktualisiert und nachfolgende TPDOs mit diesen Standardwerten ausgegeben.

**Task- und E/A-Verhalten bei deaktivierter Option "E/A im STOP-Zustand aktualisieren"**

Wenn die Einstellung **E/A im STOP-Zustand aktualisieren** nicht aktiviert ist, setzt die Steuerung die E/A entweder auf die Einstellung **Werte beibehalten** oder **Alle Ausgänge auf Standardwert setzen** (mit Anpassung für Ausgangsforcierung, sofern verwendet). Danach gilt Folgendes:

- Der Vorgang "Ausgänge lesen" wird nicht mehr fortgesetzt. Die Eingangsspeichervariablen %I werden mit ihren letzten Werten eingefroren.
- Der Taskverarbeitungsvorgang wird nicht ausgeführt.
- Der Vorgang "Ausgänge schreiben" wird nicht mehr fortgesetzt. Die Ausgangsspeichervariablen %Q können über Ethernet, serielle und USB-Verbindungen aktualisiert werden. Die physikalischen Ausgänge werden jedoch hiervon nicht beeinflusst und behalten den über die Konfigurationsoptionen vorgegebenen Zustand bei.

**CAN-Verhalten bei deaktivierter Option "E/A im STOP-Zustand aktualisieren"**

Folgendes gilt für die CAN-Busse, wenn die Einstellung **E/A im STOP-Zustand aktualisieren** nicht aktiviert ist:

- Der CAN-Master kommuniziert nicht mehr. Geräte auf dem CAN-Bus werden in ihren konfigurierten Fehlerausweichzustand versetzt.
- Es findet kein TPDO- und RPDO-Austausch mehr statt.
- Der Austausch optionaler SDOs wird angehalten, sofern konfiguriert.
- Die Heartbeat- und Node Guarding-Funktionen, sofern konfiguriert, werden eingestellt.
- Die aktuellen bzw. die Standardwerte werden an die TPDOs geschrieben und einmal gesendet, bevor der CAN-Master gestoppt wird.

## Abschnitt 7.3

### Zustandsübergänge und Systemereignisse

---

#### Übersicht

Zunächst werden in diesem Abschnitt die Ausgangszustände für die Steuerung beschrieben. Anschließend werden die Systembefehle vorgestellt, mit denen ein Übergang von einem Steuerungszustand zum einem anderen bewirkt werden kann, sowie die Systemereignisse, die ebenfalls Auswirkungen auf diese Zustände haben können. Zuletzt folgt eine Erläuterung der remanenten Variablen sowie der Umstände, unter denen verschiedene Variablen und Datentypen bei Zustandsübergängen beibehalten werden.

#### Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Steuerungszustände und Ausgangsverhalten	63
Befehlen von Zustandswechseln	66
Fehlererkennung, Fehlertypen und Fehlerhandhabung	71
Remanente Variablen	72

## Steuerungszustände und Ausgangsverhalten

### Einführung

Der Modicon LMC078 Motion Controller definiert das Ausgangsverhalten als Antwort auf Befehle und Systemereignisse, um eine größere Flexibilität zu ermöglichen. Bevor die Auswirkungen der Befehle und Ereignisse beschrieben werden, sollte dieses Verhalten genauer untersucht werden. So definieren standardmäßig verwendete Steuerungen z. B. nur zwei Optionen für das Ausgangsverhalten bei Stop: Das Zurückkehren zum Standardwert oder das Beibehalten der aktuellen Werte.

Im Folgenden werden das mögliche Ausgangsverhalten und die Steuerungszustände beschrieben, auf die diese Optionen angewendet werden.

- Verwaltung durch das **Anwendungsprogramm**
- Beibehalten der **Aktuellen Werte**
- **Alle Ausgänge auf Standardwert setzen**
- **Programm ausführen**
- **Hardware-Initialisierungswerte**
- **Software-Initialisierungswerte**
- **Ausgangsforcing**

### Vom Anwendungsprogramm verwaltet

Das Anwendungsprogramm verwaltet die Ausgänge wie gewohnt. Dies gilt für die Zustände RUNNING und RUNNING mit Externem Fehler.

### Werte beibehalten

Zur Auswahl dieser Option wählen Sie **Werte beibehalten** im Dropdown-Menü **Verhalten der Ausgänge bei Stop** auf der Unterregisterkarte **SPS-Einstellungen** des **Steuerungseditors**. Um auf den Steuerungseditor zuzugreifen, klicken Sie im Gerätebaum mit der rechten Maustaste auf die entsprechende Steuerung und wählen **Objekt bearbeiten**.

Dieses Ausgangsverhalten gilt für den Steuerungszustand STOPPED. Es gilt außerdem für den CAN-Bus im Steuerungszustand HALT. Die Ausgänge werden auf den jeweiligen Zustand gesetzt und behalten diesen bei, auch wenn das Ausgangsverhalten im Einzelnen je nach der Einstellung der Option **E/As aktualisieren im Stop** und den über die konfigurierten Feldbusse ausgelösten Aktionen stark abweicht. Weitere Einzelheiten zu diesen Variationen finden Sie unter Beschreibung der Steuerungszustände (*siehe Seite 58*).

### Alle Ausgänge auf Standardwert setzen

Zur Auswahl dieser Option wählen Sie **Alle Ausgänge auf Standardwert setzen** im Dropdown-Menü **Verhalten der Ausgänge bei Stop** auf der Unterregisterkarte **SPS-Einstellungen** des **Steuerungseditors**. Um auf den **Steuerungseditor** zuzugreifen, klicken Sie im Gerätebaum mit der rechten Maustaste auf die entsprechende Steuerung, und wählen Sie **Objekt bearbeiten**.

Das Ausgangsverhalten wird beim Wechsel der Anwendung vom Zustand RUN in den Zustand STOPPED sowie vom Zustand RUN in den Zustand HALT angewendet. Es gilt außerdem für den CAN-Bus im Steuerungszustand HALT. Die Ausgänge werden auf den jeweiligen Zustand gesetzt und behalten diesen bei, auch wenn das Ausgangsverhalten im Einzelnen je nach der Einstellung der Option **E/As aktualisieren im Stop** und den über die konfigurierten Feldbusse ausgelösten Aktionen stark abweicht. Weitere Einzelheiten zu diesen Variationen finden Sie unter Beschreibung der Steuerungszustände (*siehe Seite 58*).

### Programm ausführen

Sie können das Verhalten der Ausgänge über ein im Projekt verfügbares Programm vorgeben.

Zur Auswahl dieser Option wählen Sie **Programm ausführen** im Dropdown-Menü **Verhalten der Ausgänge bei Stop** auf der Unterregisterkarte **SPS-Einstellungen** im **Steuerungseditor** aus.

Klicken Sie auf die Schaltfläche ... und wählen Sie eine **POU** mit der **Eingabehilfe** aus.

Dieses Programm wird ausgeführt, wenn sich die Steuerung im Zustand STOPPED befindet.

### Hardware-Initialisierungswerte

Dies gilt für die Zustände BOOTING und EMPTY (im Anschluss an das Trennen und Wiederherstellen der Stromversorgung ohne eine Bootapplikation oder nach dem Auftreten eines Systemfehlers).

Im Initialisierungszustand werden die Ausgänge auf 0 gesetzt.

### Software-Initialisierungswerte

Dieser Ausgangszustand gilt für das Laden oder ein Reset der Anwendung. Sie gilt am Ende des Downloads oder am Ende eines kalten oder warmen Resets.

Die **Software-Initialisierungswerte** sind die Initialisierungswerte von Ausgangsimages (%I, %Q oder Variablen, die %I % oder %Q zugewiesen sind).

Standardmäßig sind sie auf 0 gesetzt, aber es ist möglich, die E/A in einer GVL zuzuweisen und den Ausgängen einen anderen Wert als 0 zuzuweisen.

## Ausgangsforcierung

Die Steuerung ermöglicht es, den Zustand bestimmter Ausgänge für Systemtests, Inbetriebnahme und Wartung auf einen definierten Wert zu forcieren.

Sie können den Wert eines Ausgangs nur dann forcieren, wenn die Steuerung mit SoMachine verbunden ist.

Dazu verwenden Sie den Befehl **Wert forcen** im Menü **Debug**.

Die Ausgangsforcierung setzt alle anderen Befehle an einen Ausgang außer Kraft, unabhängig von der gerade ausgeführten Taskprogrammierung.

Wenn Sie sich bei definierter Ausgangsforcierung von SoMachine abmelden, haben Sie die Möglichkeit, die Einstellungen der Ausgangsforcierung beizubehalten. Wenn Sie diese Option auswählen, steuert das Ausgangsforcing weiterhin die Zustände der ausgewählten Ausgänge, bis Sie eine Anwendung herunterladen oder einen der Reset-Befehle verwenden.

Wenn die Option **E/A im Zustand STOP aktualisieren**, sofern von Ihrer Steuerung unterstützt, aktiviert ist (Standardeinstellung), behalten die forcierten Ausgänge auch dann den forcierten Wert bei, wenn sich die speicherprogrammierbare Steuerung im Zustand STOP befindet.

## Hinweise zur Forcierung der Ausgänge

Der zu forcierende Ausgang muss in einer Task enthalten sein, die von der Steuerung ausgeführt wird. Die Forcierung von Ausgängen in nicht ausgeführten Tasks bzw. in Tasks, deren Ausführung durch Prioritäten oder Ereignisse verzögert wurde, bleibt ohne Wirkung auf den Ausgang. Sobald jedoch der verzögerte Task ausgeführt wird, wird die Forcierung angewendet.

Je nach Taskausführung kann eine Forcierung Folgen für die Anwendung haben, die für Sie nicht unbedingt direkt ersichtlich sind. Ein Beispiel: Ein Ereignistask schaltet einen Ausgang ein. Sie versuchen später, den betreffenden Ausgang auszuschalten, das Ereignis wird zu diesem Zeitpunkt jedoch nicht ausgelöst. In diesem Fall wird die Forcierung kurzerhand ignoriert. Zu einem späteren Zeitpunkt jedoch kann das Ereignis den Task auslösen, wobei dann auch die Forcierung angewendet wird.

### **WARNUNG**

#### **UNBEABSICHTIGTER GERÄTEBETRIEB**

- Sie müssen genau mit den Folgen einer Forcierung für die Ausgänge in Verbindung mit den ausgeführten Tasks vertraut sein.
- Versuchen Sie keinesfalls, Ein-/Ausgänge in Tasks zu forcieren, deren Ausführung zeitlich nicht präzise festgelegt werden kann, es sei denn, die Forcierung soll bei der nächsten Ausführung der Task angewendet werden, ungeachtet des jeweiligen Zeitpunkts.
- Wenn Sie einen Ausgang forcieren und keine direkte Wirkung auf den physischen Ausgang festzustellen ist, beenden Sie SoMachine nicht, ohne die Forcierung wieder aufzuheben.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

## Befehlen von Zustandswechseln

### Run-Befehl

Auswirkung: Befiehlt den Wechsel in den Steuerungszustand RUNNING.

Startbedingungen: Zustand BOOTING oder STOPPED.

Methoden zur Ausgabe eines Run-Befehls:

- Der `AutoRun`-Parameter wird auf der Registerkarte **Konfiguration** (*siehe Seite 79*) auf 1 gesetzt: Automatischer Start nach dem Bootvorgang.
- Der Parameter `IECProgramStateSet` wird auf der Registerkarte **Konfiguration** auf 1 gesetzt (*siehe Seite 79*).
- SoMachine-Online-Menü: Wählen Sie den Befehl **Start**.
- Option **Mit Online Change einloggen**: Wenn eine Online-Änderung (partieller Download) durchgeführt wird, während sich die Steuerung im Zustand RUNNING befindet, kehrt die Steuerung nach der erfolgreichen Durchführung der Änderung in den Zustand RUNNING zurück.
- Befehl **Mehrfach-Download**: Setzt die Steuerungen in den Zustand RUNNING, wenn die Option **Nach Download oder Online Change alle Applikationen starten** ausgewählt wurde, wobei es keine Rolle spielt, ob sich die Steuerungen ursprünglich im Zustand HALTRUNNING, STOPPED oder EMPTY befanden.
- Unter bestimmten Bedingungen wird die Steuerung automatisch im Zustand RUNNING neu gestartet.

Weitere Informationen finden Sie im Diagramm der Steuerungszustände (*siehe Seite 54*).

### Stop-Befehl

Auswirkung: Befiehlt den Wechsel in den Steuerungszustand STOPPED.

Startbedingungen: Zustand BOOTING, EMPTY oder RUNNING.

Methoden zur Ausgabe eines Stop-Befehls:

- Der Parameter `IECProgramStateSet` wird auf der Registerkarte **Konfiguration** auf 0 gesetzt (*siehe Seite 79*).
- SoMachine Online-Menü: Wählen Sie den Befehl **Stop**.
- Option **Mit Online Change einloggen**: Wenn eine Online-Änderung (partieller Download) durchgeführt wird, während sich die Steuerung im Zustand STOPPED befindet, kehrt die Steuerung nach der erfolgreichen Durchführung der Änderung in den Zustand STOPPED zurück.
- Befehl **Download**: Setzt die Steuerung implizit auf den Zustand STOPPED.
- Befehl **Mehrfach-Download**: Setzt die Steuerungen auf den Zustand STOPPED, wenn die Option **Nach Download oder Online Change alle Applikationen starten** ausgewählt wurde, wobei es keine Rolle spielt, ob sich die Steuerungen ursprünglich im Zustand HALTRUNNING, STOPPED oder EMPTY befanden.
- Unter bestimmten Bedingungen wird die Steuerung automatisch im Zustand STOPPED neu gestartet.

Weitere Informationen finden Sie im Diagramm der Steuerungszustände (*siehe Seite 54*).

## Reset (warm)

Auswirkung: Setzt alle Variablen, mit Ausnahme der remanenten Variablen, auf ihre Standardwerte zurück. Dadurch wird die Steuerung in den Zustand STOPPED gesetzt.

Startbedingungen: Zustand RUNNING, STOPPED oder HALT.

Methoden zur Ausgabe eines Befehls „Reset (warm)“:

- SoMachine Online-Menü: Wählen Sie den Befehl **Reset (warm)** aus.
- Verwendung der Funktion FC\_PrgResetAndStart aus der LMC078 PLCSystem-Bibliothek (*siehe Modicon LMC078 Motion Controller, Systemfunktionen und -Variablen, PLCSystem Bibliothekshandbuch*)

Auswirkungen des Befehls „Reset (warm)“:

1. Die Anwendung wird gestoppt.
2. Die Forcierung wird gelöscht.
3. Die Diagnoseanweisungen für Fehler werden zurückgesetzt.
4. Die Werte von Retain-Variablen werden beibehalten.
5. Die Werte von Retain-Persistent-Variablen werden beibehalten.
6. Alle nicht lokalisierten und nicht remanenten Variablen werden auf ihre Initialisierungswerte zurückgesetzt.
7. Die gesamte Feldbuskommunikation wird angehalten und neu gestartet, sobald der Reset abgeschlossen ist.
8. Alle E/A werden auf ihre Initialwerte zurückgesetzt.

Einzelheiten zu den Variablen finden Sie unter Remanente Variable (*siehe Seite 72*).

## Reset (kalt)

Auswirkung: Setzt alle Variablen mit Ausnahme remanenter Variablen des Typ Retain-Persistent auf ihre Initialisierungswerte zurück. Dadurch wird die Steuerung in den Zustand STOPPED gesetzt.

Startbedingungen: Zustand RUNNING, STOPPED oder HALT.

Methoden zur Ausgabe eines Befehls „Reset (kalt)“:

- SoMachine Online-Menü: Wählen Sie den Befehl **Reset (kalt)** aus.

Auswirkungen des Befehls „Reset (kalt)“:

1. Die Anwendung wird gestoppt.
2. Die Forcierung wird gelöscht.
3. Die Diagnoseanweisungen für Fehler werden zurückgesetzt.
4. Die Werte von Retain-Variablen werden auf ihre Initialisierungswerte zurückgesetzt.
5. Die Werte von Retain-Persistent-Variablen werden beibehalten.
6. Alle nicht lokalisierten und nicht remanenten Variablen werden auf ihre Initialisierungswerte zurückgesetzt.
7. Die gesamte Feldbuskommunikation wird angehalten und neu gestartet, sobald der Reset abgeschlossen ist.
8. Alle E/A werden auf ihre Initialwerte zurückgesetzt.

Einzelheiten zu den Variablen finden Sie unter Remanente Variablen (*siehe Seite 72*).

## Reset (Ursprung)

Auswirkung: Setzt alle Variablen, einschließlich der remanenten Variablen, auf ihre Initialisierungswerte zurück. Löscht sämtliche Benutzerdateien in der Steuerung. Dadurch wird die Steuerung in den Zustand EMPTY gesetzt.

Startbedingungen: Zustand RUNNING, STOPPED oder HALT.

Methoden zur Ausgabe eines Befehls „Reset (Ursprung)“:

- SoMachine Online-Menü: Wählen Sie den Befehl **Reset (Ursprung)** aus.

Auswirkungen des Befehls „Reset (Ursprung)“:

1. Die Anwendung wird gestoppt.
2. Die Forcierung wird gelöscht.
3. Alle Benutzerdateien (Bootapplikation, Datenprotokollierung, Post-Konfiguration) werden gelöscht.
4. Die Diagnoseanweisungen für Fehler werden zurückgesetzt.
5. Die Werte von Retain-Variablen werden zurückgesetzt.
6. Die Werte von Retain-Persistent-Variablen werden zurückgesetzt.
7. Alle nicht lokalisierten und nicht remanenten Variablen werden zurückgesetzt.
8. Jegliche Feldbuskommunikation wird gestoppt.
9. Alle E/A werden auf ihre Initialwerte zurückgesetzt.

Einzelheiten zu den Variablen finden Sie unter Remanente Variablen (*siehe Seite 72*).

## Neustart

Auswirkung: Löst den Neustart der Steuerung aus.

Startbedingungen: Jeder Zustand.

Methoden zum Erteilen eines Neustart-Befehls:

- Aus- und Einschalten
- Verwendung der Funktion `FC_SysReset` aus der LMC078 PLCSystem-Bibliothek (*siehe Modicon LMC078 Motion Controller, Systemfunktionen und -Variablen, PLCSystem Bibliothekshandbuch*)

Auswirkungen des Neustarts:

1. Das Zustand der Steuerung ist von mehreren Bedingungen abhängig:
  - a. In den folgenden Fällen weist die Steuerung den Zustand RUNNING auf:

Der Neustart wurde durch Aus- und Wiedereinschalten veranlasst:

    - der `AutoRun`-Parameter wurde auf 1 gesetzt und wenn sich die Steuerung vor dem Aus- und Einschalten nicht im Zustand HALT befunden hat und die remanenten Variablen gültig sind.

Der Neustart wurde per Skript veranlasst und:

    - als **Startmodus** ist **Start in RUN** festgelegt und wenn der Run/Stop-Eingang konfiguriert und auf RUN gesetzt ist, sich die Steuerung vor dem Aus- und Einschalten nicht im Zustand HALT befand und die remanenten Variablen gültig sind.

- b. In den folgenden Fällen weist die Steuerung den Zustand STOPPED auf:  
Der Neustart wurde durch Aus- und Wiedereinschalten veranlasst:
    - der `AutoRun`-Parameter wurde auf 0 gesetzt.
  - c. In den folgenden Fällen weist die Steuerung den Zustand EMPTY auf:
    - Es liegt keine Bootapplikation vor, oder die Bootapplikation ist ungültig, oder
    - Der Neustart wurde durch bestimmte Systemfehler veranlasst.
  - d. Die Steuerung befindet sich im Zustand INVALID\_OS, wenn keine gültige Firmware vorhanden ist.
2. Die Forcierung wird aufrechterhalten, wenn die Bootapplikation erfolgreich geladen wird. Wenn nicht, wird die Forcierung gelöscht.
  3. Die Diagnoseanweisungen für Fehler werden zurückgesetzt.
  4. Die Werte von Retain-Variablen werden wiederhergestellt, wenn der gespeicherte Kontext gültig ist.
  5. Die Werte von Retain-Persistent-Variablen werden wiederhergestellt, wenn der gespeicherte Kontext gültig ist.
  6. Alle nicht lokalisierten und nicht remanenten Variablen werden auf ihre Initialisierungswerte zurückgesetzt.
  7. Alle Feldbus-Kommunikationen werden nach dem erfolgreichen Laden und Neustarten der Bootapplikation angehalten und neu gestartet.
  8. Alle E/A werden auf ihre Initialisierungswerte und dann auf ihre benutzerdefinierten Standardwerte gesetzt, falls die Steuerung nach dem Neustart in den Zustand STOPPED übergeht.

Einzelheiten zu den Variablen finden Sie unter Remanente Variablen (*siehe Seite 72*).

**HINWEIS:** Der Test "Kontext prüfen" folgert, dass der Kontext gültig ist, wenn die Anwendung und die remanenten Variablen mit den in der Bootapplikation definierten Variablen identisch sind.

**HINWEIS:** Wenn Sie einen Online Change am Anwendungsprogramm vornehmen, während sich die Steuerung im Zustand RUNNING oder STOPPED befindet, und Sie Ihre Bootapplikation nicht manuell aktualisieren, stellt die Steuerung beim nächsten Neustart eine Diskrepanz im Kontext fest. In diesem Fall werden die remanenten Variablen wie bei einem Befehl für ein Kalt-Reset zurückgesetzt, und die Steuerung begibt sich in den Zustand STOPPED.

## Download Application-Befehl

Auswirkung: Lädt die ausführbare Anwendung in den RAM-Speicher. Erstellt optional eine Bootapplikation auf der SD-Karte.

Startbedingungen: Zustände RUNNING, STOPPED, HALT und EMPTY.

Methoden zur Ausgabe eines Befehls zum Anwendungsdownload:

- SoMachine:
  - Für den Download einer gesamten Anwendung stehen zwei Optionen zur Auswahl:
  - Download-Befehl
  - Mehrfach-Download-Befehl

Wichtige Informationen zu den Befehlen zum Anwendungsdownload finden Sie im Diagramm der Steuerungszustände (*siehe Seite 54*).

- FTP: Laden der Bootapplikationsdatei auf die SD-Karte unter Verwendung von FTP. Die aktualisierte Datei wird beim darauf folgenden Neustart angewendet.

Auswirkungen des SoMachine-Befehls Download:

1. Die vorhandene Anwendung wird gestoppt und anschließend gelöscht.
2. Die neue Anwendung wird, sofern sie gültig ist, geladen und die Steuerung wechselt in den Zustand STOPPED.
3. Die Forcierung wird gelöscht.
4. Die Diagnoseanweisungen für Fehler werden zurückgesetzt.
5. Die Werte von Retain-Variablen werden auf ihre Initialisierungswerte zurückgesetzt.
6. Die Werte jeglicher Retain-Persistent-Variablen werden beibehalten.
7. Alle nicht lokalisierten und nicht remanenten Variablen werden auf ihre Initialisierungswerte zurückgesetzt.
8. Jegliche Feldbuskommunikation wird gestoppt. Anschließend werden alle konfigurierten Feldbusse der neuen Anwendung gestartet, sobald der Download abgeschlossen ist.
9. Alle E/A werden auf ihre Initialisierungswerte zurückgesetzt und dann auf die neuen benutzerdefinierten Standardwerte gesetzt, sobald der Download abgeschlossen ist.

Einzelheiten zu den Variablen finden Sie unter Remanente Variablen (*siehe Seite 72*).

Auswirkungen des FTP-Download-Befehls:

Bis zum nächsten Neustart bleibt der Befehl ohne Wirkung. Beim nächsten Neustart hat dies dieselben Auswirkungen wie ein Neustart mit einem ungültigen Kontext. Siehe Neustart (*siehe Seite 68*).

## Fehlererkennung, Fehlertypen und Fehlerhandhabung

### Fehlerverwaltung

Die Steuerung erkennt und verwaltet drei Fehlertypen:

- Externe Fehler
- Anwendungsfehler
- Systemfehler

Diese Tabelle enthält eine Beschreibung der eventuell auftretenden Fehlertypen:

Fehlertyp	Beschreibung	Resultierender Steuerungszustand
Externe Fehler	<p>Externe Fehler werden vom System im Zustand RUNNING oder STOPPED erkannt, wirken sich jedoch nicht auf den laufenden Steuerungszustand aus. Ein externer Fehler tritt in folgenden Fällen auf:</p> <ul style="list-style-type: none"> <li>• Ein angeschlossenes Gerät generiert einen Fehler auf der Steuerung.</li> <li>• Die Steuerung stellt einen Fehler bei einem externen Gerät fest, z. B. wenn das externe Gerät kommuniziert, jedoch nicht ordnungsgemäß für die Verwendung mit der Steuerung konfiguriert ist.</li> <li>• Die Steuerung erkennt einen Fehler mit dem Zustand eines Ausgangs.</li> <li>• Die Steuerung erkennt eine Unterbrechung der Kommunikation mit einem Gerät.</li> <li>• Die Bootapplikation auf der SD-Karte ist nicht mit der Bootapplikation im RAM-Speicher identisch.</li> </ul>	<p>RUNNING mit Externem Fehler: Oder STOPPED mit Externem Fehler:</p>
Anwendungsfehler	Ein Anwendungsfehler wird im Fall einer falschen Programmierung oder bei Überschreiten des Watchdog-Schwellenwerts erkannt.	HALT
Systemfehler	<p>Ein Systemfehler tritt auf, wenn die Steuerung in eine Bedingung wechselt, die während der Laufzeit nicht gehandhabt werden kann. Die meisten derartigen Bedingungen sind auf Firmware- oder Hardwareausnahmen zurückzuführen, doch es gibt auch einige Fälle, in denen eine unsachgemäße Programmierung zu einem Systemfehler führen kann, etwa bei dem Versuch, in einen Speicherbereich zu schreiben, der während der Laufzeit reserviert ist, oder bei Auftreten eines Watchdog-Timeouts.</p> <p><b>HINWEIS:</b> Einige Systemfehler können während der Laufzeit verwaltet werden und werden daher wie Anwendungsfehler behandelt.</p>	BOOTING → EMPTY

**HINWEIS:** Ausführlichere Diagnoseinformationen finden Sie im LMC078 PLCSystem-Bibliothekshandbuch (*siehe Modicon LMC078 Motion Controller, Systemfunktionen und -Variablen, PLCSystem Bibliothekshandbuch*).

## Remanente Variablen

### Überblick

Der Wert remanenter Variablen kann im Fall von Stromausfällen, Neustarts, Resets und Anwendungsdownloads entweder reinitialisiert oder beibehalten werden. Es gibt zahlreiche Typen von remanenten Variablen, die jeweils einzeln als "Retain" oder "Persistent" oder kombiniert als "Retain-Persistent" deklariert werden.

**HINWEIS:** Bei dieser Steuerung weisen als "Persistent" deklarierte Variablen dasselbe Verhalten auf wie solche, die als "Retain-Persistent" deklariert wurden.

Diese Tabelle beschreibt das Verhalten von remanenten Variablen in den einzelnen Fällen:

Aktion	VAR	VAR RETAIN	VAR GLOBAL PERSISTENT RETAIN
Online Change am Anwendungsprogramm	X	X	X
Online-Change zum Ändern der Bootapplikation <sup>(1)</sup>	–	X	X
Stop	X	X	X
Aus- und Einschalten	–	X	X
Reset Warm	–	X <sup>(2)</sup>	X
Reset Kalt	–	–	X
Reset Ursprung	–	–	–
Download des Anwendungsprogramms <sup>(3)</sup>	–	–	X

**X** Der Wert wird beibehalten.  
 – Der Wert wird neu initialisiert.


**(1)** Die Werte von Retain-Variablen werden beibehalten, wenn ein Online-Change nur den Code-Teil der Bootapplikation ändert (beispielsweise  $a:=a+1$ ;  $\Rightarrow a:=a+2$ ;). In allen anderen Fällen werden die Retain-Variablen reinitialisiert.

**(2)** Weitere Details zu VAR RETAIN finden Sie in den Auswirkungen des Befehls „Reset (warm)“ (*siehe Seite 67*).

**(3)** Wenn die Anwendung über eine SD-Karte heruntergeladen wird, werden alle vorhandenen, von der Anwendung genutzten persistenten Variablen neu initialisiert. Beim Download der Anwendung mit SoMachine behalten die persistenten Variablen jedoch ihren jeweiligen Wert. Wenn die heruntergeladene Anwendung dieselben persistenten Variablen wie die vorhandene Anwendung enthält, behalten die vorhandenen Retain-Variablen in beiden Fällen ihren Wert.

### Hinzufügen von RETAIN-persistenten Variablen

Retain-persistente Symbole (**VAR GLOBAL PERSISTENT RETAIN**) werden im Fenster **PersistentVars** deklariert:

Schritt	Aktion
1	Wählen Sie auf der Registerkarte <b>Anwendungsbaumstruktur</b> den Knoten <b>Anwendung</b> aus.
2	Klicken Sie auf die Schaltfläche  .
3	Wählen Sie <b>Andere Objekte hinzufügen</b> → <b>Persistente Variablen</b> .
4	Klicken Sie auf <b>Hinzufügen</b> . <b>Ergebnis:</b> Das Fenster <b>PersistentVars</b> wird angezeigt.



---

# Kapitel 8

## Geräte-Editor der Steuerung

---

### Einführung

In diesem Kapitel wird die Konfiguration der Steuerung beschrieben.

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Steuerungsparameter	76
Konfigurationsparameter	78
Steuerungsauswahl	88
SPS-Einstellungen	91

## Steuerungsparameter

### Steuerungsparameter

Um den Geräteeditor zu öffnen, doppelklicken Sie in der **MyController** auf **Meine Steuerung**:



### Beschreibung der Registerkarten

Registerkarte	Beschreibung	Einschränkung
<b>Konfiguration</b>	Aufruf und Konfiguration der Steuerungsparameter.	–
<b>Steuerungsauswahl</b> <i>(siehe Seite 88)</i>	<p>Ermöglicht die Verwaltung der Verbindung zwischen PC und Steuerung:</p> <ul style="list-style-type: none"> <li>● Suchen einer Steuerung im Netzwerk</li> <li>● Anzeigen der Liste der verfügbaren Steuerungen, über die Sie eine Verbindung zur ausgewählten Steuerung herstellen und die Anwendung in der Steuerung verwalten können.</li> <li>● Physisches Erkennen der Steuerung vom Geräteeditor aus</li> <li>● Ändern der Kommunikationseinstellungen der Steuerung</li> </ul> <p>Die Steuerungsliste wird je nach den Kommunikationseinstellungen über NetManage oder über den aktiven Pfad erkannt. Um auf die <b>Kommunikationseinstellungen</b> zuzugreifen, klicken Sie in der Menüleiste auf <b>Projekt → Projekteinstellungen...</b> Weitere Informationen finden Sie im SoMachine - Programmierhandbuch (<i>Kommunikationseinstellungen</i>).</p>	Nur Online-Modus
<b>Anwendungen</b>	Zeigt die auf der Steuerung ausgeführte Anwendung an und ermöglicht das Entfernen der Anwendung aus der Steuerung.	Nur Online-Modus
<b>Dateien</b>	Dateiverwaltung zwischen PC und Steuerung.	Nur Online-Modus
<b>Protokoll</b>	Anzeigen der Protokolldatei der Steuerung.	Nur Online-Modus

Registerkarte	Beschreibung	Einschränkung
<b>SPS-Einstellungen</b> <i>(siehe Seite 91)</i>	Konfiguration von: <ul style="list-style-type: none"> <li>● Anwendung für E/A-Behandlung</li> <li>● E/A-Verhalten bei Stopp</li> <li>● Buszyklus-Optionen</li> </ul>	–
<b>Taskkonfiguration</b>	Zeigt einer Liste der E/As und deren Zuordnung zu Tasks an.	Erst nach Kompilierung
<b>Benutzer und Gruppen</b>	Die Registerkarte <b>Benutzer und Gruppen</b> steht für Geräte zur Verfügung, die die Online-Benutzerverwaltung unterstützen. Hier können Sie Benutzer und Zugriffsgruppen einrichten und ihnen Zugriffsrechte zuweisen, um den Zugriff auf SoMachine-Projekte und -Geräte im Online-Modus zu steuern. Weitere Informationen finden Sie im SoMachine - Programmierhandbuch.	–
<b>Zugriffsrechte</b>	Die Registerkarte <b>Zugriffsrechte</b> steht für Geräte zur Verfügung, die die Online-Benutzerverwaltung unterstützen. Mithilfe dieser Ansicht können aktuell definierten Benutzergruppen bestimmte Rechte erlaubt oder verboten werden, das bedeutet, die Definition der Zugriffsrechte für Benutzer auf Dateien oder Objekte (z. B. eine Anwendung) auf der Steuerung während der Laufzeit. Weitere Informationen finden Sie im SoMachine - Programmierhandbuch.	–
<b>Information</b>	Zeigt allgemeine Informationen zum Gerät an (Name, Beschreibung, Hersteller, Version, Bild).	–

## Konfigurationsparameter

### Überblick

In dieser Abbildung wird die Registerkarte **Konfiguration** dargestellt:

Meine Steuerung X					
Konfiguration   Steuerungsauswahl   SPS-Einstell.   Dateien   Protokoll   Anwendungen   Benutz. und Gruppen   Zugriffsrechte   Taskkonfiguration   Informationen					
Parameter	Typ	Wert	Standardwert	Einheit	
<b>[-] Allgemein</b>					
Name	STRING(40)	"	"		
AutoRun	BOOL-Aufzählung	Nein / 0	Nein / 0		
IP_SubNetMask	STRING(15)	'255.255.0.0'	'255.255.0.0'		
IP_Address	STRING(15)	'192.168.100.1'	'192.168.100.1'		
IP_Gateway	STRING(15)	'0.0.0.0'	'0.0.0.0'		
EthernetAdr	STRING(19)	"	"		
MsgFilter	DWORD(0 - 65535)	16#0000FFFF	16#0000FFFF		
ControllerReset	BOOL-Aufzählung	True / 1	True / 1		
IOReset	DINT-Aufzählung	Nach Downl. oder Prg.- Reset / 2	Nach Downl. oder Prg.- Reset / 2		
<b>[-] Diagnose</b>					
DiagClass	DINT				
DiagCode	DINT				
+ DiagSource					
DiagMsg	STRING(39)	"	"		
DiagExtMsg	STRING(14)	"	"		
MsgEntries	DINT				
FastTimer	UDINT				µs
Timer1	DINT				ms
Timer10	DINT				10 ms
CycleLoad	DINT				%
RTBReadRes	DINT				µs
RTBWriteRes	DINT				µs
PowerOK	BOOL-Aufzählung	False / 0	False / 0		
SetRealTimeClock	DT	0	0		
RealTimeClock	DT	0	0		
BatteryLowWarningDelay	REAL(0-50)	0.0	0.0		h
<b>[-] Versionen</b>					
FW_Version	STRING(28)	"	"		
ControllerType	STRING(254)	"	"		
ControllerType1	STRING(254)	"	"		
HW_Code	STRING(20)	"	"		
SerialNumber	STRING(20)	"	"		
<b>[-] Speicher und Festplatten</b>					
RamDiskSize	DINT(128-4096)	1024	1024		KB
RamDiskFree	DINT				KB
Diskfree	DINT				Byte
Memoryfree	DINT				Byte
<b>[-] System</b>					
Systemticks	DINT(10-20000)	4000	4000		
EnableLoadEff	BOOL-Aufzählung	Aus / 0	Aus / 0		
AvailableLoad	DINT	0	0		%
AvailableLoadPeriod	DINT(1-2000)	100	100		ms
RemoteCommunicationAccess	DINT-Aufzählung	Lesen/Schreiben/Speichern / 2	Lesen/Schreiben/Speichern / 2		
ActivateFatalCrashReaction	BOOL-Aufzählung	Ja / 1	Ja / 1		
<b>[-] IEC-Programm</b>					

**HINWEIS:** Die Parameter sind auch in der Anwendung (`ObjectName.ParameterName`, z. B. `MyController.AvailableLoad`) verfügbar.

**HINWEIS:** Auf die Parameter kann auch über Kommunikationsprotokolle zugegriffen werden.

### Parameterbeschreibung

In der folgenden Tabelle werden die Steuerungsparameter für Diagnose und Konfiguration beschrieben:

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
<b>Allgemein</b>						
Name	R/W(*)	EF	STRING	"	"	Symbolischer Name des Konfigurationsobjekts.
AutoRun	R/W	ER	BOOL Enum	no / 0 yes / 1	no / 0	<ul style="list-style-type: none"> <li>● 0 = Autorun ist nicht aktiviert.</li> <li>● 1 = Das Programm startet automatisch nach dem Start der Steuerung.</li> </ul>
IP_SubNetMask	R	AF	STRING	255.255.0.0	255.255.0.0	Zeigt die IP-Subnetzmaske an.
IP_Address	R	AF	STRING	192.168.100.1	192.168.100.1	Zeigt die IP-Adresse an.
IP_Gateway	R	AF	STRING	0.0.0.0	0.0.0.0	Zeigt die Gateway-Adresse an.
EthernetAddr	R	AF	STRING	"	"	Zeigt die gerätespezifische Ethernet-Adresse an.
MsgFilter	R/W	EF	DWORD	0 bis FFFFh	FFFFh	Konfiguriert die Meldungslogger-Filterung. Es gibt 16 Meldungsklassen: <ul style="list-style-type: none"> <li>● Bit 0: Allgemeine Systemmeldungen</li> <li>● Bit 1: Diagnosemeldungen</li> <li>● Bit 2: Programmsystemfunktionsbaustein</li> <li>● Bit 3: Feldbuspezifische Informationen</li> <li>● Bit 4 bis 11: Nicht verwendet</li> <li>● Bit 12: Erweiterte Systemmeldungen</li> <li>● Bit 13 und 14: Nicht verwendet</li> </ul>

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
ControllerReset	R/W(*)	ED	BOOL Enum	FALSE / 0 TRUE / 1	TRUE / 1	Zur Unterscheidung zwischen Programm-Reset und Steuerung-Reset. Dieser Parameter wird auf 1 gesetzt, wenn die Steuerung zurückgesetzt wird. Der Parameter kann mit einem Anwendungsprogramm auf 0 gesetzt werden. Auf diese Weise kann zwischen Programm-Reset und Steuerung-Reset unterschieden werden. <ul style="list-style-type: none"> <li>● 0 = Programm-Reset</li> <li>● 1 = Steuerung-Reset</li> </ul>
IOReset	R/W	EF	DINT Enum	0...3	2	Definiert die Reset-Modi von E/A-Bereichen: <ul style="list-style-type: none"> <li>● 0 = Kein Reset</li> <li>● 1 = Reset nach Download</li> <li>● 2 = Reset nach Download oder Programm-Reset</li> <li>● 3 = Reset nach Download, Programm-Reset oder Programmstopp</li> </ul>
<b>Diagnose</b>						
DiagClass	R	AD	DINT	-	-	Zeigt die Diagnoseklasse ( <i>siehe Seite 272</i> ) in Dezimalcode an.
DiagCode	R	AD	DINT	-	-	Zeigt den Diagnosecode ( <i>siehe Seite 272</i> ) in Dezimalcode an.
DiagSource	R	AD	ST_LogicalAddress	-	-	Gibt die Quelle der Diagnose an (Typ <code>stLogicalAddress</code> ).
DiagMsg	R	AD	STRING	-	-	Zeigt den Diagnostext an.
DiagExtMsg	R	AD	STRING	-	-	Zeigt die erweiterte Diagnosemeldung an.
MsgEntries	R	AF	DINT	-	-	Anzahl der Einträge im Meldungslogger.

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
FastTimer	R	AF	UDINT	-	-	Zeigt den Timer in $\mu\text{s}$ an. Der Timer wird vom TSC (TimeStampCounter) der CPU abgeleitet. Der TSC ist ein 64-Bit-Zähler, der mit dem CPU-Zyklus läuft. Der Zähler läuft endlos von 0 bis $2^{31}-1$ . Der Endwert entspricht etwa 71 Minuten.
Timer1	R	AF	DINT	-	-	Zeigt die Zeit seit dem Systemstart in ms an. Nach Erreichen des Maximums ( $2^{31}-1$ ms, d. h. etwa 24,8 Tage) zählt er von 0 aufwärts. Der Zähler wird nach einem Reset automatisch gestartet. Die Auflösung von Timer1 ist die CycleTime des Sercos-Antriebsbus.
Timer10	R	AF	DINT	-	-	Zeigt die Zeit seit dem Systemstart in Schritten von 10 ms an.
CycleLoad	R	AF	DINT	-	-	Zeigt die Nutzung in % des Echtzeitzyklus an (CycleLoad-Parameter ( <i>siehe Seite 86</i> )).
RTBReadRes	R	AD	DINT	-	-	Zeigt die Echtzeit-Lesereserve in $\mu\text{s}$ an.
RTBWriteRes	R	AD	DINT	-	-	Zeigt die Echtzeit-Schreibreserve in $\mu\text{s}$ an.
PowerOK	R	AF	BOOL Enum	FALSE / 0 TRUE / 1	FALSE / 0	Zeigt eine Unterspannung der Spannungsversorgung an: <ul style="list-style-type: none"> <li>● 0 = Spannung &lt; 18 V</li> <li>● 1 = Spannung &gt; 18 V</li> </ul>
SetRealTime-Clock	R/W(*)	EF	DT	-	-	Der RTC der Steuerung wird gesetzt, wenn der Parameter SetRealTimeClock geschrieben wird. Der Parameter zeigt nur die Zeit an, zu der die Uhr eingestellt wurde, oder die Zeit des letzten Starts, wenn die Hardwareuhr installiert ist und ordnungsgemäß funktioniert.

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
RealTimeClock	R	AF	DT	-	-	Zeigt Datum und Zeit der Softwareuhr an. Wird automatisch nach dem Start der Steuerung über die Echtzeit-Hardwareuhr gestellt (Batteriebetrieb bei ausgeschalteter Steuerung).
BatteryLow-WarningDelay	R/W	EF	REAL	0 bis 50	0.0	Verzögerung für Meldungen bei niedrigem Batteriestand in Stunden. Wenn die Kapazität der Batterie unter einen Mindestwert fällt, wird die Diagnosemeldung 8037 Battery low ausgelöst. Wenn die Diagnosemeldung bestätigt, aber die Batterie nicht ausgetauscht wird, wird die Diagnosemeldung nach der in diesem Parameter definierten Verzögerung ausgelöst. Die Verzögerung der Diagnosemeldung wird nicht fortgesetzt, wenn das System zurückgesetzt wird.
<b>Versionen</b>						
FW_Version	R	AK	STRING	-	-	Zeigt die Firmwareversion der Steuerung und das Erstellungsdatum an.
ControllerType	R	AK	STRING	-	-	Zeigt verschiedene Hardwareinformationen an: <ul style="list-style-type: none"> <li>● Steuerungstyp: Name des Steuerungstyps.</li> <li>● AX: Maximale Anzahl der Achsen gemäß Zykluszeit.</li> <li>● RAM: Hauptspeichererweiterung in Megabyte.</li> <li>● NVRAM: NVRAM-Erweiterung in Kilobyte.</li> <li>● Disks: SD-Kartengröße in Megabyte.</li> </ul>

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
ControllerTyp1	R	AK	STRING	-	-	Zeigt verschiedene Hardwareinformationen an: <ul style="list-style-type: none"> <li>● PFPGA-Version.</li> <li>● CPU-Version.</li> <li>● Bios-Version.</li> <li>● SFPGA-Version.</li> </ul>
HW_Code	R	AK	STRING	-	-	Zeigt den Hardwarecode der Steuerung an.
SerialNumber	R	AK	STRING	-	-	Zeigt die Seriennummer der Steuerung an.
<b>Speicher und Festplatten</b>						
RamDiskSize	R/W	ER	DINT	128 bis 4096	1024	Definiert die Größe der RamDisk in Kilobyte. Wenn die Steuerung eingeschaltet wird, wird im Hauptspeicher eine RamDisk mit der Kennung ram0: erstellt. Das System verwendet die RamDisk als temporären Datenspeicher, wenn aus dem Meldungslogger gelesen wird.
RamDiskFree	R	AF	DINT	-	-	Zeigt den freien Speicherplatz der RamDisk an.
Diskfree	R	AF	DINT	-	-	Zeigt den freien Speicherplatz der SD-Karte an.
Memoryfree	R	AF	DINT	-	-	Zeigt den freien Speicherplatz des RAM-Systemspeichers an.
<b>System</b>						
Systemticks	R/W	ER	DINT	10 bis 20000	4000	Dieser Parameter kann für die Einstellung des Systemzyklus der Steuerung verwendet werden.  <b>HINWEIS:</b> Wenn Sie diesen Parameter ändern möchten, wenden Sie sich an die Anwendungsabteilung von Schneider Electric.

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
EnableLoadEff	R/W(*)	EF	BOOL Enum	off / 0 on / 1	off / 0	Dieser Parameter startet und stoppt die effektive CPU-Lastmessung: <ul style="list-style-type: none"> <li>● 0 = Effektive CPU-Lastmessung gestoppt.</li> <li>● 1 = Effektive CPU-Lastmessung gestartet.</li> </ul>
AvailableLoad	R	AD	DINT	0 bis 100	0	Zeigt die verfügbare CPU-Zeit in % an.
AvailableLoad-Period	R/W	EF	DINT	1 bis 2000	100	Definiert den Messzeitraum für die effektive CPU-Lastmessung in ms. Für diesen Zeitraum wird die verbleibende Berechnungszeit berechnet.
RemoteCommunicationAccess	R/W	ER	DINT Enum	read only / 0 read/write / 1 read/write/save / 2	read/write/save / 2	Mit dem Tool NetManage können Sie verschiedene Kommunikationsparameter der Steuerung ändern. Wenn dies nicht gewünscht ist, können Sie Änderungen mit dem Parameter RemoteCommunicationAccess unterbinden: <ul style="list-style-type: none"> <li>● 0 = Die Werte werden im Tool NetManage angezeigt, aber die Steuerung lässt keine Änderungen an den Kommunikationsparametern zu.</li> <li>● 1 = Die Steuerung ändert ihre Kommunikationsparameter zeitweise.</li> <li>● 2 = Die Steuerung ändert ihre Kommunikationsparameter und speichert sie sofort auf der SD-Karte (und behält diese Einstellung auch bei einem Neustart). Speichern ist optional und kann im Tool NetManage ausgewählt werden.</li> </ul>

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
ActivateFatal-CrashReaction	R/W	ED	BOOL Enum	no / 0 yes / 1	yes / 1	Aktiviert/deaktiviert das spezifische Verhalten, wenn die Steuerung nicht mehr reagiert.
<b>IEC-Programm</b>						
IECRetainFree	R	AF	DINT	-	-	Zeigt den freien Speicher im Retain-Bereich an.
ProjectDate	R	AD	DT	-	-	Zeigt das Projektdatum an.
ProjectName	R	AD	STRING	-	-	Zeigt den Projektnamen an.
ProjectTitle	R	AD	STRING	-	-	Zeigt den Projekttitel an.
ProjectVersion	R	AD	STRING	-	-	Zeigt die Projektversion an.
ProjectAuthor	R	AD	STRING	-	-	Zeigt den Projektersteller an.
ProjectDescription	R	AD	STRING	-	-	Zeigt die Projektbeschreibung an.
ProgrammingSystem	R	AD	STRING	-	-	Zeigt die Programmiersystemversion (PSV) und die Version der Gerätebeschreibung (TSV) an, mit denen das Programm erstellt wurde.
OnlineChangeCounter	R	AD	DINT	-	0	Der Wert gibt an, wie viele Online-Change-Aktualisierungen der Programmsequenzen seit dem letzten Programm-Download durchgeführt wurden. Mit diesem Wert können nach einer Online-Change im Programm bestimmte Initialisierungsschritte aufgerufen werden.
IECProgramStateSet	R/W	EF	DINT Enum	stop / 0 start / 1	stop / 0	Startet bzw. beendet das ICE-Programm und zeigt den aktuellen Zustand an.
-						
ObjectType	R	AD	STRING	LMCxx8	LMCxx8	Objekttyp.

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
stLogicalAddress	R	AD	ST_LogicalAddress	-	-	Logische Adresse der Steuerungsparameter. stLogicalAddress = STRUCT (udiType, udiInstance, udiParameterId)
udiType	R	-	UDINT	-	-	
udiInstance	R	-	UDINT	-	-	
udiParameterId	R	-	UDINT	-	-	

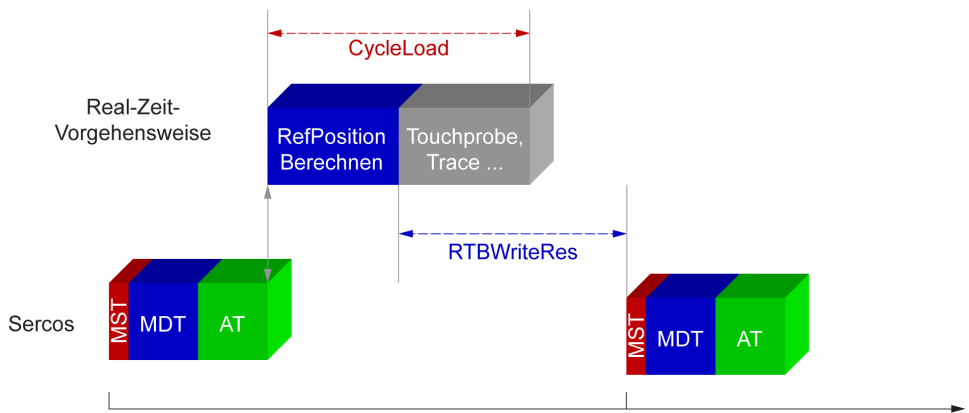
(\*) Weitere Informationen zu den Parameter-Zugriffsrechten finden Sie unter Parametertypen (siehe Seite 29).

### CycleLoadParameter

Der Parameter `CycleLoad` gibt die genutzte Kapazität (in %) der Steuerung durch den Echtzeitprozess an. Der Wert des Parameters sollte durchschnittlich 40 bis 50 % nicht überschreiten (kurze Spitzenwerte sind zulässig). Die verbleibenden 50 bis 60 % sind dann für andere Systemfunktionen verfügbar, z. B. den Feldbus-Server, das Netzwerk und Programme. Die Diagnosemeldung `8511 CPU time overflow` wird gesendet, wenn 100 % erreicht werden.

Der Parameter `CycleLoad` ist eine einfache und verständliche Variable für die Bewertung der Systemlast.

In dieser Abbildung wird die Beziehung zwischen den Parametern `CycleLoad` und `RTBWriteRes` dargestellt:



Der Echtzeitprozess (RTP) ist die wichtigste Systemtask. Er ist für die Ausführung sämtlicher Echtzeittasks zum jeweils richtigen Zeitpunkt verantwortlich. Die Echtzeitverarbeitung wird während jedes Buszyklus über den Sercos-Echtzeit-Buszyklus ausgelöst.

CycleLoad wird in zwei Schritten ausgeführt:

Schritt	Beschreibung
1	<ul style="list-style-type: none"> <li>● Vorbereitung des Zyklus.</li> <li>● Initialisierung von Messvariablen und Überwachung.</li> <li>● Akzeptanz der Echtzeitdaten, die von den aktuellsten Antriebstelegrammen (ATs) der Sercos-Slaves bereitgestellt werden.</li> <li>● Verarbeitung aller Mastergeber wie virtueller Mastergeber und physischer Mastergeber.</li> <li>● Verarbeitung aller Slave-Geber.</li> <li>● Alle Achsen: <ul style="list-style-type: none"> <li>○ Diagnosestatus</li> <li>○ Zustandsmaschine der Antriebe.</li> <li>○ Echtzeit-Jobvorbereitung</li> <li>○ POS- und CAM-Generatoren (Master- und Slave-Kurven) ...</li> <li>○ Neue Referenzwerte verfügbar</li> <li>○ Daten werden im nächsten Zyklus im Master Data Telegram (MDT) übermittelt</li> </ul> </li> <li>● Die Referenzwerte werden im nächsten Zyklus übermittelt.</li> </ul>
2	<p>Die verbleibenden Echtzeitfunktionen werden ausgeführt:</p> <ul style="list-style-type: none"> <li>● Touchprobe</li> <li>● Trace usw.</li> </ul>

**HINWEIS:** Bestimmte extern ereignisgesteuerte Tasks sowie Tasks mit hoher Priorität können die Ausführung des Echtzeitprozesses beeinträchtigen. Durch eine Überwachung der Variablen `CycleLoad`, `RTBReadRes` und `RTBWriteRes` kann das dynamische Verhalten analysiert werden.

## Steuerungsauswahl

### Einführung

Auf dieser Registerkarte können Sie die Verbindung zwischen PC und Steuerung verwalten. Die folgenden Funktionen stehen zur Verfügung:

- Suchen einer Steuerung im Netzwerk
- Anzeigen der Liste der verfügbaren Steuerungen, über die Sie eine Verbindung zur ausgewählten Steuerung herstellen und die Anwendung in der Steuerung verwalten können.
- Physisches Erkennen der Steuerung vom Geräteeditor aus
- Verarbeiten der Kommunikationseinstellungen der Steuerung

### Symbolleiste "Steuerungsauswahl"

Die Symbolleiste umfasst die folgenden Schaltflächen:



Bezeichnung	Schaltfläche	Beschreibung
1	<b>Optisch</b>	Klicken Sie auf diese Schaltfläche, damit die ausgewählte Steuerung ein optisches Signal anzeigt: Eine Kontroll-LED blinkt schnell. So können Sie die entsprechende Steuerung ausfindig machen, falls viele Steuerungen verwendet werden. Die Funktion stoppt nach einem zweiten Klick oder nach etwa 30 Sekunden.
2	<b>Optisch und akustisch</b>	Nicht unterstützt.

Bezeichnung	Schaltfläche	Beschreibung
3	<b>Aktualisieren</b>	<p>Klicken Sie auf diese Schaltfläche, um die Liste der Steuerungen zu aktualisieren. Dazu wird ein Request an die Steuerungen im Netzwerk gesendet. Jede Steuerung, die auf den Request antwortet, wird mit aktuellen Werten aufgeführt.</p> <p>Bereits vorhandene Einträge von Steuerungen werden mit jedem neuen Request aktualisiert.</p> <p>Steuerungen, die bereits in der Liste vorhanden sind, aber nicht auf einen neuen Request antworten, werden nicht gelöscht. Sie werden mit einem roten Kreuz, das dem Steuerungssymbol hinzugefügt wird, als inaktiv markiert.</p> <p>Die Schaltfläche <b>Aktualisieren</b> entspricht dem Befehl <b>Liste aktualisieren</b>, der im Kontextmenü verfügbar ist, wenn Sie mit der rechten Maustaste auf eine Steuerung in der Liste klicken.</p> <p>Um die Informationen einer ausgewählten Steuerung zu aktualisieren, stellt das Kontextmenü den Befehl <b>Diese Steuerung aktualisieren</b> bereit. Dieser Befehl ruft detailliertere Informationen von der ausgewählten Steuerung ab.</p> <p><b>HINWEIS:</b> Über den Befehl <b>Diese Steuerung aktualisieren</b> können auch die Informationen anderer Steuerungen aktualisiert werden.</p>
4	<b>Nicht aktive Steuerungen aus Liste entfernen</b>	<p>Steuerungen, die auf eine Netzwerkabfrage nicht antworten, werden in der Liste als inaktiv markiert. Dies wird durch ein rotes Kreuz gekennzeichnet, das dem Steuerungssymbol hinzugefügt wird. Klicken Sie auf diese Schaltfläche, um alle als inaktiv markierten Steuerungen gleichzeitig von der Liste zu entfernen.</p> <p><b>HINWEIS:</b> Aufgrund von Netzwerkproblemen kann eine Steuerung als inaktiv markiert werden, auch wenn das nicht der Fall ist.</p> <p>Das Kontextmenü, das Sie durch Klicken mit der rechten Maustaste in die Liste öffnen können, bietet zwei weitere Befehle zum Entfernen von Steuerungen:</p> <ul style="list-style-type: none"> <li>● Mithilfe des Befehls <b>Ausgewählte Steuerung aus Liste entfernen</b> können Sie nur die ausgewählte Steuerung aus der Liste entfernen.</li> <li>● Mithilfe des Befehls <b>Alle Steuerungen aus Liste entfernen</b> können Sie alle Steuerungen gleichzeitig aus der Liste entfernen.</li> </ul>

Bezeichnung	Schaltfläche	Beschreibung
5	<b>Neuer Favorit</b>	<p>Sie können <b>Favoriten</b> verwenden, um die Auswahl der Steuerungen an Ihre persönlichen Anforderungen anzupassen. Das hilft Ihnen dabei, viele Steuerungen im Netzwerk nachzuverfolgen.</p> <p>Ein <b>Favorit</b> bezeichnet eine Sammlung von Steuerungen, die über eine eindeutige Kennung identifiziert werden.</p> <p>Klicken Sie auf eine Favoritenschaltfläche (z. B. <b>Favorit 0</b>), um den Favoriten auszuwählen oder die Auswahl aufzuheben. Wenn Sie keinen Favoriten ausgewählt haben, werden alle erkannten Steuerungen angezeigt.</p> <p>Sie können auch über das Kontextmenü auf <b>Favoriten</b> zugreifen. Das Kontextmenü wird durch Klicken mit der rechten Maustaste auf eine Steuerung in der Liste eingeblendet.</p> <p>Bewegen Sie den Cursor über eine Favoritenschaltfläche, um die zugewiesenen Steuerungen als Tooltip anzuzeigen.</p>
6	<b>Favorit x</b>	

Weitere Informationen zur Registerkarte **Steuerungsauswahl** des Geräteeditors finden Sie im SoMachine - Programmierhandbuch.

### Kommunikationseinstellungen bearbeiten

Im Fenster **Kommunikationseinstellungen bearbeiten** können Sie die Ethernet-Kommunikationseinstellungen ändern. Klicken Sie dafür auf die Registerkarte **Steuerungsauswahl**. Die Liste der im Netzwerk verfügbaren Steuerungen wird angezeigt. Markieren Sie die erforderliche Zeile, klicken Sie mit der rechten Maustaste darauf und wählen Sie dann **Kommunikationseinstellungen bearbeiten** im Kontextmenü.

Es gibt zwei Möglichkeiten, die Ethernet-Einstellungen im Fenster **Kommunikationseinstellungen bearbeiten** zu konfigurieren:

- Ohne die Option **Einstellungen permanent speichern**:  
Konfigurieren Sie die Kommunikationsparameter und klicken Sie auf **OK**. Diese Einstellungen werden sofort wirksam und gehen bei einem Reset der Steuerung verloren.
- Mit der Option **Einstellungen permanent speichern**:  
Sie können auch die Option **Einstellungen permanent speichern** aktivieren, bevor Sie auf **OK** klicken. Wenn diese Option aktiviert ist, werden die konfigurierten Ethernet-Parameter auf der SD-Karte gespeichert. Nach dem Zurücksetzen der Steuerung werden die konfigurierten Ethernet-Parameter von der SD-Karte aktiv.

## SPS-Einstellungen

### Übersicht

Die folgende Abbildung zeigt die Registerkarte **SPS-Einstellungen**:

Element	Beschreibung
<b>Applikation für E/A-Behandlung</b>	Ist standardmäßig auf "Applikation" eingestellt, da in der Steuerung nur eine Anwendung vorhanden ist.
<b>SPS-Einstellungen</b>	<b>E/A im STOP-Zustand aktualisieren</b> Wenn diese Option aktiviert ist, werden die Werte der Ein- und Ausgangskanäle auch aktualisiert, wenn die Steuerung gestoppt wurde.
	<b>Verhalten der Ausgänge bei Stop</b> Wählen Sie aus der Liste eine der folgenden Optionen aus, um zu konfigurieren, auf welche Weise die Werte der Ausgangskanäle im Fall eines Steuerungsstopps gehandhabt werden sollen: <ul style="list-style-type: none"> <li>● <b>Werte beibehalten</b></li> <li>● <b>Alle Ausgänge auf Standardwert setzen</b></li> <li>● <b>Programm ausführen</b></li> </ul>
	<b>Alle Variablen in allen Geräten aktualisieren</b> Wenn diese Option aktiviert ist, werden alle E/A-Variablen für sämtliche Geräte der aktuellen Steuerungskonfiguration in jedem Zyklus des Buszyklus-Tasks aktualisiert. Dies entspricht der Option <b>Variablen immer aktualisieren</b> , die im Dialogfeld <b>E/A-Abbild</b> für jedes Gerät separat eingestellt werden kann.

Element		Beschreibung
Buszyklus-Optionen	Buszyklus-Task	<p>Diese Konfigurationseinstellung gilt für alle Parameter der <b>Buszyklus-Task</b>, die in der Gerätestruktur verwendet werden.</p> <p>Einige Geräte mit zyklischen Aufrufen, wie z. B. der <b>CANopen-Manager</b>, können einer bestimmten Task zugeordnet werden. Wenn für die Einstellung des Geräts die Einstellung <b>Zykluseinstellungen des übergeordneten Busses verwenden</b> eingegeben wurde, wird die Einstellung für die Steuerung verwendet.</p> <p>Diese Auswahlliste enthält alle Tasks, die derzeit in der aktiven Anwendung definiert sind. Die Standardeinstellung ist „MAST-Task“.</p> <p><b>HINWEIS:</b> &lt;Unspezifiziert&gt; bedeutet, dass die Task im Modus der "Langsamsten zyklischen Task" ausgeführt wird.</p>
	Zusätzliche Einstellungen	<p><b>Force-Variablen für E/A-Abbild erzeugen</b> Nicht verwendet.</p> <p><b>Diagnose für Geräte aktivieren</b> Nicht verwendet.</p>

---

# Kapitel 9

## Konfiguration integrierter Ein- und Ausgänge

---

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Integrierte E/A-Konfiguration	94
Master-Geber-Eingangskonfiguration	102

## Integrierte E/A-Konfiguration

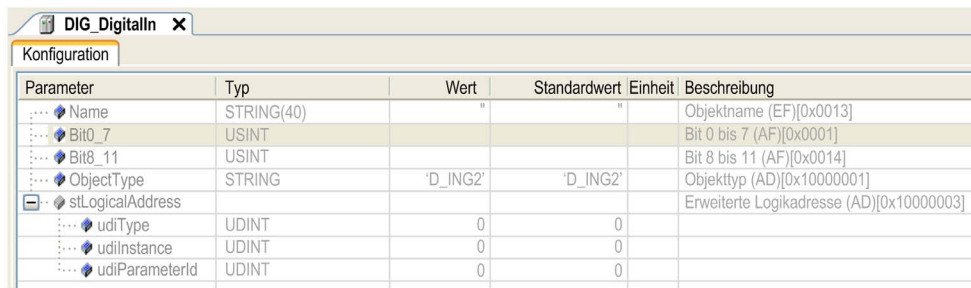
### Einführung

Der Funktionsbaustein Modicon LMC078 Motion Controller umfasst Folgendes:

- 12 integrierte Eingänge:
  - 8 Digitaleingänge: **DI\_0...DI\_7**
  - 4 erweiterte Digitaleingänge (Touchprobe und Interrupt): **ADI\_0** bis **ADI\_3**
- 8 integrierte Ausgänge:
  - 7 Digitalausgänge: **DQ\_0** bis **DQ\_6**
  - 1 Digitalausgang, als Watchdog-Ausgang konfigurierbar: **DQ\_WD**

### Gruppenkonfiguration der Digitaleingänge

Um die Digitaleingangsgruppe zu konfigurieren, doppelklicken Sie auf den Knoten **DIG\_DigitalIn** in der **Gerätebaumstruktur**.



Parameter	Typ	Wert	Standardwert	Einheit	Beschreibung
Name	STRING(40)	"	"	"	Objektname (EF)[0x0013]
Bit0_7	USINT				Bit 0 bis 7 (AF)[0x0001]
Bit8_11	USINT				Bit 8 bis 11 (AF)[0x0014]
ObjectType	STRING	'D_ING2'	'D_ING2'		Objekttyp (AD)[0x10000001]
stLogicalAddress					Erweiterte Logikadresse (AD)[0x10000003]
udiType	UDINT	0	0		
udiInstance	UDINT	0	0		
udiParameterId	UDINT	0	0		

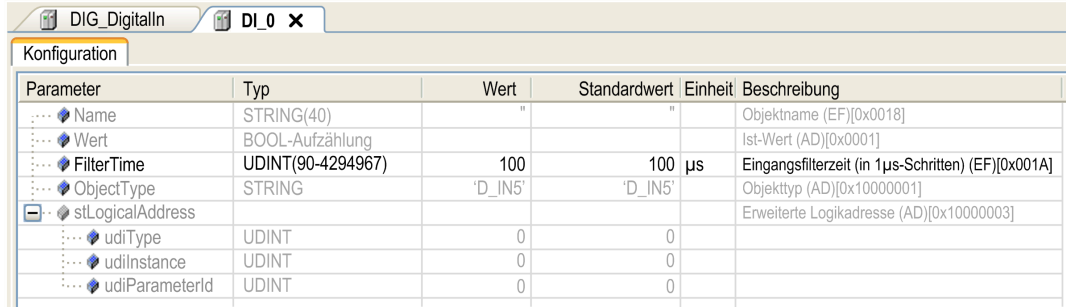
In der folgenden Tabelle werden die verschiedenen Parameter beschrieben:

Parameter	Zugriff	Param. Typ	Datentyp	Beschreibung
Name	R/W <sup>(*)</sup>	EF	STRING(40)	Symbolischer Name des Konfigurationsobjekts.
Bit0_7	R	AF	USINT	Wert der Digitaleingänge <b>DI_0</b> bis <b>DI_7</b> , jedes Bit wird einem Eingang zugewiesen. Bit x = Wert des Eingangs <b>DI_x</b>
Bit8_11	R	AF	USINT	Wert der erweiterten Digitaleingänge <b>ADI_0</b> bis <b>ADI_3</b> , jedes Bit wird einem erweiterten Eingang zugewiesen. Bit x = Wert des Eingangs <b>ADI_x</b>
ObjectType	R	AD	STRING	Objekttyp.
stLogicalAddress	R	AD	ST_LogicalAddress	Logische Adresse der Eingangsgruppe.

<sup>(\*)</sup> Weitere Informationen zu den Parameter-Zugriffsrechten finden Sie unter Parametertypen (siehe Seite 29).

## Konfiguration der Digitaleingänge

Um einen Digitaleingang zu konfigurieren, doppelklicken Sie auf den Knoten **DI\_x** in der Gerätebaumstruktur.



Parameter	Typ	Wert	Standardwert	Einheit	Beschreibung
Name	STRING(40)	"	"		Objektname (EF)[0x0018]
Wert	BOOL-Aufzählung				Ist-Wert (AD)[0x0001]
FilterTime	UDINT(90-4294967)	100	100	µs	Eingangsfilterszeit (in 1µs-Schritten) (EF)[0x001A]
ObjectType	STRING	'D_IN5'	'D_IN5'		Objekttyp (AD)[0x10000001]
stLogicalAddress					Erweiterte Logikadresse (AD)[0x10000003]
udiType	UDINT	0	0		
udiInstance	UDINT	0	0		
udiParameterId	UDINT	0	0		

In der folgenden Tabelle werden die Eingangsparameter beschrieben:

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
Name	R/W(*)	EF	STRING(40)	"	"	Symbolischer Name des Konfigurationsobjekts.
Value	R	AD	BOOL Enum	L / 0 H / 1	-	Wert des Digitaleingangs.
FilterTime	R/W	EF	UDINT	90 bis 4294967	100	Filterzeit des Eingangs in µs.
ObjectType	R	AD	STRING	D_IN5	D_IN5	Objekttyp.
stLogicalAddress	R	AD	ST_LogicalAddress	-	-	Logische Adresse der Eingänge.

(\*) Weitere Informationen zu den Parameter-Zugriffsrechten finden Sie unter Parametertypen (siehe Seite 29).

### Konfiguration der erweiterten Digitaleingänge

Um einen erweiterten Digitaleingang zu konfigurieren, doppelklicken Sie auf den Knoten **ADI\_x** in der **Gerätebaumstruktur**.

Parameter	Typ	Wert	Standardwert	Einheit	Beschreibung
Name	STRING(40)	"	"		Objektname (EF)[0x0018]
Wert	BOOL-Aufzählung				Ist-Wert (AD)[0x0001]
FilterTime	UDINT(90-4294967)	100	100	µs	EingangsfILTERzeit (in 1µs-Schritten) (EF)[0x001A]
CaptureState	DINT-Aufzählung	Inaktiv / 0	Inaktiv / 0		Status der Erfassungsfunktion (AD)[0x0005]
SensorDelay	UDINT(-100-4294967)	0.0	0.0	msec	Sensorverzögerung (EF)[0x0004]
Zähler	UDINT	0	0		Zähler für Eingang (EF)[0x000C]
Aktivieren	BOOL-Aufzählung	Aus / 0	Aus / 0		Interrupt für Eingang aktivieren (EF)[0x000D]
ExtEventEdge	DINT-Aufzählung	Positiv / 1	Positiv / 1		Aktive Flanke für Eingang (EF)[0x001B]
ObjectType	STRING	'D_IN62'	'D_IN62'		Objekttyp (AD)[0x10000001]
stLogicalAddress					Erweiterte Logikadresse (AD)[0x10000003]
udiType	UDINT	0	0		
udiInstance	UDINT	0	0		
udiParameterId	UDINT	0	0		

In der folgenden Tabelle werden die erweiterten Eingangsparameter beschrieben:

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
Name	R/W(*)	EF	STRING(40)	"	"	Symbolischer Name des Konfigurationsobjekts.
Value	R	AD	BOOL Enum	L / 0 H / 1	-	Wert des erweiterten Digitaleingangs.
FilterTime	R/W	EF	UDINT	90 bis 4294967	100	Filterzeit des Eingangs in µs.
CaptureState	R	AD	DINT Enum	inactive / 0 active / 1 captured / 2 overflow / 3 disabled / 4 not ready / 5 virtual / 6	inactive / 0	Zustand der Erfassungsfunktion.
SensorDelay	R/W	EF	LREAL	-100 bis 100	0	Sensorverzögerung in ms.
Counter	R	EF	UDINT	-	0	Zähler des Eingangs.
Enable	R/W(*)	EF	BOOL Enum	off / 0 on / 1	Aus / 0	Aktiviert die Interrupt-Funktion des Eingangs.
ExtEventEdge	R/W	EF	DINT Enum	negative / 0 positive / 1 negative and positive / 2	Positiv / 1	Definiert die aktive Flanke des Eingangs.

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
ObjectType	R	AD	STRING	D_IN62	D_IN62	Objekttyp.
stLogicalAddress	R	AD	ST_LogicalAddress	-	-	Logische Adresse der erweiterten Eingänge.

(\*) Weitere Informationen zu den Parameter-Zugriffsrechten finden Sie unter Parametertypen (siehe Seite 29).

### Gruppenkonfiguration der Digitalausgänge

Um die Digitalausgangsgruppe zu konfigurieren, doppelklicken Sie auf den Knoten **DQG\_DigitalOut** in der **Gerätebaumstruktur**.

Parameter	Typ	Wert	Standardwert	Einheit	Beschreibung
Name	STRING(40)	"	"		Objektname (EF)[0x0008]
Bit0_7	USINT(0-255)	2#00000000	2#00000000		Bit 0 bis 7 (EF)[0x0001]
DiagMask	UINT	2#11111111	2#11111111		DiagnosisMask (EF)[0x0004]
OpenloadDiagMsk	UINT	2#11111111	2#11111111		DiagnosisMask (EF)[0x0005]
OverloadDiagMsk	UINT	2#11111111	2#11111111		DiagnosisMask (EF)[0x0006]
ObjectType	STRING	'D_OUTG5'	'D_OUTG5'		Objekttyp (AD)[0x10000001]
stLogicalAddress					Erweiterte Logikadresse (AD)[0x10000003]
udiType	UDINT	0	0		
udiInstance	UDINT	0	0		
udiParameterId	UDINT	0	0		

In der folgenden Tabelle werden die verschiedenen Parameter beschrieben:

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
Name	R/W(*)	EF	STRING(40)	"	"	Symbolischer Name des Konfigurationsobjekts.
Bit0_7	R/W(*)	EF	USINT	0	0	Wert der Digitalausgänge <b>DQ_0</b> bis <b>DQ_7</b> , jedes Bit wird einem Ausgang zugewiesen. Bit x = Wert des Ausgangs <b>DQ_x</b>

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
DiagMask	R/W	EF	UINT	2#11111111	2#11111111	Aktiviert die Diagnosemeldung 8788 Wiring error für jeden Ausgang, jedes Bit ist einem Ausgang zugewiesen. Bit x = 0, der Ausgang x wird nicht überwacht, und die Diagnosemeldung wird nicht angezeigt.
OpenloadDiagMask	R/W	EF	UINT	2#11111111	2#11111111	Aktiviert die Diagnosemeldung 8788 Wiring error / Openload für jeden Ausgang, jedes Bit ist einem Ausgang zugewiesen. Bit x = 0, der Ausgang x wird nicht überwacht, und die Diagnosemeldung wird nicht angezeigt.
OverloadDiagMask	R/W	EF	UINT	2#11111111	2#11111111	Aktiviert die Diagnosemeldung 8788 Wiring error / Overload für jeden Ausgang, jedes Bit ist einem Ausgang zugewiesen. Bit x = 0, der Ausgang x wird nicht überwacht, und die Diagnosemeldung wird nicht angezeigt.
ObjectType	R	AD	STRING	D_OUTG5	D_OUTG5	Objekttyp.
stLogicalAddress	R	AD	ST_LogicalAddress	-	-	Logische Adresse der Ausgangsgruppe.

(\*) Weitere Informationen zu den Parameter-Zugriffsrechten finden Sie unter Parametertypen (siehe Seite 29).

## Konfiguration der Digitalausgänge

Um einen Digitalausgang zu konfigurieren, doppelklicken Sie auf den Knoten **DQ\_x** in der Gerätebaumstruktur.

Parameter	Typ	Wert	Standardwert	Einheit	Beschreibung
Name	STRING(40)	"	"		Objektname (EF)[0x0007]
Wert	BOOL-Aufzählung	L / 0	L / 0		Ist-Wert (ED)[0x0001]
Status	DINT-Aufzählung	Standard / 0	Standard / 0		Status (AF)[0x0002]
EnableDiagMsg	BOOL-Aufzählung	H / 1	H / 1		Text "Diagnose freigeben" (EF)[0x0003]
OpenloadDiagMsg	BOOL-Aufzählung	H / 1	H / 1		Text "Openload-Diagnose freigeben" (EF)[0x0004]
OverloadDiagMsg	BOOL-Aufzählung	H / 1	H / 1		Text "Overload-Diagnose freigeben" (EF)[0x0005]
ObjectType	STRING	'D_OUT1'	'D_OUT1'		Objekttyp (AD)[0x10000001]
stLogicalAddress					Erweiterte Logikadresse (AD)[0x10000003]
udiType	UDINT	0	0		
udiInstance	UDINT	0	0		
udiParameterId	UDINT	0	0		

In der folgenden Tabelle werden die Ausgangsparameter beschrieben:

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
Name	R/W(*)	EF	STRING(40)	"	"	Symbolischer Name des Konfigurationsobjekts.
Value	R/W(*)	ED	BOOL Enum	L / 0 H / 1	L / 0	Zeigt den Wert des Digitalausgangs an.
Status	R	AF	DINT Enum	default / 0 openload / 1 overload / 2	default / 0	Zeigt den Zustand des Digitalausgangs an. <ul style="list-style-type: none"> <li>● 0: Standardzustand</li> <li>● 1: Keine Last verfügbar</li> <li>● 2: Kurzschluss aufgetreten</li> </ul>
EnableDiagMsg	R/W(*)	EF	BOOL Enum	L / 0 H / 1	H / 1	0 = Der Ausgang gibt nicht die Diagnosemeldung 8788 Wiring error aus. 1 = Der Ausgang gibt die Diagnosemeldung 8788 Wiring error aus.
OpenloadDiagMsg	R/W(*)	EF	BOOL Enum	L / 0 H / 1	H / 1	0 = Der Ausgang gibt nicht die Diagnosemeldung 8788 Wiring error / Openload aus. 1 = Der Ausgang gibt die Diagnosemeldung 8788 Wiring error / Openload aus.

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
OverloadDiagMsg	R/W(*)	EF	BOOL Enum	L / 0 H / 1	H / 1	0 = Der Ausgang gibt nicht die Diagnosemeldung 8788 Wiring error / Overload aus. 1 = Der Ausgang gibt die Diagnosemeldung 8788 Wiring error / Overload aus.
ObjectType	R	AD	STRING	D_OUT1	D_OUT1	Objekttyp.
stLogicalAddress	R	AD	ST_LogicalAddress	-	-	Logische Adresse der Ausgänge.

(\*) Weitere Informationen zu den Parameter-Zugriffsrechten finden Sie unter Parametertypen (siehe Seite 29).

### Konfiguration des Watchdog-Ausgangs

Um den Watchdog-Ausgang zu konfigurieren, doppelklicken Sie auf den Knoten **DQ\_WD** in der Gerätebaumstruktur.

Parameter	Typ	Wert	Standardwert	Einheit	Beschreibung
Name	STRING(40)	"	"		Objektname (EF)[0x0007]
Wert	BOOL-Aufzählung	L / 0	L / 0		Ist-Wert (ED)[0x0001]
Status	DINT-Aufzählung	Standard / 0	Standard / 0		Status (AF)[0x0002]
EnableDiagMsg	BOOL-Aufzählung	H / 1	H / 1		Text "Diagnose freigeben" (EF)[0x0003]
OpenloadDiagMsg	BOOL-Aufzählung	H / 1	H / 1		Text "Openload-Diagnose freigeben" (EF)[0x0004]
OverloadDiagMsg	BOOL-Aufzählung	H / 1	H / 1		Text "Overload-Diagnose freigeben" (EF)[0x0005]
WDOutEnable	BOOL-Aufzählung	Aus / 0	Aus / 0		Ausgang wird von Watchdog gesteuert (ED)[0x0008]
ObjectType	STRING	'D_OUT2'	'D_OUT2'		Objekttyp (AD)[0x10000001]
stLogicalAddress					Erweiterte Logikadresse (AD)[0x10000003]
udiType	UDINT	0	0		
udiInstance	UDINT	0	0		
udiParameterId	UDINT	0	0		

In der folgenden Tabelle werden die Watchdog-Ausgangsparameter beschrieben:

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
Name	R/W(*)	EF	STRING(40)	"	"	Symbolischer Name des Konfigurationsobjekts.
Value	R/W(*)	ED	BOOL Enum	L / 0 H / 1	L / 0	Zeigt den Wert des Watchdog-Ausgangs an.

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
Status	R	AF	DINT Enum	default / 0 openload / 1 overload / 2	default / 0	Zeigt den Zustand des Watchdog-Ausgangs an. <ul style="list-style-type: none"> <li>● 0: Standardzustand</li> <li>● 1: Keine Last verfügbar</li> <li>● 2: Kurzschluss aufgetreten</li> </ul>
EnableDiagMsg	R/W(*)	EF	BOOL Enum	L / 0 H / 1	H / 1	0 = Der Ausgang gibt nicht die Diagnosemeldung 8788 Wiring error aus. 1 = Der Ausgang gibt die Diagnosemeldung 8788 Wiring error aus.
OpenloadDiagMsg	R/W(*)	EF	BOOL Enum	L / 0 H / 1	H / 1	0 = Der Ausgang gibt nicht die Diagnosemeldung 8788 Wiring error / Openload aus. 1 = Der Ausgang gibt die Diagnosemeldung 8788 Wiring error / Openload aus.
OverloadDiagMsg	R/W(*)	EF	BOOL Enum	L / 0 H / 1	H / 1	0 = Der Ausgang gibt nicht die Diagnosemeldung 8788 Wiring error / Overload aus. 1 = Der Ausgang gibt die Diagnosemeldung 8788 Wiring error / Overload aus.
WDOutEnable	R/W	ED	BOOL Enum	off / 0 on / 1	off / 0	0 = Der Watchdog-Ausgang ist deaktiviert. 1 = Der Watchdog-Ausgang ist aktiviert und wird vom System-Watchdog ( <i>siehe Seite 47</i> ) gesteuert.
ObjectType	R	AD	STRING	D_OUT2	D_OUT2	Objekttyp.
stLogicalAddress	R	AD	ST_LogicalAddress	-	-	Logische Adresse des Watchdog-Ausgangs.

(\*) Weitere Informationen zu den Parameter-Zugriffsrechten finden Sie unter Parametertypen (*siehe Seite 29*).

## Master-Geber-Eingangskonfiguration

### Einführung

Die Steuerung verfügt über eine spezifische Hardware-Geberschnittstelle, die folgende Geber unterstützt:

- Inkrementalgeber (RS422)
- Absoluter Geber (SinCos Hiperface)

Diese Funktion verbindet einen Geber, um eine Position zu erreichen, die mit dem Sercos-Echtzeitbus synchron ist. Diese erworbene Funktion kann dann als Masterachse für Bewegungsantriebe auf dem Sercos verwendet werden.

Die Konfiguration des Mastergebers ist in zwei Teile untergliedert:

- Der Geberknoten, der die Hardwarekonfiguration unterstützt.
- Der Knoten **SoftMotion\_Encoder**, der die Skalierungskonfiguration unterstützt.

### Hinzufügen eines Gebers

Um der Steuerung einen Geber hinzuzufügen, wählen Sie das Element **Inkrementalgebereingang** oder **SinCos-Gebereingang** im **Hardwarekatalog** aus, ziehen Sie das Element in die **Gerätebaumstruktur** und legen Sie es auf dem Steuerungsknoten ab.

**Ergebnis:** Der Geber wird Ihrer Steuerung als neuer Unterknoten hinzugefügt, und ein Knoten namens **SoftMotion\_Encoder** wird als Unterknoten Ihrer Steuerung hinzugefügt.

Weitere Informationen zum Hinzufügen von Geräten in einem Projekt finden Sie unter:

- Verwenden der Methode Drag&Dop (*siehe SoMachine, Programmierhandbuch*) (Ziehen und Ablegen)
- Verwenden der Kontextmenüs oder Plus-Schaltflächen (*siehe SoMachine, Programmierhandbuch*)

## Konfiguration des Inkrementalgebers

Um den Inkrementalgeber zu konfigurieren, doppelklicken Sie in der **Gerätebaumstruktur** auf den Geberknoten.

Parameter	Typ	Wert	Standardwert	Einheit	Beschreibung
Name	STRING(40)	"	"		Objektname (EF)[0x0014]
Aktivieren	BOOL-Aufzählung	Ein / 1	Ein / 1		Aktivieren (EF)[0x0001]
Status	DINT-Aufzählung	Nicht bereit / 0	Nicht bereit / 0		Status des Inkrementalgebers (AD)[0x000E]
Filter	DINT(0-1024)	0	0	ms	Filter (EF)[0x0004]
CheckOff	BOOL-Aufzählung	Nein / 0	Nein / 0		Geber-CheckOff (EF)[0x0005]
NullspurCheckOff	BOOL-Aufzählung	Nein / 0	Nein / 0		Nullspur-CheckOff (EF)[0x0013]
ZeroTrackStart	BOOL-Aufzählung	False / 0	False / 0		Nullspur-Erkennungsstart (EF)[0x0009]
ZeroTrackDetected	BOOL-Aufzählung	False / 0	False / 0		Nullspur erkannt (AD)[0x000A]
DiagClass	DINT				Diagnoseklasse (AD)[0x0077]
DiagCode	DINT				Diagnosecode (AD)[0x000F]
DiagSource					Diagnosequelle (AD)[0x0078]
DiagMsg	STRING(39)	"	"		Diagnosetext (AD)[0x0011]
DiagExtMsg	STRING(14)	"	"		Erweiterte Diagnosemeldung (AD)[0x0079]
ObjectType	STRING	'INC_IN2'	'INC_IN2'		Objekttyp (AD)[0x10000001]
stLogicalAddress					Erweiterte Logikadresse (AD)[0x10000003]

In der folgenden Tabelle werden die Parameter des Inkrementalgebers beschrieben:

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
Name	R/W(*)	EF	STRING(40)	"	"	Symbolischer Name des Konfigurationsobjekts.
Enable	R/W	EF	BOOL Enum	off / 0 on / 1	on / 1	Aktiviert den Geber: <ul style="list-style-type: none"> <li>0 = Der Geber wird nicht vom Echtzeitprozess verarbeitet.</li> <li>1 = Der Geber wird vom Echtzeitprozess verarbeitet.</li> </ul>

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
State	R	AD	DINT Enum	not ready / 0 initialization / 1 no sync / 2 ready / 3	not ready / 0	Zeigt die Verfügbarkeit und Gültigkeit der Geberpositionsdaten an: <ul style="list-style-type: none"> <li>● 0 = Der Geber oder die Geberverarbeitung sind nicht bereit.</li> <li>● 1 = Die Geberverarbeitung wird initialisiert.</li> <li>● 2 = Die Geberverarbeitung läuft nicht synchron mit dem RTP.</li> <li>● 3 = Die Geschwindigkeit und Positionswerte des Gebers sind gültig.</li> </ul>
Filter	R/W	EF	DINT	0 bis 1024	0	Filterwert in ms. Der Filter ist ein Tiefpassfilter und wirkt sich auf die tatsächliche Geschwindigkeit aus. Wenn der Filter zu hoch gesetzt ist, kann dies zu einer Oszillation des Systems führen, abhängig vom Geschwindigkeitspfad.
CheckOff	R/W	EF	BOOL Enum	no / 0 yes / 1	no / 0	Deaktiviert die Geberüberwachung: <ul style="list-style-type: none"> <li>● 0 = Geberüberprüfung aktiv.</li> <li>● 1 = Geberüberwachung nicht aktiv (Spur- und Kabelüberwachung).</li> </ul>
ZeroTrack-CheckOff	R/W	EF	BOOL Enum	no / 0 yes / 1	no / 0	Deaktiviert die Nullspurüberwachung des Gebers: <ul style="list-style-type: none"> <li>● 0 = Nullspurüberwachung aktiv.</li> <li>● 1 = Nullspurüberwachung nicht aktiv.</li> </ul>
ZeroTrack-Start	R/W <sup>(*)</sup>	EF	BOOL Enum	FALSE / 0 TRUE / 1	FALSE / 0	Eine steigende Flanke dieses Parameters aktiviert die Nullspurerkennung: <ul style="list-style-type: none"> <li>● 0 = Nullspurerkennung deaktiviert.</li> <li>● 1 = Nullspurerkennung aktiviert.</li> </ul>

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
ZeroTrack-Detected	R	AD	BOOL Enum	FALSE / 0 TRUE / 1	FALSE / 0	Wenn die Nullspurerkennung aktiviert ist, gibt dieser Parameter die Nullspurerkennung an: <ul style="list-style-type: none"> <li>● 0 = Nullspur nicht erkannt.</li> <li>● 1 = Nullspur erkannt.</li> </ul> Wenn eine Nullspur erkannt wird, wird die Position des Gebers auf 0 gesetzt.
DiagClass	R	AD	DINT	-	-	Zeigt die Diagnoseklasse ( <i>siehe Seite 272</i> ) in Dezimalcode an.
DiagCode	R	AD	DINT	-	-	Zeigt den Diagnosecode ( <i>siehe Seite 272</i> ) in Dezimalcode an.
DiagSource	R	AD	ST_LogicalAddress	-	-	Logische Adresse der Diagnosequelle.
DiagMsg	R	AD	STRING	-	-	Diagnosemeldung.
DiagExtMsg	R	AD	STRING	-	-	Erweiterte Diagnosemeldung.
ObjectType	R	AD	STRING	INC_IN2	INC_IN2	Objekttyp.
stLogicalAddress	R	AD	ST_LogicalAddress	-	-	Logische Adresse des Inkrementalgebers.

(\*) Weitere Informationen zu den Parameter-Zugriffsrechten finden Sie unter Parametertypen (*siehe Seite 29*).

## Konfiguration des Hiperface (SinCos)-Gebers

Um den Hiperface-Geber zu konfigurieren, doppelklicken Sie in der **Gerätebaumstruktur** auf den Geberknoten.

Parameter	Typ	Wert	Standardwert	Einheit	Beschreibung
Name	STRING(40)	"	"		Objektname (EF)[0x0028]
Aktivieren	BOOL-Aufzählung	Ein / 1	Ein / 1		Aktivieren (EF)[0x0001]
Filter	DINT(0-1024)	0	0	ms	Filter (EF)[0x0003]
CheckOff	BOOL-Aufzählung	Nein / 0	Nein / 0		Geber-CheckOff (EF)[0x0004]
EncoderType	DINT				Geber typ 22-SRS 27-SRM 02-SCS 07-SCM (AK)[0x0008]
Status	DINT-Aufzählung	Nicht bereit / 0	Nicht bereit / 0		Phys. Master-Geber-Status (AD)[0x0024]
SignalQuality	UDINT			%	Qualität der analogen Positionssignale (AF)[0x0020]
SignalQualityLimit	UDINT(0-100)	50	50	%	Fehlergrenze der analogen Positionssignale (EF)[0x0021]
DiagClass	DINT				Diagnoseklasse (AD)[0x0077]
DiagCode	DINT				Diagnosecode (AD)[0x0025]
DiagSource					Diagnosequelle (AD)[0x0079]
DiagMsg	STRING(39)	"	"		Diagnosetext (AD)[0x0027]
DiagExtMsg	STRING(14)	"	"		Erweiterte Diagnosemeldung (AD)[0x0078]
ObjectType	STRING	'P_ENC2'	'P_ENC2'		Objekttyp (AD)[0x10000001]
stLogicalAddress					Erweiterte Logikadresse (AD)[0x10000003]

In der folgenden Tabelle werden die Parameter des Hiperface-Gebers beschrieben:

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
Name	R/W(*)	EF	STRING(40)	"	"	Symbolischer Name des Konfigurationsobjekts.
Enable	R/W	EF	BOOL Enum	off / 0 on / 1	on / 1	Aktiviert den Geber: <ul style="list-style-type: none"> <li>● 0 = Der Geber wird nicht vom Echtzeitprozess verarbeitet.</li> <li>● 1 = Der Geber wird vom Echtzeitprozess verarbeitet.</li> </ul>
Filter	R/W	EF	DINT	0 bis 1024	0	Filterwert in ms. Der Filter ist ein Tiefpassfilter und wirkt sich auf die tatsächliche Geschwindigkeit aus. Wenn der Filter zu hoch gesetzt ist, kann dies zu einer Oszillation des Systems führen, abhängig vom Geschwindigkeitspfad.

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
CheckOff	R/W	EF	BOOL Enum	no / 0 yes / 1	no / 0	Deaktiviert die Geberüberwachung: <ul style="list-style-type: none"> <li>● 0 = Geberüberwachung aktiv.</li> <li>● 1 = Geberüberwachung nicht aktiv. Die Diagnosemeldung 8601 Master Encoder signal out of range ist deaktiviert.</li> </ul>
Encoder-Type	R	AK	DINT	-	-	Zeigt den unterstützten SinCos Hiperface-Gebertyp an (Lesen bei Steuerungsstart): <ul style="list-style-type: none"> <li>● 00h: Wert noch nicht gelesen</li> <li>● 02h: SCS 60/70</li> <li>● 07h: SCM 60/70 Multi Turn</li> <li>● 22h: SRS 50/60/64 und SCK 25/35/40/45/53 Single Turn</li> <li>● 27h: SRM 50/60/64 und Motor 25/35/40/45/53 Multi Turn</li> <li>● 32h: SKS 36 Single Turn</li> <li>● 37h: SKM 36 Multi Turn</li> <li>● 42h: SEK 52 Single Turn</li> </ul>

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
State	R	AD	DINT Enum	not ready / 0 initialization / 1 no sync / 2 get type / 3 ready / 10 read position / 11 write position / 12 read error code / 13	not ready / 0	Zeigt die Verfügbarkeit und Gültigkeit der Geberpositionsdaten an: <ul style="list-style-type: none"> <li>● 0 = Der Geber oder die Geberverarbeitung sind nicht bereit.</li> <li>● 1 = Die Geberverarbeitung wird initialisiert.</li> <li>● 2 = Die Geberverarbeitung läuft nicht synchron mit dem RTP.</li> <li>● 3 = Der Gebertyp wird gelesen.</li> <li>● 4 bis 9 = Ungültiger Zustand.</li> <li>● 10 = Die Geschwindigkeit und Positionswerte des Gebers sind gültig.</li> <li>● 11 = Die absolute Position wird aus dem Geber gelesen.</li> <li>● 12 = Die absolute Position wird in den Geber geschrieben.</li> <li>● 13 = Der Fehlercode wird aus dem Geber gelesen.</li> </ul>
Signal-Quality	R	AF	UDINT	-	-	Beschreibt die Signalqualität der analogen Sinus- und Cosinus-Spur des Gebers (in %). Die Sinus- und Cosinus-Signale müssen der folgenden Formel entsprechen: $\text{Sinus-Signal}^2 + \text{Cosinus-Signal}^2 = 1$ (100 %) Dieser Parameter stellt diese Formel normalisiert (100 %) dar.
Signal-QualityLimit	R	EF	UDINT	50	50	Dieser Parameter legt fest, bei welchem Wert der SignalQuality der Diagnosecode 8601 Master encoder signal out of the range ausgegeben wird.

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
DiagClass	R	AD	DINT	-	-	Zeigt die Diagnoseklasse ( <i>siehe Seite 272</i> ) in Dezimalcode an.
DiagCode	R	AD	DINT	-	-	Zeigt den Diagnosecode ( <i>siehe Seite 272</i> ) in Dezimalcode an.
DiagSource	R	AD	ST_LogicalAddress	-	-	Logische Adresse der Diagnosequelle.
DiagMsg	R	AD	STRING	-	-	Diagnosemeldung.
DiagExtMsg	R	AD	STRING	-	-	Erweiterte Diagnosemeldung.
ObjectType	R	AD	STRING	P_ENC2	P_ENC2	Objektyp.
stLogicalAddress	R	AD	ST_LogicalAddress	-	-	Logische Adresse des Hiperface-Gebers.

### SoftMotion-Geberkonfiguration

Um den SoftMotion-Geber zu konfigurieren, doppelklicken Sie auf den Knoten **SoftMotion\_Encoder** in der **Gerätebaumstruktur**.

Die Konfiguration des SoftMotionGebers wird in der Online-Hilfe zu SoMachine im Kapitel *Programmieren mit SoMachine/SoftMotion/SoftMotion-Geräteeditor* beschrieben.



---

# Kapitel 10

## Kommunikationsmodule

---

### Einführung

In diesem Kapitel wird das Hinzufügen und Konfigurieren eines Kommunikationsmoduls erläutert.

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Abschnitte:

Abschnitt	Thema	Seite
10.1	Konfiguration des PROFIBUS DP-Slave-Moduls	112
10.2	Konfiguration des EtherNet/IP-Adapters	123
10.3	Konfiguration eines Ethernet/IP-Scanners	134

# Abschnitt 10.1

## Konfiguration des PROFIBUS DP-Slave-Moduls

---

### Einleitung

In diesem Abschnitt wird die Konfiguration des VW3E704000000 PROFIBUS DP-Moduls beschrieben.

### Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Hinzufügen eines PROFIBUS DP-Slave-Moduls	113
Konfiguration des PROFIBUS DP-Slave-Moduls	115
Azyklischer Datenaustausch	120

## Hinzufügen eines PROFIBUS DP-Slave-Moduls

### Überblick

Mit dem PROFIBUS-Protokoll werden Daten nach dem Master-Slave-Prinzip ausgetauscht. Die Kommunikation kann nur vom Master initialisiert werden. Die Slaves antworten auf Requests von den Mastern. Auf demselben Bus können mehrere Master vorhanden sein. In diesem Fall kann die Slave-E/A von allen Mastern gelesen werden. Es hat jedoch nur ein Master Schreibzugriff auf die Ausgänge. Die Anzahl ausgetauschter Datenelemente wird bei der Konfiguration definiert.

Von diesem Modul werden zwei Arten von Datenaustauschdiensten unterstützt:

- Zyklischer Austausch von E/A-Frames
- Azyklischer Datenaustausch mit der PROFIBUS DPV1-Funktion

### Hinzufügen eines PROFIBUS DP-Slave-Moduls

Wählen Sie das **PROFIBUS-DPV1-Slave**-Modul im **Hardwarekatalog** aus, ziehen Sie es in die **Gerätebaumstruktur**, und legen Sie es auf dem Steuerungsknoten ab.

Weitere Informationen zum Hinzufügen von Geräten in einem Projekt finden Sie unter:

- Verwenden der Methode Drag&Dop (*siehe SoMachine, Programmierhandbuch*) (Ziehen und Ablegen)
- Verwenden der Kontextmenüs oder Plus-Schaltflächen (*siehe SoMachine, Programmierhandbuch*)

### Hinzufügen virtueller E/A-Geräte

Unter einem PROFIBUS DP-Slave-Modul können Sie ein oder mehrere virtuelle E/A-Geräte hinzufügen.

Das PROFIBUS DP-Slave-Modul fungiert als Vermittler zwischen dem PROFIBUS-Master und der Steuerung, und Daten werden mittels virtueller E/A-Geräte ausgetauscht, die Sie bei der Konfiguration des Moduls definieren. Bei den virtuellen Geräten handelt es sich nicht um physische E/A-Module, sondern um logische Eingangs- und Ausgangsobjekte im Modul, die Sie dem Arbeitsspeicher in der Steuerung zuordnen können. Der Lese- und Schreibzugriff auf diese Eingangs- und Ausgangsobjekte erfolgt durch den PROFIBUS-Master. Das Modul wiederum liest diese Daten und schreibt sie an E/A-Speicherorte in der Steuerung, sodass Sie diese Daten in Ihrem Anwendungsprogramm nutzen können.

Wählen Sie das virtuelle E/A-Gerät im **Hardwarekatalog** aus, ziehen Sie es in die **Gerätebaumstruktur**, und legen Sie es auf dem Knoten **PROFIBUS-DPV1-Slave** ab.

Weitere Informationen zum Hinzufügen von Geräten in einem Projekt finden Sie unter:

- Verwenden der Methode Drag&Dop (*siehe SoMachine, Programmierhandbuch*) (Ziehen und Ablegen)
- Verwenden der Kontextmenüs oder Plus-Schaltflächen (*siehe SoMachine, Programmierhandbuch*)

Die virtuellen E/A-Geräte, die Sie im Modul definieren, können Eingangs- oder Ausgangsgeräte sein. Ihre Größe kann gemäß der Definition in der unten stehenden Tabelle variieren:

Name	Anzahl E/A	Format
<b>X byte input (0x**)</b>	X = 1, 2, 3, 4, 8, 12, 16, 20, 32 oder 64	Byte
<b>X byte input con (0x**)</b>	X = 1, 2, 3, 4, 8, 12, 16, 20, 32 oder 64	
<b>X byte output (0x**)</b>	X = 1, 2, 3, 4, 8, 12, 16, 20, 32 oder 64	
<b>X byte output con (0x**)</b>	X = 1, 2, 3, 4, 8, 12, 16, 20, 32 oder 64	
<b>X word input (0x**)</b>	X = 1, 2, 3, 4, 8, 12, 16, 20, 32 oder 64	Wort
<b>X word input con (0x**)</b>	X = 1, 2, 3, 4, 8, 12, 16, 20, 32 oder 64	
<b>X word output (0x**)</b>	X = 1, 2, 3, 4, 8, 12, 16, 20, 32 oder 64	
<b>X word output con (0x**)</b>	X = 1, 2, 3, 4, 8, 12, 16, 20, 32 oder 64	

Sobald Sie diese virtuellen Eingangs- und/oder Ausgangsgeräte im Modul definiert haben, können Sie diese Geräte Arbeitsspeicherorten in der Steuerung zuordnen. Welchem Typ von Speicherobjekten Sie diese virtuellen E/A-Geräte zuordnen, hängt von dem Datenaustausch ab, den Sie zwischen Master- und Slave-Geräte definieren.

## Konfiguration des PROFIBUS DP-Slave-Moduls

### Konfiguration des PROFIBUS DP-Slave-Moduls

Doppelklicken Sie in der **Gerätebaumstruktur** auf **PBS\_Slave (PROFIBUS-DPV1-Slave)**.

Parameter	Typ	Wert	Standardwert	Einheit
InitParameters				
BusAddr	BYTE(2-126)	2	2	
BaudRate	BYTE-Aufzählung	Baudrate_AUTO / 15	Baudrate_AUTO / 15	
Anschluss	INT-Aufzählung	Front / 1	Front / 1	
Nicht verwendet	BYTE-Aufzählung	NOT_IN_USE	NOT_IN_USE	
WdgAktiviert	BOOL	TRUE	TRUE	
WdgZeit	UDINT(20-65535)	100	100	
DPV1Aktivieren	BOOL	TRUE	TRUE	
SyncUnterstützt	BOOL	TRUE	TRUE	
FreezeUnterstützt	BOOL	TRUE	TRUE	
FailsafeUnterstützt	BOOL	TRUE	TRUE	
AdressÄndernNichtUnterstützt	BOOL	TRUE	TRUE	
Intern				
Info				
Diag				

**Parameter in Schwarz** Schreib-/Lesezugriff

**Parameter in Grau** Schreibgeschützt

Auf der Registerkarte **PROFIBUS-Konfiguration** werden folgende Parameter bereitgestellt:

Parameter	Typ	Wert	Standardwert	Beschreibung
<b>InitParameters</b>				
BusAddr	BYTE	2 bis 126	2	Adresse des PROFIBUS DP-Slave.
BaudRate	BYTE Enum	0 bis 15	15	PROFIBUS-Übertragungsrate: <ul style="list-style-type: none"> <li>● 0: 9600 Baud</li> <li>● 1 = 19,2 KBAud</li> <li>● 2 = 93,75 KBAud</li> <li>● 3 = 187,5 KBAud</li> <li>● 4 = 500 KBAud</li> <li>● 6 = 1500 KBAud</li> <li>● 7 = 3 MBAud</li> <li>● 8 = 6 MBAud</li> <li>● 9 = 12 MBAud</li> <li>● 10 = 31,25 KBAud</li> <li>● 11 = 45,45 KBAud</li> <li>● 15 = Automatisch. Die auf dem Master festgelegte Übertragungsrate wird vom Slave erkannt.</li> </ul>

Parameter	Typ	Wert	Standardwert	Beschreibung
WdgEnabled	BOOL	TRUE FALSE	TRUE	Aktiviert den Watchdog: <ul style="list-style-type: none"> <li>● TRUE = Der Watchdog ist aktiviert. Der Wert des Parameters <code>WdgTime</code> wird als Watchdog-Timeout verwendet.</li> <li>● FALSE = Der Watchdog ist deaktiviert. Der Wert des Parameters <code>WdgTime</code> wird ignoriert.</li> </ul>
WdgTime	UDINT	20 bis 65535	100	Legt fest, nach welcher Zeit ein Slave vom Watchdog als nicht kommunizierend erkannt wird. Die Zeit wird in ms angegeben.
DPV1Enable	BOOL	TRUE FALSE	TRUE	Aktiviert die DPV1-Funktionen für die azyklische Kommunikation: <ul style="list-style-type: none"> <li>● TRUE = Der DPV1-Modus ist aktiviert.</li> <li>● FALSE = Der DPV1-Modus ist deaktiviert.</li> </ul>
SyncSupported	BOOL	TRUE FALSE	TRUE	Aktiviert den Sync-Modus, der den Sync-Befehl unterstützt: <ul style="list-style-type: none"> <li>● TRUE = Der Sync-Modus ist aktiviert.</li> <li>● FALSE = Der Sync-Modus ist deaktiviert.</li> </ul>
FreezeSupported	BOOL	TRUE FALSE	TRUE	Aktiviert den Freeze-Modus, der den Freeze-Befehl unterstützt: <ul style="list-style-type: none"> <li>● TRUE = Der Freeze-Modus ist aktiviert.</li> <li>● FALSE = Der Freeze-Modus ist deaktiviert.</li> </ul>
FailsafeSupported	BOOL	TRUE FALSE	TRUE	Aktiviert einen proprietären Failsafe-Modus (Ausfallsicherheit), wenn dies vom Gerät unterstützt wird: <ul style="list-style-type: none"> <li>● TRUE = Der Modus ist aktiviert.</li> <li>● FALSE = Der Modus ist deaktiviert.</li> </ul>
<b>Internal</b>				
Reserviert				
<b>Info</b>				
IdentNumber	WORD	3424	3424	Zeigt die Identifizierungsnummer des PROFIBUS DP-Slave-Moduls an.
VendorName	STRING	'Schneider Electric'	'Schneider Electric'	Zeigt den Herstellernamen des PROFIBUS DP-Slave-Moduls an.
ModelName	STRING	'PROFIBUS DPV1 slave'	'PROFIBUS DPV1 slave'	Zeigt den Modellnamen des PROFIBUS DP-Slave-Moduls an.
DriverInstance	DWORD	0	0	Zeigt die Kennung für diese Antriebsinstanz an.

Parameter	Typ	Wert	Standardwert	Beschreibung
<b>Diag / ChannelCommonStatusBlock</b>				
CommState	UDINT Enum	UNKNOWN / 0 NOT_CONFIGURED / 1 STOP / 2 IDLE / 3 OPERATE / 4	UNKNOWN / 0	Zeigt den aktuellen Netzwerkstatus des Kommunikationskanals an. <ul style="list-style-type: none"> <li>● 0 = Nicht bestimmbar.</li> <li>● 1 = Nicht konfiguriert.</li> <li>● 2 = Angehalten.</li> <li>● 3 = Leerlauf.</li> <li>● 4 = In Betrieb.</li> </ul>
CommError	UDINT Enum	Diagnosecodes (siehe Seite 118)	SUCCESS / 0x0	Zeigt den aktuellen Diagnosecode des Kommunikationskanals an.
ErrorCount	UDINT	-	0	Zeigt die Gesamtfehleranzahl an, die seit dem letzten Start oder Neustart entdeckt wurden.
<b>Diag / ChannelExtendedStatusBlock</b>				
Baudrate	UDINT Enum	0 bis 15	Baudrate_ AUTO / 15	Zeigt die angewendete Baudrate an. Der Parameter zeigt die vom Master verwendete Baudrate an, wenn im Initialisierungsparameter des Slaves der Wert AUTO festgelegt ist. PROFIBUS-Übertragungsrate: <ul style="list-style-type: none"> <li>● 0: 9600 Baud</li> <li>● 1 = 19,2 KBaud</li> <li>● 2 = 93,75 KBaud</li> <li>● 3 = 187,5 KBaud</li> <li>● 4 = 500 KBaud</li> <li>● 6 = 1500 KBaud</li> <li>● 7 = 3 MBaud</li> <li>● 8 = 6 MBaud</li> <li>● 9 = 12 MBaud</li> <li>● 10 = 31,25 KBaud</li> <li>● 11 = 45,45 KBaud</li> <li>● 15 = Automatisch. Der Master hat keine Verbindung zum Slave.</li> </ul>

## Diagnosecodes

Es wurde kein Fehler festgestellt:

Wert	Bedeutung
SUCCESS / 0x0	Kein Fehler erkannt.

Es wurde ein Laufzeitfehler festgestellt:

Wert	Bedeutung
WATCHDOG_TIMEOUT / 0xC000000C	Die Watchdog-Zeit wurde überschritten.

Es wurden Initialisierungsfehler festgestellt:

Wert	Bedeutung
INIT_FAULT / 0xC0000100	Die Initialisierung war nicht erfolgreich.
DATABASE_ACCESS_FAILED / 0xC0000101	Zugriff auf Datenspeicher war nicht erfolgreich.

Es wurden Konfigurationsfehler festgestellt:

Wert	Bedeutung
NOT_CONFIGURED / 0xC0000119	Das Modul ist nicht konfiguriert.
CONFIGURATION_FAULT / 0xC0000120	Es wurde ein Konfigurationsfehler festgestellt.
INCONSISTENT_DATA_SET / 0xC0000121	Es wurden inkonsistente Datensätze festgestellt.
DATA_SET_MISMATCH / 0xC0000122	Es wurden nicht übereinstimmende Datensätze festgestellt.
INSUFFICIENT_LICENSE / 0xC0000123	Es wurde eine unzureichende Lizenz festgestellt.
PARAMETER_ERROR / 0xC0000124	Es wurde ein Parameterfehler festgestellt.
INVALID_NETWORK_ADDRESS / 0xC0000125	Die Netzwerkadresse ist nicht korrekt.
SECURITY_MEMORY / 0xC0000126	Der Sicherheitsspeicher ist nicht verfügbar.

Es wurden Netzwerkfehler festgestellt:

Wert	Bedeutung
COMM_NETWORK_FAULT / 0xC0000126	Es wurde ein Netzwerkkommunikationsfehler festgestellt.
COMM_CONNECTION_CLOSED / 0xC0000141	Die Kommunikationsverbindung wurde geschlossen.
COMM_CONNECTION_TIMEOUT / 0xC0000142	Ein Timeout der Kommunikationsverbindung wurde festgestellt.
COMM_DUPLICATE_NODE / 0xC0000144	Ein doppelt vorhandener Knoten wurde festgestellt.

---

Wert	Bedeutung
COMM_CABLE_DISCONNECT / 0xC0000145	Ein getrenntes Kabel wurde festgestellt.
PROFIBUS_CONNECTION_TIMEOUT / 0xC009002E	Ein Timeout der PROFIBUS-Verbindung wurde festgestellt.

## Azyklischer Datenaustausch

### Registrierung für Callback

Im Fall einer nicht-zyklischen Prüfung kann ein Callback registriert werden. Wenn der PROFIBUS DPV1-Slave-Antrieb eine nicht-zyklische Prüfung des Masters empfängt, wird die Prüfung zuerst an registrierte Datenbereiche weitergeleitet.

Dann werden alle registrierten Funktionsbaustein-Instanzen (FB) aufgerufen, die über die Schnittstellen `IF_AsyncRead` und `IF_AsyncWrite` für den PROFIBUS DPV1-Slave registriert sind. Gleichzeitig wird der Zustand der automatischen Prüfungsverarbeitung übertragen, sodass auf mögliche Probleme reagiert werden kann. Der aktuelle Zustand wird dann in den Parameter `iq_stError` geschrieben.

Um im Fall einer eingehenden Leseanforderung die `AsyncRead`-Methode aufzurufen, muss für den PROFIBUS DPV1-Slave eine Instanz desselben Typs wie des Funktionsbausteins registriert werden, der die `IF_AsyncRead`-Schnittstelle implementiert.

Falls die `AsyncRead`-Methode dieser Funktionsbausteininstanz nicht mehr aufgerufen wird, wenn eine nicht-zyklische Leseanforderung empfangen wird, muss die Registrierung der Instanz entfernt werden.

**HINWEIS:** Verwenden Sie die Methode `IsRegisteredAsyncRead`, um zu überprüfen, ob eine bestimmte Funktionsbausteininstanz bereits als Callback für den PROFIBUS DPV1-Slave registriert ist.

Im Folgenden finden Sie eine Liste aller relevanten Methoden.

**HINWEIS:** Die `IoDrvPROFIBUSDPV1Slave`-Bibliothek wird dem Bibliotheksverwalter hinzugefügt, wenn Sie das Modul hinzufügen. Diese Bibliothek enthält die Schnittstellen, Methoden und Funktionsbausteine für die Verwaltung des PROFIBUS DP-Moduls.

### Relevante Methoden für die Callback-Registrierung

In der folgenden Tabelle werden die relevanten Methoden für die Registrierung der PROFIBUS DP-Schnittstelle beschrieben:

Methoden	Beschreibung
<code>IsRegisteredAsyncAlarmAck</code>	Gibt an, ob der übertragene Funktionsbaustein bei Empfang einer Alarmbestätigung bereits für das Callback registriert ist.
<code>IsRegisteredAsyncRead</code>	Gibt an, ob der übertragene Funktionsbaustein bei Empfang einer nicht-zyklischen Leseprüfung bereits für das Callback registriert ist.
<code>IsRegisteredAsyncWrite</code>	Gibt an, ob der übertragene Funktionsbaustein bei Empfang einer nicht-zyklischen Schreibprüfung bereits für das Callback registriert ist.
<code>RegisterAsyncAlarmAck</code>	Registriert die übertragene Funktionsbausteininstanz bei Empfang einer Alarmbestätigung für das Callback für den PROFIBUS-Slave.

Methoden	Beschreibung
<code>RegisterAsyncRead</code>	Registriert die übertragene Funktionsbausteininstanz bei Empfang einer nicht-zyklischen Leseprüfung für das Callback für den PROFIBUS-Slave.
<code>RegisterAsyncWrite</code>	Registriert die übertragene Funktionsbausteininstanz bei Empfang einer nicht-zyklischen Schreibprüfung für das Callback für den PROFIBUS-Slave.
<code>UnregisterAsyncAlarmAck</code>	Deregistriert die übertragene Funktionsbausteininstanz bei Empfang einer Alarmbestätigung für das Callback für den PROFIBUS-Slave.
<code>UnregisterAsyncRead</code>	Deregistriert die übertragene Funktionsbausteininstanz bei Empfang einer nicht-zyklischen Leseprüfung für das Callback für den PROFIBUS-Slave.
<code>UnregisterAsyncWrite</code>	Deregistriert die übertragene Funktionsbausteininstanz bei Empfang einer nicht-zyklischen Schreibprüfung für das Callback für den PROFIBUS-Slave.

### Registrieren nicht-zyklischer Datenbereiche

Datenbereiche können im PROFIBUS DPV1-Slave registriert werden. Nicht-zyklische Prüfungen (`AsyncRead` und `AsyncWrite`) für registrierte Datenbereiche werden dann automatisch vom PROFIBUS DPV1-Slave vorverarbeitet. Es sind ein Steckplatz und ein Index erforderlich, um nicht-zyklische Datenbereiche adressieren zu können.

Zum Erstellen eines nicht-zyklischen Moduls wird die Struktur `ST_PROFIBUSDV1AsyncData`-Module verwendet. In diese Struktur müssen ein Steckplatz und ein Index sowie ein Zeiger auf einen Datenbereich und die Länge der Daten an dieser Adresse eingegeben werden.

Stellen Sie sicher, dass der Zeiger `pbData` auf einen Speicherbereich zeigt, der während des Betriebs tatsächlich vorhanden ist und nicht gelöscht wird. Deklarieren Sie dafür z. B. das Array als Variable im Programm. Die Datenmodulstrukturen müssen nicht permanent verfügbar sein, da der Inhalt der Struktur beim Registrieren des Datenmoduls kopiert werden.

Im Folgenden finden Sie eine Liste aller relevanten Methoden.

## Relevante Methoden für die Datenbereichsregistrierung

In der folgenden Tabelle werden die relevanten Methoden für die Registrierung der Datenbereiche der IF\_PROFIBUS\_DPv1\_Slave-Schnittstelle beschrieben:

Methode	Beschreibung
IsRegisteredDataModule	<p>Überprüft, ob das übertragene Datenmodul registriert ist. Die Ausgabewerte lauten wie folgt:</p> <ul style="list-style-type: none"> <li>● 0: Kein Datenmodul ist auf einem bestimmten Steckplatz und Index registriert.</li> <li>● 1: Ein Datenmodul ist auf einem bestimmten Steckplatz und Index registriert.</li> <li>● 2: Der Steckplatz ist ungültig.</li> <li>● 3: Der Index ist ungültig.</li> </ul>
RegisterAsyncDataModule	<p>Registriert das übertragene Datenmodul mit einem Steckplatz und Index. Die Ausgabewerte lauten wie folgt:</p> <ul style="list-style-type: none"> <li>● 0: Das Datenmodul ist registriert.</li> <li>● 1: Der Steckplatz ist ungültig.</li> <li>● 2: Der Index ist ungültig.</li> <li>● 3: Der Zeiger auf den Datenbereich ist 0.</li> <li>● 4: Die Länge ist ungültig.</li> <li>● 5: Das Datenmodul ist bereits vorhanden.</li> </ul>
UnregisterAllAsyncDataModules	<p>Löscht die Registrierung aller registrierten Datenmodule.</p>
UnregisterAsyncDataModule	<p>Gibt an, warum das übertragene Datenmodul nicht registriert ist. Die Ausgabewerte lauten wie folgt:</p> <ul style="list-style-type: none"> <li>● 0: Das Datenmodul wurde deregistriert.</li> <li>● 1: Der Steckplatz ist ungültig.</li> <li>● 2: Der Index ist ungültig.</li> <li>● 3: Der Zeiger auf den Datenbereich ist 0.</li> <li>● 4: Das Datenmodul ist nicht registriert.</li> </ul>
GetAsyncDataModule	<p>Die Daten eines definierten registrierten Datenmoduls werden gelesen und ausgegeben. Die Ausgabewerte lauten wie folgt:</p> <ul style="list-style-type: none"> <li>● 0: Das Datenmodul wurde erfolgreich ausgelesen.</li> <li>● 1: Der Steckplatz ist ungültig.</li> <li>● 2: Der Index ist ungültig.</li> <li>● 3: Kein Datenmodul ist für diesen Steckplatz und Index registriert.</li> </ul>

---

# Abschnitt 10.2

## Konfiguration des EtherNet/IP-Adapters

---

### Einführung

Dieser Abschnitt beschreibt die Konfiguration des EtherNet/IP-Adapterdienstes des VW3E704100000-Kommunikationsmoduls.

### Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Konfiguration des EtherNet/IP-Adapters	124
Zyklischer Datenaustausch	128
Azyklischer Datenaustausch	129

## Konfiguration des EtherNet/IP-Adapters

### Einführung

In diesem Abschnitt wird die Konfiguration des EtherNet/IP-Adapterdienstes beschrieben.

Der EtherNet/IP-Adapter unterstützt die folgenden Austauschdienste:

- Zyklischer Datenaustausch (*siehe Seite 128*)
- Azyklischer Datenaustausch (*siehe Seite 129*)

Weitere Informationen zu EtherNet/IP (CIP), finden Sie auf der Website [www.odva.org](http://www.odva.org).

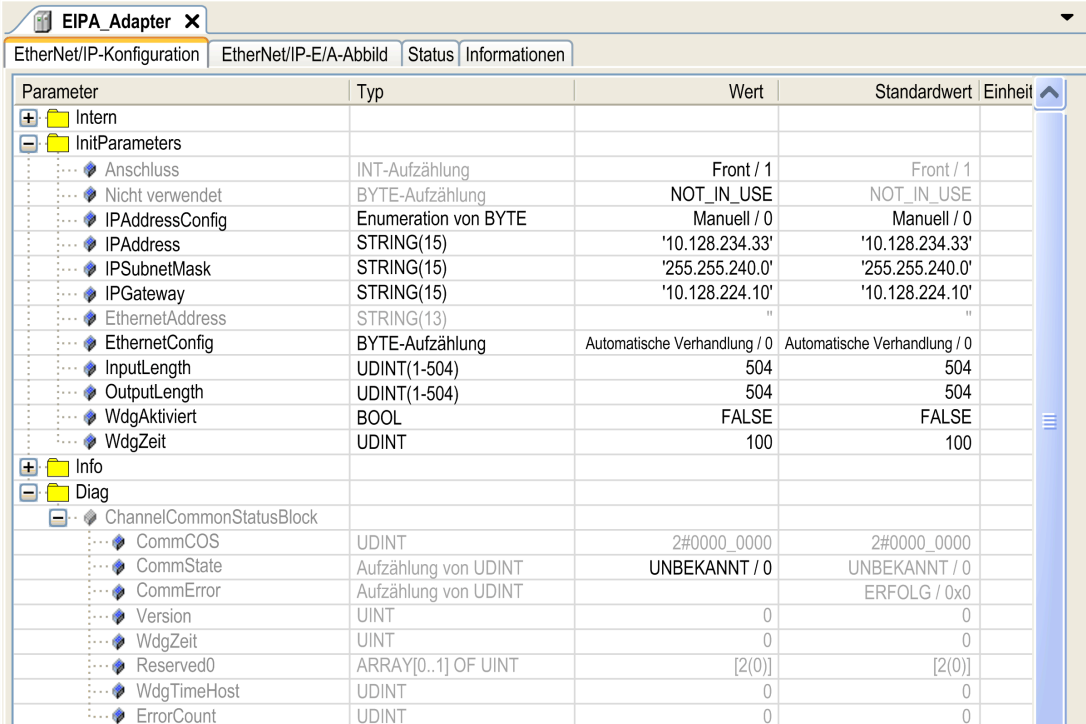
### Hinzufügen des EtherNet/IP -Adapters

Wenn Sie Ihrer Steuerung den EtherNet/IP-Adapter hinzufügen möchten, wählen Sie **EtherNet-IP-Adapter** im **Hardwarekatalog** aus, ziehen Sie ihn in die **Gerätebaumstruktur**, und legen Sie es auf dem Steuerungsknoten ab.

**HINWEIS:** Die Knoten **InputArea** und **OutputArea** werden dem EtherNet/IP-Adapterknoten hinzugefügt. Diese zwei E/A-Module werden für die zyklische Kommunikation verwendet.

## Konfiguration eines Ethernet/IP-Adapters

Um auf die Parameter für den EtherNet/IP-Adapter zuzugreifen, doppelklicken Sie auf **EtherNet-IP-Adapter** in der **Gerätebaumstruktur**:



The screenshot shows a software interface for configuring an EIPA\_Adapter. The main window is titled 'EIPA\_Adapter' and has several tabs: 'EtherNet/IP-Konfiguration', 'EtherNet/IP-E/A-Abbild', 'Status', and 'Informationen'. The 'EtherNet/IP-Konfiguration' tab is active, displaying a table of parameters. The table has columns for 'Parameter', 'Typ', 'Wert', 'Standardwert', and 'Einheit'. The parameters are organized into folders: 'Intern', 'InitParameters', 'Info', and 'Diag'. The 'InitParameters' folder is expanded, showing various configuration options like 'Anschluss', 'IPAddressConfig', 'IPAddress', etc. The 'Diag' folder is also expanded, showing diagnostic parameters like 'CommCOS', 'CommState', etc. The 'Wert' column contains the current values, and the 'Standardwert' column contains the default values.

Parameter	Typ	Wert	Standardwert	Einheit
+	Intern			
-	InitParameters			
...	Anschluss	Front / 1	Front / 1	
...	Nicht verwendet	NOT_IN_USE	NOT_IN_USE	
...	IPAddressConfig	Manuell / 0	Manuell / 0	
...	IPAddress	'10.128.234.33'	'10.128.234.33'	
...	IPSubnetMask	'255.255.240.0'	'255.255.240.0'	
...	IPGateway	'10.128.224.10'	'10.128.224.10'	
...	EthernetAddress	"	"	
...	EthernetConfig	Automatische Verhandlung / 0	Automatische Verhandlung / 0	
...	InputLength	504	504	
...	OutputLength	504	504	
...	WdgAktiviert	FALSE	FALSE	
...	WdgZeit	100	100	
+	Info			
-	Diag			
-	ChannelCommonStatusBlock			
...	CommCOS	2#0000_0000	2#0000_0000	
...	CommState	UNBEKANNT / 0	UNBEKANNT / 0	
...	CommError		ERFOLG / 0x0	
...	Version	0	0	
...	WdgZeit	0	0	
...	Reserved0	[2(0)]	[2(0)]	
...	WdgTimeHost	0	0	
...	ErrorCount	0	0	

**Parameter in Schwarz** Schreib-/Lesezugriff

**Parameter in Grau** Schreibgeschützt

Die folgenden Parameter werden auf der Registerkarte **EtherNet/IP-Konfiguration** angezeigt:

Parameter	Typ	Wert	Standardwert	Beschreibung
<b>Internal</b>				
Reserviert				
<b>InitParameters</b>				
IPAddressConfig	BYTE Enum	manual / 0 enable BOOTP / 1 enable DHCP / 2	manual / 0	Definiert, wie die IP-Adresse des Moduls festgelegt wird: <ul style="list-style-type: none"> <li>● 0 = Es wird eine manuell eingegebene IP-Adresse verwendet.</li> <li>● 1 = Die IP-Adresse wird vom Bootstrap-Protokoll festgelegt.</li> <li>● 2 = Die IP-Adresse wird vom DHCP-Protokoll festgelegt.</li> </ul>
IPAddress	STRING	'10.128.234.33'	'10.128.234.33'	Legt die IP-Adresse des EtherNet/IP-Adapters fest.
IPSubnetMask	STRING	'255.255.240.0'	'255.255.240.0'	Legt die Subnetzmaske des EtherNet/IP-Adapters fest.
IPGateway	STRING	'10.128.224.10'	'10.128.224.10'	Legt die Gateway-Adresse des EtherNet/IP-Adapters fest.
EthernetAddress	STRING	"	"	Zeigt die MAC-Adresse des EtherNet/IP-Adapters an.
EthernetConfig	BYTE Enum	Auto-negotiation / 0 Full Duplex / 100 Mbit/s / 1 Full Duplex / 10 Mbit/s / 2 Half Duplex / 100 Mbit/s / 3 Half Duplex / 10 Mbit/s / 4	Auto-negotiation / 0	Zeigt die Ethernet-Konfiguration des EtherNet/IP-Adapters an. <ul style="list-style-type: none"> <li>● 0 = Wenn dieser Wert festgelegt ist, verhandelt das Gerät die Verbindungsparameter unabhängig mit dem Remote-Hub oder Switch.</li> <li>● 1 = Das Gerät arbeitet mit 100 MBit/s und in Vollduplex.</li> <li>● 2 = Das Gerät arbeitet mit 10 MBit/s und in Vollduplex.</li> <li>● 3 = Das Gerät arbeitet mit 100 MBit/s und in Halbduplex.</li> <li>● 4 = Das Gerät arbeitet mit 10 MBit/s und in Halbduplex.</li> </ul>
InputLength	UDINT	1 bis 504	504	Definiert die Größe der Eingangsdaten der zyklischen Datenübertragung in Byte ( <b>InputArea</b> -Modul in der <b>Gerätebaumstruktur</b> ).
OutputLength	UDINT	1 bis 504	504	Definiert die Größe der Ausgangsdaten der zyklischen Datenübertragung in Byte ( <b>OutputArea</b> -Modul in der <b>Gerätebaumstruktur</b> ).

Parameter	Typ	Wert	Standardwert	Beschreibung
WdgEnabled	BOOL	TRUE FALSE	FALSE	Aktiviert den Watchdog: <ul style="list-style-type: none"> <li>● TRUE = Der Watchdog ist aktiviert. Der Wert des Parameters <code>WdgTime</code> wird als Watchdog-Timeout verwendet.</li> <li>● FALSE = Der Watchdog ist deaktiviert. Der Wert des Parameters <code>WdgTime</code> wird ignoriert.</li> </ul>
WdgTime	UDINT	20 bis 65535	100	Legt fest, nach welcher Zeit ein Slave vom Watchdog als nicht kommunizierend erkannt wird. Die Zeit wird in ms angegeben.
<b>Info</b>				
IdentNumber	WORD	0	0	Zeigt die Identifizierungsnummer des EtherNet/IP-Adapters an.
VendorName	STRING	'Schneider Electric'	'Schneider Electric'	Zeigt den Herstellernamen des EtherNet/IP-Adapters an.
ModelName	STRING	'EtherNet/IP-Adapter'	'EtherNet/IP-Adapter'	Zeigt den Modellnamen des EtherNet/IP-Adapters an.
DriverInstance	DWORD	0	0	Zeigt die Kennung für diese Antriebsinstanz an.
<b>Diag / ChannelCommonStatusBlock</b>				
CommState	UDINT Enum	UNKNOWN / 0 NOT_CONFIGURED / 1 STOP / 2 IDLE / 3 OPERATE / 4	UNKNOWN / 0	Zeigt den aktuellen Netzwerkstatus des Kommunikationskanals an. <ul style="list-style-type: none"> <li>● 0 = Nicht bestimmbar.</li> <li>● 1 = Nicht konfiguriert.</li> <li>● 2 = Angehalten.</li> <li>● 3 = Leerlauf.</li> <li>● 4 = In Betrieb.</li> </ul>
CommError	UDINT Enum	Diagnosecodes (siehe Seite 118)	SUCCESS / 0x0	Zeigt den aktuellen Diagnosecode des Kommunikationskanals an.
ErrorCount	UDINT	-	0	Zeigt die Gesamtfehleranzahl an, die seit dem letzten Start oder Neustart entdeckt wurden.

## Zyklischer Datenaustausch

### Instanz-ID der Eingangs-/Ausgangsbereiche

Die Ein- und Ausgangsbereiche haben die folgende Instanz-ID:

Element	Instanz-ID	Größe (Byte)	Beschreibung
Eingangs-Assembly ( <b>InputArea</b> )	101	0 bis 504	Befehlsword des Master-Steuerungsausgangs (%QW)
Ausgangs-Assembly ( <b>OutputArea</b> )	100	0 bis 504	Zustand der Steuerungseingänge (%IW)

Die Menge der ausgetauschten Daten ist von den Parametern `InputLength` und `OutputLength` abhängig, die in der Ethernet/IP-Konfiguration (*siehe Seite 125*) konfiguriert sind.

**HINWEIS:** Ausgang bedeutet AUSGANG der Master-Steuerung (= %IW für das Modul).  
Eingang bedeutet EINGANG der Master-Steuerung (= %IW für das Modul).

### E/A-Module des EtherNet/IP-Adapters

Die Knoten **InputArea** und **OutputArea** werden dem EtherNet/IP-Adapterknoten hinzugefügt, wenn Sie das Modul hinzufügen. Diese zwei E/A-Module werden für die zyklische Kommunikation verwendet.

Die Datenlänge der zwei eingefügten E/A-Module entsprechen der maximalen Datenlänge für die zyklische Kommunikation (504 Byte).

Wenn z. B. `InputLength` auf 50 und `OutputLength` auf 20 gesetzt ist, werden die ersten 50 Byte im E/A-Modul **InputArea** und die ersten 20 Byte im E/A-Modul **OutputArea** zyklisch mit dem Scanner ausgetauscht.

Die verbleibenden 454 (504-50) Eingangsbyte und 484 (504-20) Ausgangsbyte in den E/A-Modulen werden nicht verwendet.

### EtherNet/IP-Module – E/A-Abbild

Doppelklicken Sie auf den Knoten **InputArea** oder **OutputArea** in der **Gerätebaumstruktur**.

Auf der Registerkarte **EthernetIP-Module – E/A-Abbild** können Variablen definiert und benannt werden. Weiterhin sind auf dieser Registerkarte zusätzliche Informationen wie die topologische Adressierung enthalten.

Weitere allgemeine Beschreibungen finden Sie unter Beschreibung der Registerkarte "E/A-Abbild" (*siehe SoMachine, Programmierhandbuch*).

## Azyklischer Datenaustausch

### Registrierung für Callback

Mit dem EtherNet/IP-Adapter kann im Fall einer nicht-zyklischen Prüfung ein Callback registriert werden. Wenn der EtherNet/IP-Adapterantrieb eine nicht-zyklische Prüfung des Scanners empfängt, wird die Prüfung zuerst an registrierte Datenbereiche weitergeleitet.

Dann werden alle registrierten Funktionsbaustein-Instanzen (FB) aufgerufen, die für den EtherNet/IP-Adapter registriert sind. Dafür werden die Schnittstellen `IF_EIPEventHandler_AsyncGetAttributeAll`, `IF_EIPEventHandler_AsyncGetAttributeSingle` und `IF_EIPEventHandler_AsyncSetAttributeSingle` verwendet. Gleichzeitig wird der Zustand der automatischen Prüfungsverarbeitung übertragen, sodass auf mögliche Probleme reagiert werden kann. Der aktuelle Zustand wird dann in den Parameter `iq_udiError` geschrieben.

**HINWEIS:** Die Methode `RegisterAsyncClass()` muss aufgerufen werden, damit die Callbacks einer bestimmten Klasse in der Callback-Methode erfasst werden.

**HINWEIS:** Wenn nicht-zyklische Datenbereiche registriert sind, wird diese Methode intern automatisch ausgeführt. In diesem Fall müssen Sie die Methode `RegisterAsyncClass()` nicht manuell aufrufen.

Im Folgenden finden Sie eine Liste aller relevanten Methoden.

**HINWEIS:** Die `IoDrvEtherNetIPAdapter`-Bibliothek wird dem Bibliotheksverwalter hinzugefügt, wenn Sie das Modul hinzufügen. Diese Bibliothek enthält die Schnittstellen, Methoden und Funktionsbausteine für die Verwaltung des EtherNet/IP-Moduls.

### Relevante Methoden für die Callback-Registrierung

In der folgenden Tabelle werden die relevanten Methoden für die Registrierung der Datenbereiche der `IF_EtherNetIP_Adapter`-Schnittstelle beschrieben:

Methode	Beschreibung
<code>IsRegisteredAsyncGetAttributeAll</code>	Gibt an, ob der übertragene Funktionsbaustein bei Empfang einer <code>GetAttributeAll</code> -Prüfung) bereits für das Callback registriert ist.
<code>IsRegisteredAsyncGetAttributeSingle</code>	Gibt an, ob der übertragene Funktionsbaustein bei Empfang einer <code>GetAttributeSingle</code> -Prüfung) bereits für das Callback registriert ist.
<code>IsRegisteredAsyncSetAttributeSingle</code>	Gibt an, ob der übertragene Funktionsbaustein bei Empfang einer <code>SetAttributeSingle</code> -Prüfung) bereits für das Callback registriert ist.
<code>RegisterAsyncGetAttributeAll</code>	Registriert den übertragenen Funktionsbaustein für das Callback im Fall von <code>GetAttributeAll</code> -Prüfungen.

Methode	Beschreibung
RegisterAsyncGetAttributeSingle	Registriert den übertragenen Funktionsbaustein für das Callback im Fall von GetAttributeSingle-Prüfungen.
RegisterAsyncSetAttributeSingle	Registriert den übertragenen Funktionsbaustein für das Callback im Fall von SetAttributeSingle-Prüfungen.
UnregisterAsyncGetAttributeAll	Deregistriert den übertragenen Funktionsbaustein für das Callback im Fall von GetAttributeAll-Prüfungen.
UnregisterAsyncGetAttributeSingle	Deregistriert den übertragenen Funktionsbaustein für das Callback im Fall von GetAttributeSingle-Prüfungen.
UnregisterAsyncSetAttributeSingle	Deregistriert den übertragenen Funktionsbaustein für das Callback im Fall von SetAttributeSingle-Prüfungen.
RegisterAsyncClass	Aktiviert die Callbacks für die übertragene ClassId. Nur wenn die Callbacks für eine bestimmte ClassId aktiviert wurden, wird der für die Callbacks registrierte Code aufgerufen. Die Ausgabewerte lauten wie folgt: <ul style="list-style-type: none"> <li>● 0: Die ClassId wurde erfolgreich registriert.</li> <li>● 1: Die ClassId wurde bereits registriert.</li> <li>● : Die ClassId ist ungültig.</li> <li>● 3: Die Prüfungsanforderung RegisterAsyncClass kann nicht gesendet werden.</li> <li>● 4: Die Prüfungsanforderung RegisterAsyncClass war nicht korrekt.</li> </ul>

### Registrieren nicht-zyklischer Datenbereiche

Datenbereiche für nicht-zyklische Prüfungen können über den EtherNet/IP-Adapter registriert werden. Nicht-zyklische Prüfungen (Get\_Attribute\_All, Get\_Attribute\_Single und Set\_Attribute\_Single) für registrierte Datenbereiche werden dann automatisch vom EtherNet/IP-Adapter vorverarbeitet. Ein nicht-zyklischer Datenbereich wird durch Verwendung einer ClassId, einer InstanceId und einer AttributeId adressiert.

Dafür muss ein Attribut vom Typ ST\_EtherNetIPAttribute erstellt werden. Dieses Attribut muss die folgenden Programminhalte aufweisen: eine ID, eine Länge und einen Zeiger auf einen Datenbereich (z. B. ein Array).

Stellen Sie sicher, dass der Bereich, auf den der Zeiger eines Attributs zeigt, während des Betriebs tatsächlich vorhanden ist und nicht gelöscht wird. Deklarieren Sie dafür z. B. das Array als Variable im Programm.

Bei Verwendung der Schnittstelle IF\_EtherNetIPInstance wird das Attribut dann mit einer Instanz des Funktionsbausteins FB\_EtherNetIPInstance registriert.

**HINWEIS:** Stellen Sie sicher, dass die Instanzen des Funktionsbausteins FB\_EtherNetIPInstance sowie die Attribute permanent verfügbar sind und nicht während des Betriebs versehentlich gelöscht werden (z. B. werden lokale Variablen einer Methode nach Ausführung der Methode gelöscht).

Um die Instanz über den EtherNet/IP-Adapter zu registrieren, muss die erstellte nicht-zyklische Instanz eine `ClassId` und eine `InstanceId` umfassen.

Diese können mit den Methoden `SetClassId` und `SetInstanceId` erstellt werden.

Die Instanzen werden unter Verwendung der `IF_EtherNetIP_Adapter`-Schnittstelle des Ethernet/IP-Adapters registriert.

Im Folgenden finden Sie eine Liste aller relevanten Methoden.

### Relevante Methoden für die Datenbereichsregistrierung

In der folgenden Tabelle werden die relevanten Methoden für die Registrierung der Datenbereiche der Schnittstellen `IF_EtherNetIP_Adapter` und `IF_EtherNetIPInstance` beschrieben:

Methoden	Beschreibung
<b>IF_EtherNetIP_Adapter-Schnittstelle</b>	
<code>RegisterInstance</code>	Registriert die Instanz über den EtherNet/IP-Adapter. Die Ausgabewerte lauten wie folgt: <ul style="list-style-type: none"> <li>● 0: Die Instanz wird registriert.</li> <li>● 1: Die Instanz ist bereits registriert.</li> <li>● 2: Die Instanz ist ungültig.</li> <li>● 3: Interner Fehler.</li> </ul>
<code>UnregisterAllInstances</code>	Deregistriert alle Instanzen, die über den EtherNet/IP-Adapter registriert wurden. Die Ausgabewerte lauten wie folgt: <ul style="list-style-type: none"> <li>● 0: Alle Registrierungen wurden entfernt.</li> <li>● 1: Interner Fehler.</li> <li>● 2: Es sind keine registrierten Instanzen vorhanden.</li> <li>● 3: Interner Fehler.</li> </ul>
<code>UnregisterInstance</code>	Deregistriert eine definierte Instanz im EtherNet/IP-Adapter. Die Ausgabewerte lauten wie folgt: <ul style="list-style-type: none"> <li>● 0: Die Registrierung wurde ungültig gemacht.</li> <li>● <code>CmpErrors.Errors.ERR_NO_OBJECT</code>: Die letzte Instanz wurde nicht gefunden.</li> <li>● <code>CmpErrors.Errors.ERR_INVALID_HANDLE</code>: Interner Fehler.</li> </ul>
<b>IF_EtherNetIPInterface-Schnittstelle</b>	
<code>SetClassId</code>	Weist eine <code>ClassId</code> zu einer Instanz des Funktionsbausteins <code>FB_EtherNetIPInstance</code> zu. Die Ausgabewerte lauten wie folgt: <ul style="list-style-type: none"> <li>● 0: Die <code>ClassId</code> ist gültig und festgelegt.</li> <li>● 1: Die <code>ClassId</code> liegt nicht zwischen <code>Gc_EtherNetIPClass_MinID</code> und <code>Gc_EtherNetIPClass_MaxID</code>.</li> </ul>

Methodenname	Beschreibung
SetInstanceId	Weist eine InstanceId zu einer Instanz des Funktionsbausteins FB_EtherNetIPInstance zu. Die Ausgabewerte lauten wie folgt: <ul style="list-style-type: none"> <li>● 0: Die InstanceId ist gültig und festgelegt.</li> <li>● 1: Die ClassId liegt nicht zwischen 0 und Gc_EtherNetIPClass_MaxID.</li> </ul>
AddAttribute	Fügt das übertragene Attribut der Instanz des Funktionsbausteins FB_EtherNetIPInstance hinzu. Die Ausgabewerte lauten wie folgt: <ul style="list-style-type: none"> <li>● 0: Das Attribut wurde hinzugefügt.</li> <li>● 1: Die AttributeId wird bereits verwendet.</li> <li>● 2: Das Attribut ist ungültig.</li> </ul>
AddAttributeArray	Teilt ein Array in mehrere Teile auf, die die Größe von <code>_byAttributeLength</code> haben, und fügt alle Teile als Attribute zur Instanz des Funktionsbausteins FB_EtherNetIPInstance hinzu. Die Attribut-IDs sind aufeinander folgend nummeriert. <code>o_byStartAttributeId</code> gibt die AttributeId des zuerst hinzugefügten Attributs aus. Die Länge des zuletzt hinzugefügten Attributs kann geringer sein als der Übertragungsparameter <code>i_byAttributeLength</code> . Die Ausgabewerte lauten wie folgt: <ul style="list-style-type: none"> <li>● 0: Alle Attribute wurden hinzugefügt, und es wurden keine Fehler erkannt.</li> <li>● 1: Nicht ausreichend Platz in der Instanz.</li> </ul>
RemoveAttribute	Entfernt ein Attribut mit einer definierten ID. Die Ausgabewerte lauten wie folgt: <ul style="list-style-type: none"> <li>● 0: Das Attribut wurde erfolgreich entfernt.</li> <li>● <code>CmpErrors.Errors.ERR_INVALIDID</code>: Das Attribut wurde nicht gefunden.</li> </ul>
RemoveAllAttributes	Entfernt alle Attribute in der Instanz und gibt die Anzahl der entfernten Attribute aus.
GetClassId	Gibt die ClassId aus.
GetInstanceId	Gibt die InstanceId aus.
GetAttribute	Gibt die Daten der übertragenen Attribut-ID im Ein-/Aus-Parameter <code>iq_stAttribute</code> aus. Ausgabewert <code>CmpErrors.Errors.ERR_INVALIDID</code> : Das Attribut wurde nicht gefunden.
GetAttributeCount	Gibt die Anzahl der Attribute aus.
GetInstanceFreeSize	Gibt die Größe des freien Speicherplatzes in der Instanz aus.
IsValid	Gibt an, ob die Instanzinformationen gültig sind. Dafür müssen ClassID und InstanceID festgelegt werden.

Methode	Beschreibung
GetFirstAttribute	Gibt die Daten des ersten Attributs der Instanz im Ein-/Aus-Parameter <code>iq_stAttribute</code> aus. Die Ausgabewerte lauten wie folgt: <ul style="list-style-type: none"><li>● 0: Es wurden Attribute gefunden.</li><li>● 1: Es wurden keine Attribute gefunden.</li></ul>
GetNextAttribute	Gibt die Daten des nächsten Attributs der Instanz im Ein-/Aus-Parameter <code>iq_stAttribute</code> aus. Die Ausgabewerte lauten wie folgt: <ul style="list-style-type: none"><li>● 0: Es wurden Attribute gefunden.</li><li>● 1: Es wurden keine Attribute gefunden.</li></ul>

## Abschnitt 10.3

### Konfiguration eines Ethernet/IP-Scanners

#### Einführung

Dieser Abschnitt beschreibt die Konfiguration des EtherNet/IP-Scannerdiensts des VW3E704100000-Ethernet-Kommunikationsmoduls.

#### Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Beschreibung	135
Unterstützte Geräte	136
Konfiguration des EtherNet/IP Scanners	138
EtherNet/IP Scanner-E/A-Abbild	140
Status und Diagnose des EtherNet/IP Scanners	141
Deklaration des Zielgeräts	143
Zieleinstellungen	145
Verbindungskonfiguration	147
Austausch von Geräten über Benutzerparameter	165
EtherNet/IP-E/A-Abbild	169

## Beschreibung

### EtherNet/IP im Überblick

EtherNet/IP ist der Name des CIP-Protokolls (Common Industrial Protocol), das über Standard-Ethernet implementiert wurde. Weitere Informationen zu CIP finden Sie auf der Website [www.odva.org](http://www.odva.org).

EtherNet/IP verwendet eine Urheber/Ziel-Architektur für den Datenaustausch:

- Urheber sind Geräte, die den Datenaustausch mit Zielgeräten im Netzwerk initiieren. Dies trifft sowohl auf E/A-Kommunikationen als auch auf Service-Nachrichten zu. Dies entspricht der Rolle des Clients in einem Modbus-Netzwerk.
- Zielgeräte sind Geräte, die auf Datenanfragen reagieren, die von Urhebergeräten ausgegeben werden. Dies trifft sowohl auf E/A-Kommunikationen als auch auf Service-Nachrichten zu. Dies entspricht der Rolle des Servers in einem Modbus-Netzwerk.

Die Kommunikation zwischen den EtherNet/IP-Urhebern und den Zielgeräten wird über EtherNet/IP-Verbindungen hergestellt.

### Funktionen des EtherNet/IP-Scanners

Der EtherNet/IP Scanner auf dem LMC078 Motion Controller ist ein Urhebergerät, das Verbindungen mit Zielgeräten herstellt und Konfigurationsinformationen, sowie Eingangs-/Ausgangsdaten mit diesen austauscht. Der Scanner gibt beim Zielgerät beispielsweise an, wie oft das Gerät seine Eingangsdaten übermitteln soll und wie oft es Ausgangsdaten vom Scanner erwartet.

Die folgende Tabelle enthält die Funktionen des EtherNet/IP Scanner-Dienstes für den LMC078 Motion Controller:

Merkmal	Beschreibung
Leistung	Bis zu 64 EtherNet/IP-Zielgeräte, die von der Steuerung verwaltet und in einem Intervall von 10 ms überwacht werden
Anzahl der Verbindungen	1 bis 64
Anzahl der Eingangswörter	0 bis 5712. Maximal 504 für jedes Zielgerät
Anzahl der Ausgangswörter	0 bis 5712. Maximal 504 für jedes Zielgerät
E/A-Kommunikation	EtherNet/IP-E/A-Scannerdienst Funktionsbaustein für Konfiguration und Datenübertragung Quellgerät/Zielgerät
Andere Dienste	EDS-Dateiverwaltung

## Unterstützte Geräte

### Unterstützte Geräte

Diese Tabelle enthält die Zielgeräte, die von EtherNet/IP Scanner unterstützt werden:

Gerätename		TVDA	Wichtige Merkmale
Vordefinierte Geräte	Altivar 32	X	Bibliotheken, vordefinierte Verbindungen, vordefinierter Datenaustausch
	Altivar 320	X	Bibliotheken, vordefinierte Verbindungen, vordefinierter Datenaustausch
	Altivar 340	X	Bibliotheken, vordefinierte Verbindungen, vordefinierter Datenaustausch
	Altivar 6••	X	Bibliotheken, vordefinierte Verbindungen, vordefinierter Datenaustausch
	Altivar 71	X	Bibliotheken, vordefinierte Verbindungen, vordefinierter Datenaustausch
	Altivar 9••	X	Bibliotheken, vordefinierte Verbindungen, vordefinierter Datenaustausch
	Lexium 32 M	X	Bibliotheken, vordefinierte Verbindungen, vordefinierter Datenaustausch
	Lexium ILA	X	Bibliotheken, vordefinierte Verbindungen, vordefinierter Datenaustausch
	Lexium ILE	X	Bibliotheken, vordefinierte Verbindungen, vordefinierter Datenaustausch
	Lexium ILS	X	Bibliotheken, vordefinierte Verbindungen, vordefinierter Datenaustausch
	OsiSense XG	X	Vordefinierte Verbindungen, vordefinierter Datenaustausch
	OsiSense XUW	X	Vordefinierte Verbindungen, vordefinierter Datenaustausch
	XPSMCM	X	Vordefinierte Verbindungen, vordefinierter Datenaustausch
Sonstige Geräte	Gerät mit EDS-Datei <sup>(1)</sup>	-	Benutzerparameter, vordefinierte Verbindungen
	Generisches Slave-Gerät <sup>(2)</sup>	-	Benutzerparameter
<p><b>(1)</b> Eine EDS-Datei stellt unter anderem die vordefinierten Verbindungen zur Verfügung, um die Integration des Netzwerks zu erleichtern.</p> <p><b>(2)</b> Ein generisches Slave-Gerät wird in SoMachine verwendet, um Ethernet/IP-Geräte wie Regelantriebe, Sensoren oder andere Steuerungen hinzuzufügen, die keine vordefinierten Verbindungen haben.</p>			

## Hauptmerkmale

In dieser Tabelle werden die wichtigsten Merkmale vorgestellt:

Wichtige Merkmale	Beschreibung
Bibliotheken	Dem Gerät zugewiesene Funktionen und Funktionsbausteine, die für die Verwendung der Anwendung verfügbar sind. Siehe Motion Control - Bibliothekshandbuch ( <i>siehe Seite 9</i> ).
Vordefinierte Verbindungen	Wird zum Einrichten zyklischer Datenaustauschvorgänge verwendet. Wählen Sie eine der vorgeschlagenen Verbindungen mit den relevanten Informationen aus. Weitere Informationen finden Sie unter Verbindungskonfiguration ( <i>siehe Seite 147</i> ).
Vordefinierter Datenaustausch	Der zyklische Datenaustausch wird automatisch eingerichtet: wenn Sie dem Projekt ein Gerät hinzufügen, wird eine vordefinierte Verbindung automatisch ausgewählt.
Benutzerparameter	Parameter, die beim Start automatisch an das Gerät gesendet werden. Diese Parameter werden verwendet, wenn Geräte ersetzt werden, die FDR nicht unterstützen.

## TVDA

Einige unterstützte Geräte werden mit Anwendungscode-Vorlagen (so genannten Gerätemodulen) bereitgestellt, die die Integration von Geräten wie drehzahlvariablen Antrieben oder Servoantrieben in ein SoMachine-Projekt ermöglichen. Die Gerätemodule basieren auf Funktionsvorlagen, einem Mechanismus in SoMachine, um vordefinierte Anwendungsprogramminhalte abzurufen.

Jedes Gerätemodul enthält den erforderlichen SoMachine-Anwendungsinhalt, um das Feldgerät zu steuern, dessen Status zu überwachen und erkannte Fehler zu verarbeiten. Es umfasst eine separate, globale Variablendefinition, die eine Schnittstelle für den Zugriff auf die Gerätefunktionen im SoMachine-Automatisierungsprojekt bietet.

Detaillierte Informationen hierzu finden Sie im Bibliothekshandbuch TVDA Device Module Library, Function Template Library Guide (*siehe TVDA Device Module Library, Function Template Library Guide*).

## Konfiguration des EtherNet/IP Scanners

### Hinzufügen des EtherNet/IP Scanners

Um der Steuerung einen EtherNet/IP-Scanner hinzuzufügen, wählen Sie das Element **EtherNet/IP Scanner** im **Hardwarekatalog** aus, ziehen Sie das Element in die **Gerätebaumstruktur** und legen Sie es auf dem Steuerungsknoten ab.

Weitere Informationen zum Hinzufügen von Geräten in einem Projekt finden Sie unter:

- Verwenden der Methode Drag&Dop (*siehe SoMachine, Programmierhandbuch*) (Ziehen und Ablegen)
- Verwenden der Kontextmenüs oder Plus-Schaltflächen (*siehe SoMachine, Programmierhandbuch*)

### Konfiguration des EtherNet/IP Scanners

Um den EtherNet/IP Scanner zu konfigurieren, doppelklicken Sie auf den Knoten **EtherNet IP Scanner** in der **Gerätebaumstruktur**:

The screenshot shows the configuration window for an EtherNet/IP Scanner. The window title is "EtherNet\_IP\_Scanner x". It has four tabs: "Scanner-Einstellungen" (selected), "EtherNet/IP-E/A-Abbild", "Status", and "Informationen".

Under "Adresseinstellungen", there are two radio buttons: "Statische IP-Adresse verwenden" (selected) and "IP-Adresse automatisch abrufen".

When "Statische IP-Adresse verwenden" is selected, the following fields are visible:

- IP-Adresse: 192 . 168 . 0 . 1
- Subnetzmaske: 255 . 255 . 255 . 0
- Gateway-Adresse: 0 . 0 . 0 . 0

When "IP-Adresse automatisch abrufen" is selected, there are two sub-options: "BOOTP" and "DHCP", both with unselected radio buttons.

Under "Ethernet-Einstellungen", there is a "Speed & Duplex" dropdown menu set to "Auto-Negotiation".

Under "Optionen", there is a checkbox "Verbindungen automatisch erneut herstellen" which is currently unchecked.

Auf der Registerkarte **Scanner-Einstellungen** werden folgende Parameter bereitgestellt:

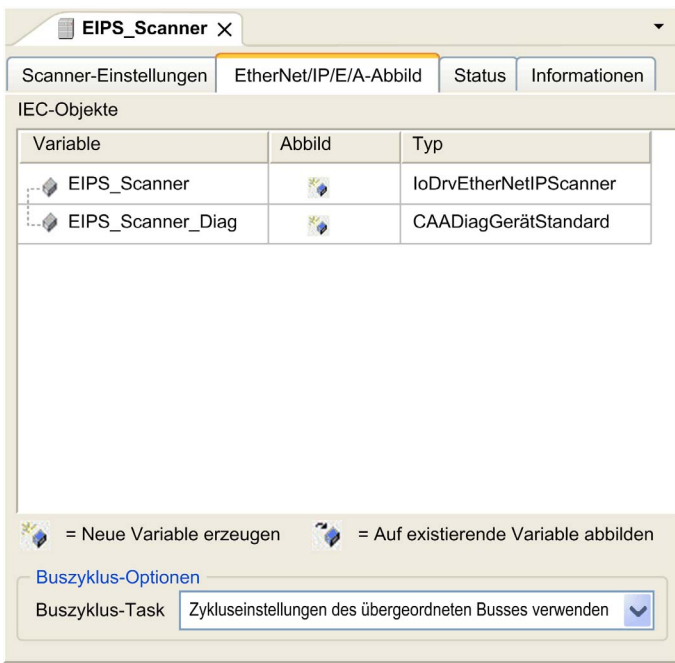
Parameter	Standardwert	Beschreibung
<b>IP-Adresse</b>	192.168.0.1	Legt die IP-Adresse des EtherNet/IP Scanners fest.
<b>Subnetzmaske</b>	255.255.255.0	Legt die Subnetzmaske des EtherNet/IP Scanners fest.
<b>Gateway-Adresse</b>	0.0.0.0	Legt die Gateway-Adresse des EtherNet/IP Scanners fest.
<b>Speed&amp;Duplex</b>	<b>Auto-Negotiation</b>	<p>Wählt die Datenübertragungsrichtung und -geschwindigkeit des EtherNet/IP Scanners aus:</p> <p><b>Auto-Negotiation.</b> Der Scanner verhandelt die Verbindungsparameter unabhängig mit dem Remote-Hub oder Switch.</p> <p><b>Vollduplex /100 Mbit/s.</b> Der Scanner arbeitet mit 100 MBit/s und in Vollduplex.</p> <p><b>Vollduplex /10 Mbit/s.</b> Der Scanner arbeitet mit 10 MBit/s und in Vollduplex.</p> <p><b>Halbduplex /100 Mbit/s.</b> Der Scanner arbeitet mit 100 MBit/s und in Halbduplex.</p> <p><b>Halbduplex /10 Mbit/s.</b> Der Scanner arbeitet mit 10 MBit/s und in Halbduplex.</p>

## EtherNet/IP Scanner-E/A-Abbild

### Konfigurieren des E/A-Abbilds für den EtherNet/IP Scanner

Die E/A-Buskonfiguration ermöglicht die Auswahl der Task für die Steuerung der zyklischen EtherNet/IP-Datenaustauschvorgänge.

Gehen Sie wie folgt vor, um das E/A-Abbild für den EtherNet/IP Scanner zu konfigurieren:

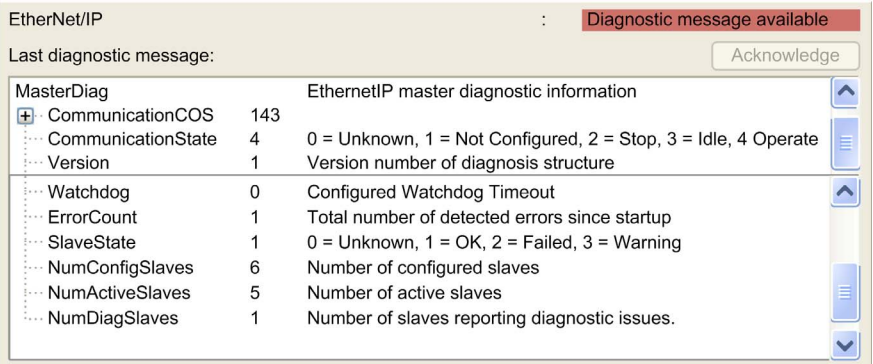
Schritt	Aktion
1	Doppelklicken Sie in der <b>Gerätebaumstruktur</b> auf <b>Ethernet-IP-Scanner</b> . <b>Ergebnis:</b> Das Konfigurationsfenster wird angezeigt.
2	<p>Wählen Sie die Registerkarte <b>EtherNet/IP-E/A-Abbild</b> aus.</p> 
3	<p>Der Parameter <b>Buszyklus-Task</b> definiert die Task zur Aktualisierung der E/A-Bilder (%QB und %IB). Diese E/A-Bilder entsprechen den EtherNet/IP-Requests, die an die Zielgeräte und die Funktionsfähigkeitsbits gesendet wurden.</p> <p>Wählen Sie die <b>Buszyklus-Task</b> aus, um Folgendes verwenden zu können:</p> <ul style="list-style-type: none"> <li>● <b>Zykluseinstellungen des übergeordneten Busses verwenden</b> (Standardeinstellung). Verwenden Sie die in der Registerkarte SPS-Einstellungen (<i>siehe Seite 91</i>) der Steuerung spezifizierte Task.</li> <li>● <b>MAST</b>. Verwenden Sie die MAST-Task (<i>siehe Seite 51</i>).</li> <li>● <b>Bewegung</b>. Verwenden Sie die Bewegungs-Task (<i>siehe Seite 44</i>).</li> </ul>

## Status und Diagnose des EtherNet/IP Scanners

### Einführung

Die Registerkarte **Status** des EtherNet/IP Scanners stellt im Online-Modus Überwachungs- und Diagnoseinformationen für den EtherNet/IP Scanner und die verbundenen Geräte zur Verfügung.

### Anzeige der Überwachungs- und Diagnoseinformationen

Schritt	Aktion																																	
1	Doppelklicken Sie in der <b>Gerätebaumstruktur</b> auf <b>Ethernet-IP-Scanner</b> .																																	
2	<p>Wählen Sie die Registerkarte <b>Status</b> aus.</p> <p><b>Ergebnis:</b> Das Fenster ‚Letzte Diagnosemeldung‘ wird angezeigt:</p>  <p>The screenshot shows a window titled 'EtherNet/IP : Diagnostic message available' with an 'Acknowledge' button. Below the title bar, it says 'Last diagnostic message:'. The main content is a table with the following data:</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>MasterDiag</td> <td></td> <td>EthernetIP master diagnostic information</td> </tr> <tr> <td>CommunicationCOS</td> <td>143</td> <td></td> </tr> <tr> <td>CommunicationState</td> <td>4</td> <td>0 = Unknown, 1 = Not Configured, 2 = Stop, 3 = Idle, 4 Operate</td> </tr> <tr> <td>Version</td> <td>1</td> <td>Version number of diagnosis structure</td> </tr> <tr> <td>Watchdog</td> <td>0</td> <td>Configured Watchdog Timeout</td> </tr> <tr> <td>ErrorCount</td> <td>1</td> <td>Total number of detected errors since startup</td> </tr> <tr> <td>SlaveState</td> <td>1</td> <td>0 = Unknown, 1 = OK, 2 = Failed, 3 = Warning</td> </tr> <tr> <td>NumConfigSlaves</td> <td>6</td> <td>Number of configured slaves</td> </tr> <tr> <td>NumActiveSlaves</td> <td>5</td> <td>Number of active slaves</td> </tr> <tr> <td>NumDiagSlaves</td> <td>1</td> <td>Number of slaves reporting diagnostic issues.</td> </tr> </tbody> </table>	Parameter	Value	Description	MasterDiag		EthernetIP master diagnostic information	CommunicationCOS	143		CommunicationState	4	0 = Unknown, 1 = Not Configured, 2 = Stop, 3 = Idle, 4 Operate	Version	1	Version number of diagnosis structure	Watchdog	0	Configured Watchdog Timeout	ErrorCount	1	Total number of detected errors since startup	SlaveState	1	0 = Unknown, 1 = OK, 2 = Failed, 3 = Warning	NumConfigSlaves	6	Number of configured slaves	NumActiveSlaves	5	Number of active slaves	NumDiagSlaves	1	Number of slaves reporting diagnostic issues.
Parameter	Value	Description																																
MasterDiag		EthernetIP master diagnostic information																																
CommunicationCOS	143																																	
CommunicationState	4	0 = Unknown, 1 = Not Configured, 2 = Stop, 3 = Idle, 4 Operate																																
Version	1	Version number of diagnosis structure																																
Watchdog	0	Configured Watchdog Timeout																																
ErrorCount	1	Total number of detected errors since startup																																
SlaveState	1	0 = Unknown, 1 = OK, 2 = Failed, 3 = Warning																																
NumConfigSlaves	6	Number of configured slaves																																
NumActiveSlaves	5	Number of active slaves																																
NumDiagSlaves	1	Number of slaves reporting diagnostic issues.																																

### Diagnoseinformationen

Das Feld **CommunicationCOS** ist ein binär kodierter Dezimalwert:

Bit	Beschreibung, wenn das Bit auf 1 gesetzt ist
0	Kommunikation bereit
1	Konfiguration korrekt ausgeführt
2	Protokoll für Kommunikation offen
3	Konfiguration gesperrt
4	Neue Konfiguration
5	Neustart der Kommunikation angefordert
6	Neustart der Kommunikation aktiviert
7	Direkter Speicherzugriff aktiviert
8 bis 31	Nicht verwendet

Beispiel: Bei einem **CommunicationCOS**-Wert von 143 (Betriebszustand) werden die folgenden Bits auf 1 gesetzt:

- Bit 0 (Kommunikation bereit)
- Bit 1 (Konfiguration korrekt ausgeführt)
- Bit 2 (Protokoll für Kommunikation offen)
- Bit 3 (Konfiguration gesperrt)
- Bit 7 (Direkter Speicherzugriff aktiviert)

## Deklaration des Zielgeräts

### Überblick

Dieser Abschnitt beschreibt, wie ein Zielgerät auf dem EtherNet/IP Scanner hinzugefügt wird. Die verfügbaren Zielgeräte sind unter Unterstützte Geräte (*siehe Seite 136*) aufgelistet.

### Automatische Einstellungen

Bei jeder Zielgerät-Deklaration legt SoMachine automatisch Folgendes fest:

- Netzwerkeinstellungen (IP-Adresse, Subnetzmaske, Gateway-Adresse) gemäß den EtherNet/IP Scanner IP-Einstellungen (*siehe Seite 138*).
- Vordefinierte Datenaustauschvorgänge bei vordefinierten Geräten.

### Hinzufügen eines Zielgeräts

Um im Knoten **Ethernet-IP-Scanner** ein Zielgerät hinzuzufügen, wählen Sie das Gerät im **Hardwarekatalog** aus, ziehen es in die **Gerätebaumstruktur** und legen es auf dem Knoten **Ethernet-IP-Scanner** ab.

Weitere Informationen zum Hinzufügen von Geräten in einem Projekt finden Sie unter:

- Verwenden der Methode Drag&Dop (*siehe SoMachine, Programmierhandbuch*) (Ziehen und Ablegen)
- Verwenden der Kontextmenüs oder Plus-Schaltflächen (*siehe SoMachine, Programmierhandbuch*)

Einige Bibliotheken können, abhängig vom Zielgerät, das Sie dem Projekt hinzufügen, automatisch geladen werden. Siehe das Motion Control - Bibliothekshandbuch (*siehe Seite 9*) für die verfügbaren Funktionsbausteine.

### Hinzufügen eines Zielgeräts aus einer Vorlage

Bei Geräten, die keine wichtigen Merkmale haben, aber TVDA unterstützen, ist es möglich, für sie die Verwendung einer Vorlage zu deklarieren. Dadurch werden zusätzliche Elemente importiert, die die Programmierung erleichtern.

Verwenden Sie diese Methode für Geräte des Typs OsiSense XGCS, XUW und Preventa XPSMCM.

Gehen Sie wie folgt vor, um ein Gerät von einer Vorlage hinzuzufügen:

Schritt	Aktion
1	Aktivieren im <b>Hardwarekatalog</b> das Kontrollkästchen <b>Gerätevorlage</b> .
2	Wählen Sie das Gerät im <b>Hardwarekatalog</b> aus, ziehen Sie es in die <b>Gerätebaumstruktur</b> und legen Sie es auf dem Knoten <b>Ethernet-IP-Scanner</b> ab.

Weitere Informationen zum Hinzufügen von Geräten in einem Projekt finden Sie unter:

- Verwenden der Methode Drag&Dop (*siehe SoMachine, Programmierhandbuch*) (Ziehen und Ablegen)
- Verwenden der Kontextmenüs oder Plus-Schaltflächen (*siehe SoMachine, Programmierhandbuch*)

### Hinzufügen eines Geräts aus einer EDS-Datei

Einige Drittanbietergeräte werden mit einer EDS-Datei geliefert.

Gehen Sie wie folgt vor, um ein Gerät mit einer EDS-Datei hinzuzufügen:

Schritt	Aktion
1	Wählen Sie im Menü SoMachine die Option <b>Tools</b> → <b>Geräte-Repository</b> aus.
2	Klicken Sie auf <b>Installieren</b> , um das Dialogfeld <b>Gerätebeschreibungsdateien</b> zu öffnen.
3	Wählen Sie in der Dateitypliste die Option <b>EDS- und DCF-Dateien</b> aus.
4	Wählen Sie die EDS-Datei aus.
5	Klicken Sie auf <b>OK</b> , um das Dialogfeld zu schließen.
6	Klicken Sie auf <b>Schließen</b> , um das Dialogfeld <b>Installiere Gerätebeschreibung</b> zu schließen.
7	Wählen Sie <b>Ethernet-IP-Scanner</b> aus und klicken Sie auf die Plus-Schaltfläche. Wählen Sie das neu hinzugefügte Zielgerät aus und klicken Sie auf <b>Gerät hinzufügen</b> . Weitere Informationen finden Sie unter Verwendung des Kontextmenüs oder der Plus-Schaltfläche ( <i>siehe SoMachine, Programmierhandbuch</i> ).

## Zieleinstellungen

### Überblick

Nachdem ein Zielgerät im EtherNet/IP Scanner hinzugefügt wurde, können Sie die Registerkarte **Zielsystemeinstellungen** des Geräts verwenden, um die Netzwerkeinstellungen des Geräts zu bearbeiten.

### Einstellungen des Zielgeräts

Doppelklicken Sie in der **Gerätebaumstruktur** auf den Knoten eines EtherNet/IP-Zielgeräts:

Altivar\_71 X

Überblick -Zieleinstellungen Verbindungen Benutzerparameter EtherNet/IP/E/A-Abbild Status Informationen

Adresseinstellungen

IP-Adresse: 192 . 168 . 0 . 2

BOOTP 00 : 00 : 00 : 00 : 00 : 00

IP-Adresse speichern

Elektronisches Keying

Gerätetyp prüfen 2

Herstellercode prüfen 243

Produktcode prüfen 5

Hauptrevision prüfen 1

Nebenrevision prüfen 1

Standardwerte wiederherstellen

Protokoll auf dem Feldbus

Vom Gerät verwendetes Protokoll: EtherNet/IP

Das Protokoll, das zwischen Steuerung und Gerät eingesetzt wird.

### Adresseinstellungen

Zielgeräte, die dem EtherNet/IP Scanner hinzugefügt werden, müssen eine feste IP-Adresse haben.

Geben Sie die IP-Adresse des Geräts im Feld **IP-Adresse** ein.

Beim Ersatz eines Geräts:

Schritt	Aktion
1	Installieren Sie das neue Gerät.
2	Konfigurieren Sie die Netzwerkeinstellungen im Gerät (IP-Adresse, Subnetzmaske und Gateway-Adresse).
3	Konfigurieren Sie die Parameter direkt im Gerät oder mithilfe von SoMachine.
4	Starten Sie das Gerät und die Anwendung.

### Elektronisches Keying

Das Gerät wird über Signaturen mit **elektronischem Keying** identifiziert.

Bei **elektronischem Keying** handelt es sich um Informationen, die in der Firmware des Geräts enthalten sind (Herstellercode, Produktcode usw.)

Beim Start des Scanners wird jeder ausgewählte elektronische Keying-Wert mit den entsprechenden Informationen im Gerät verglichen.

Sollten die Gerätewerte nicht mit den Anwendungswerten übereinstimmen, dann bricht der Logic Controller die Kommunikation mit dem Gerät ab. Dies kann innerhalb Ihrer Applikation mithilfe der Diagnoseinformationen (*siehe Seite 141*) überwacht werden, um angemessene Maßnahmen im Kontext Ihrer Maschine zu ergreifen.

Bei vorkonfigurierten Geräten können die Werte für das **elektronische Keying** nicht geändert werden.

Bei generischen EtherNet/IP-Geräten können Sie die Werte für das **elektronische Keying** ändern.

Informationen zu den Werten für das **elektronische Keying** finden Sie in der Beschreibung des Identitätsobjekts (F1 hex) in der Dokumentation zum Gerät.

## Verbindungskonfiguration

### Verbindung – Überblick

Sie müssen die Verbindungen konfigurieren, um auf ein EtherNet/IP-Gerät zugreifen zu können. Eine Verbindung ermöglicht den Austausch von Datenbausteinen, die in Assemblies kombiniert sind.

Der Start und Stopp der Verbindungen wird von der Steuerung verwaltet.

### Assemblies

E/A-Daten und Konfigurationsdaten können in Assembly-Objekten kombiniert werden.

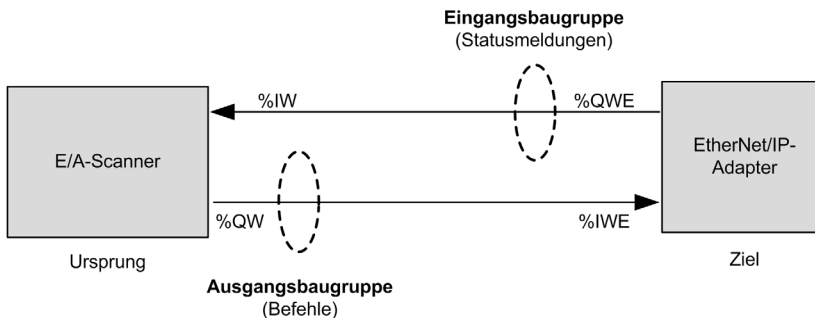
Daten (Attribute) von verschiedenen Objekten können in einem einzelnen Objekt kombiniert werden, damit Daten über eine einzelne Verbindung gesendet oder empfangen werden können.

Assembly-Objekt-Instanzen werden verwendet, um Daten für Ein- und Ausgangsdaten von E/A-Verbindungen zu aggregieren.

Assembly-Objekte sind in Klassen, Instanzen und Attribute strukturiert:

- Bei einer Klasse handelt es sich um eine Gruppe von Objekten, die dieselbe Art von Systemkomponente darstellen.
- Eine Objektinstanz entspricht einem bestimmten Objekt in einer Klasse. Jede Instanz verfügt über eine eigene Gruppe von Attributwerten.
- Attribute sind Eigenschaften eines Objekts und/oder einer Objektklasse. In der Regel stellen Attribute Statusinformationen bereit oder definieren die Funktionsweise eines Objekts.

Die folgende Abbildung zeigt die Funktionsrichtung der Eingangs- und der Ausgangs-Assembly bei einer EtherNet/IP-Kommunikation:



Die EtherNet/IP-Konfigurationsparameter sind wie folgt definiert:

- **Instanz:** Nummer der Assembly.
- **Größe:** Anzahl der Kanäle einer Assembly.  
Die Speichergröße jedes Kanals ist 2 Byte, in denen der Wert des %IB<sub>x</sub>- oder %QB<sub>x</sub>-Objekts gespeichert wird, wobei x für die Kanalnummer steht.

Wenn die **Größe** der **Ausgangs-Assembly** beispielsweise 20 ist, bedeutet dies, dass es 20 Eingangskanäle gibt (IB0 bis IB19), die %IB<sub>y</sub>...%IB<sub>(y+20-1)</sub> adressieren, wobei y der erste verfügbare Kanal für die Assembly ist.

### Konfigurieren der Geräteverbindungen

Zum Erstellen und Konfigurieren von Verbindungen klicken Sie doppelt auf das EtherNet/IP-Zielgerät in der **Gerätebaumstruktur** und wählen Sie die Registerkarte **Verbindungen** aus:



Spalte	Kommentar
<b>Verbindung Nr.</b>	Die Verbindungsnummer ist eindeutig. Sie wird automatisch von SoMachine vergeben.
<b>Verbindungsname</b>	Der standardmäßige Verbindungsname wird automatisch von SoMachine generiert. Bei vordefinierten Verbindungen ( <i>siehe Seite 136</i> ) kann dieser Name nicht bearbeitet werden. Bei allen anderen Verbindungen kann der standardmäßige Verbindungsname im Fenster zur Verbindungsbearbeitung ( <i>siehe Seite 160</i> ) bearbeitet werden.
<b>RPI O → T (ms)</b>	Intervall für angeforderte Pakete: Der Zeitraum zwischen den zyklischen Datenübertragungen, die von EtherNet/IP Scanner (O → T) oder vom Zielgerät (T → O) angefordert wurden.
<b>RPI T → O (ms)</b>	
<b>O → T Größe (Byte)</b>	Anzahl der zwischen dem Quellgerät (Originator, O) und dem Zielgerät (Target, T) zu übertragenden Byte.
<b>T → O Größe (Byte)</b>	
<b>Konfig#1 Größe (Byte)</b>	Größe des ersten Konfigurationsparametersatzes. Wird nur für Geräte mit konfigurierbaren Parametern angezeigt.

Spalte	Kommentar
Konfig#2 Größe (Byte)	Größe des zweiten Konfigurationsparametersatzes. Wird nur für Geräte mit konfigurierbaren Parametern angezeigt.
Verbindungspfad	Kodierte Transkription der anderen Verbindungsparameter.

Um eine Verbindung zu erstellen, klicken Sie auf **Verbindung hinzufügen**. Siehe im folgenden: „Hinzufügen einer EtherNet/IP-Verbindung“.

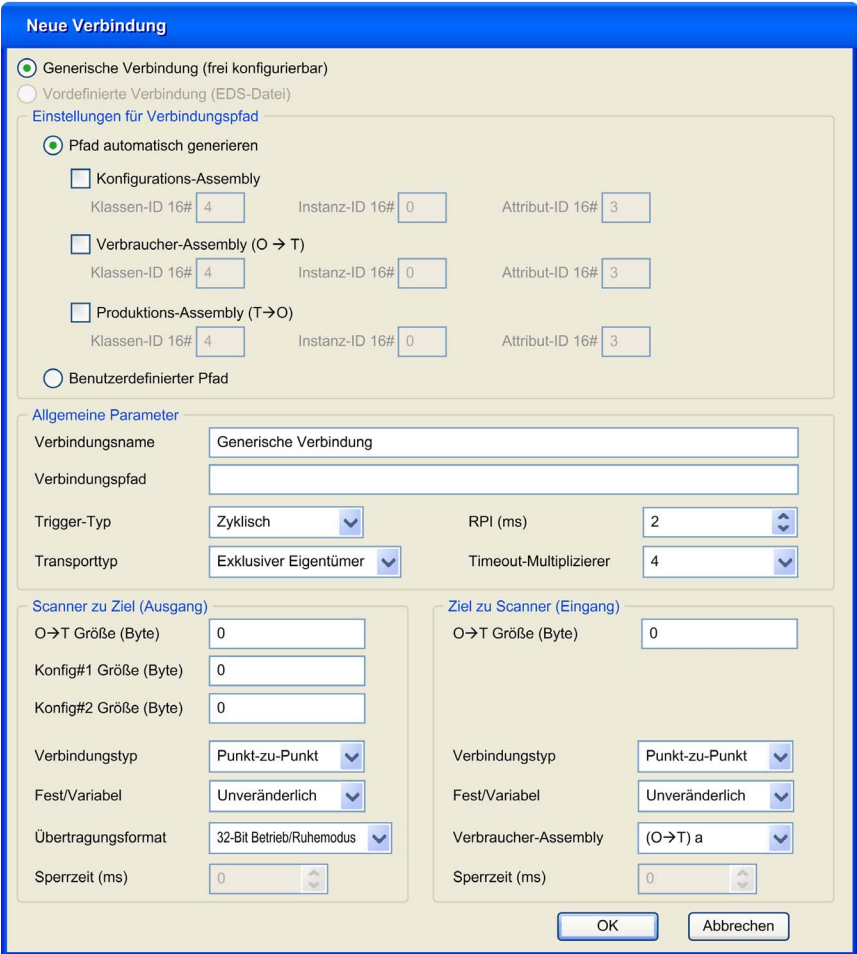
Um eine Verbindung zu ändern, wählen Sie die betreffende Verbindung aus und klicken Sie auf **Verbindung bearbeiten** oder doppelklicken Sie auf die Verbindung.

Um eine Verbindung zu entfernen, wählen Sie diese aus und klicken Sie auf **Verbindung löschen**.

### Hinzufügen einer EtherNet/IP-Verbindung

Gehen Sie wie folgt vor, um eine Verbindungen zu erstellen und zu konfigurieren:

Schritt	Aktion
1	Doppelklicken Sie in der <b>Gerätebaumstruktur</b> auf ein EtherNet/IP-Gerät.
2	Wählen Sie die Registerkarte <b>Verbindungen</b> aus.
3	Klicken Sie auf <b>Verbindung hinzufügen</b> .
<sup>(1)</sup> Die Klassen-ID, Instanz-ID und Attribut-ID können in der Gerätedokumentation gefunden werden.	

Schritt	Aktion
4	<p>Wählen Sie <b>Generische Verbindung (frei konfigurierbar)</b> aus:</p>  <p><b>Neue Verbindung</b></p> <p><input checked="" type="radio"/> Generische Verbindung (frei konfigurierbar)  <input type="radio"/> Vordefinierte Verbindung (EDS-Datei)</p> <p><b>Einstellungen für Verbindungspfad</b></p> <p><input checked="" type="radio"/> Pfad automatisch generieren</p> <p><input type="checkbox"/> Konfigurations-Assembly      Klassen-ID 16# <input type="text" value="4"/> Instanz-ID 16# <input type="text" value="0"/> Attribut-ID 16# <input type="text" value="3"/></p> <p><input type="checkbox"/> Verbraucher-Assembly (O → T)      Klassen-ID 16# <input type="text" value="4"/> Instanz-ID 16# <input type="text" value="0"/> Attribut-ID 16# <input type="text" value="3"/></p> <p><input type="checkbox"/> Produktions-Assembly (T → O)      Klassen-ID 16# <input type="text" value="4"/> Instanz-ID 16# <input type="text" value="0"/> Attribut-ID 16# <input type="text" value="3"/></p> <p><input type="radio"/> Benutzerdefinierter Pfad</p> <p><b>Allgemeine Parameter</b></p> <p>Verbindungsname <input type="text" value="Generische Verbindung"/></p> <p>Verbindungspfad <input type="text"/></p> <p>Trigger-Typ <input type="button" value="Zyklisch"/> RPI (ms) <input type="text" value="2"/></p> <p>Transporttyp <input type="button" value="Exklusiver Eigentümer"/> Timeout-Multiplizierer <input type="text" value="4"/></p> <p><b>Scanner zu Ziel (Ausgang)</b></p> <p>O→T Größe (Byte) <input type="text" value="0"/></p> <p>Konfig#1 Größe (Byte) <input type="text" value="0"/></p> <p>Konfig#2 Größe (Byte) <input type="text" value="0"/></p> <p>Verbindungstyp <input type="button" value="Punkt-zu-Punkt"/></p> <p>Fest/Variabel <input type="button" value="Unveränderlich"/></p> <p>Übertragungsformat <input type="button" value="32-Bit Betrieb/Ruhemodus"/></p> <p>Sperrzeit (ms) <input type="text" value="0"/></p> <p><b>Ziel zu Scanner (Eingang)</b></p> <p>O→T Größe (Byte) <input type="text" value="0"/></p> <p>Verbindungstyp <input type="button" value="Punkt-zu-Punkt"/></p> <p>Fest/Variabel <input type="button" value="Unveränderlich"/></p> <p>Verbraucher-Assembly <input type="button" value="(O→T) a"/></p> <p>Sperrzeit (ms) <input type="text" value="0"/></p> <p><input type="button" value="OK"/> <input type="button" value="Abbrechen"/></p>
5	<p>Wählen Sie <b>Pfad automatisch erzeugen</b> aus und konfigurieren Sie die <b>Konfigurationsassembly</b>:</p> <ul style="list-style-type: none"> <li>● <b>Klassen-ID</b> (Standardwert: 4): Klassenbezeichner<sup>(1)</sup></li> <li>● <b>Instanz-ID</b>: Instanzbezeichner<sup>(1)</sup></li> <li>● <b>Attribut-ID</b> (Standardwert: 3): Attributbezeichner<sup>(1)</sup></li> </ul>
<p><sup>(1)</sup> Die Klassen-ID, Instanz-ID und Attribut-ID können in der Gerätedokumentation gefunden werden.</p>	

Schritt	Aktion
6	Konfigurieren Sie die <b>Verbraucher-Assembly (O --&gt; T)</b> : <ul style="list-style-type: none"><li>● <b>Klassen-ID</b> (Standardwert: 4): Klassenbezeichner<sup>(1)</sup></li><li>● <b>Instanz-ID</b>: Instanzbezeichner<sup>(1)</sup></li><li>● <b>Attribut-ID</b> (Standardwert: 3): Attributbezeichner<sup>(1)</sup></li></ul>
7	Konfigurieren Sie die <b>Produktions-Assembly (T --&gt; O)</b> : <ul style="list-style-type: none"><li>● <b>Klassen-ID</b> (Standardwert: 4): Klassenbezeichner<sup>(1)</sup></li><li>● <b>Instanz-ID</b>: Instanzbezeichner<sup>(1)</sup></li><li>● <b>Attribut-ID</b> (Standardwert: 3): Attributbezeichner<sup>(1)</sup></li></ul>
<sup>(1)</sup> Die Klassen-ID, Instanz-ID und Attribut-ID können in der Gerätedokumentation gefunden werden.	

Schritt	Aktion
8	<p>Konfigurieren Sie die <b>Generischen Parameter</b>:</p> <ul style="list-style-type: none"> <li>● <b>Verbindungsname</b>. Der standardmäßige Verbindungsname wird automatisch von SoMachine generiert. Für generische Verbindungen kann der Standardname „generische Verbindung“ bearbeitet werden.</li> <li>● <b>Verbindungspfad</b>. Kodierte Transkription des physischen Verbindungsobjekts. Kann für generische Verbindungen bearbeitet werden.</li> <li>● <b>Trigger-Typ</b>. Wählen Sie aus, wie der Datenaustausch initiiert wird: <ul style="list-style-type: none"> <li>○ <b>Zyklisch</b>: Endpunkte tauschen Daten in vorab festgelegten regelmäßigen Zeitintervallen aus.</li> <li>○ <b>Statusänderung</b>: Endpunkte tauschen Daten nur bei Datenänderungen aus. Um eine Zeitüberschreitung der Verbindung zu verhindern, wenn keine Änderungen vorkommen, werden Daten außerdem in einem zyklischen Intervall in Hintergrund ausgetauscht (siehe RPI unten). Wenn die <b>Statusänderung</b> ausgewählt ist, dann sind die Felder für die <b>Sperzeit (ms)</b> in den Verbindungseigenschaften bei Scanner-zu-Ziel und Ziel-zu-Scanner aktiviert.</li> <li>○ <b>Anwendung</b>. Der Datenaustausch wird durch eine Applikation initiiert.</li> </ul> </li> <li>● <b>Transporttyp</b>: <ul style="list-style-type: none"> <li>○ <b>Exklusiver Eigentümer</b>: Dies ist eine bidirektionale Verbindung zu einem Ausgangsverbindungspunkt (in der Regel ein Assembly-Objekt), bei der die Daten dieser Assembly nur von einem Scanner gesteuert werden können. Es kann eine Verbindung zu einer Eingangs-Assembly vorhanden sein; diese Daten werden an den Scanner gesendet.</li> <li>○ <b>Nur Überwachung</b>: Der Scanner empfängt Eingangsdaten vom Zielgerät und erzeugt einen Heartbeat zum Zielgerät. Es gibt keine Ausgangsdaten. Eine Verbindung vom Typ <b>Nur Überwachung</b> kann nur mit einer vorhandenen Verbindung vom Typ <b>Exklusiver Eigentümer</b> oder <b>Nur Eingang</b> verknüpft werden. Wenn diese zugrundeliegende Verbindung abbricht, bricht auch die Verbindung vom Typ <b>Nur Überwachung</b> ab oder es tritt ein Timeout auf.</li> <li>○ <b>Nur Eingang</b>: Der Scanner empfängt Eingangsdaten vom Zielgerät und erzeugt einen Heartbeat zum Zielgerät. Es gibt keine Ausgangsdaten.</li> </ul> </li> <li>● <b>RPI (ms)</b> Angefordertes Paketintervall (Requested Packet Interval). Die Zeitdauer zwischen vom Scanner angeforderten zyklischen Datenübertragungen. Zielgeräte stellen immer einen minimalen RPI bereit, während das Ziel der Steuerung ist, das höchste RPI zu haben, um das System nicht zu überlasten. Überprüfen Sie die von den Zielgeräten verwendeten Steuerungsressourcen immer dann, wenn ein Gerät an den EtherNet/IP-Feldbus hinzugefügt oder der RPI-Wert geändert wird. Das Geräte-RPI kann in der Gerätedokumentation spezifiziert sein, ist aber normalerweise Teil der im Lieferumfang des Geräts begriffenen EDS-Datei.</li> <li>● Wählen Sie den <b>Timeout-Multiplizierer</b> aus: 4 (Standardwert) / 8 / 16 / 32 / 64 / 128 / 256 / 512. Der ausgewählte Wert wird mit dem RPI-Wert multipliziert, um den Wert für die Zeitüberschreitung (Timeout) der Verbindung zu erhalten.</li> </ul>
<p>(1) Die Klassen-ID, Instanz-ID und Attribut-ID können in der Gerätedokumentation gefunden werden.</p>	

Schritt	Aktion
9	<p>Konfigurieren der spezifischen Parameter für <b>Scanner zu Ziel (Ausgang)</b>:</p> <ul style="list-style-type: none"> <li>● <b>O → T Größe (Byte)</b>: Anzahl der zu übertragenden Byte.</li> <li>● <b>Konfig#1 Größe (Byte)</b>: Anzahl der Parameter im ersten Konfigurationsparametersatz.</li> <li>● <b>Konfig#2 Größe (Byte)</b>: Anzahl der Parameter im zweiten Konfigurationsparametersatz.</li> <li>● <b>Verbindungstyp</b>. Zu verwendender Verbindungstyp: <ul style="list-style-type: none"> <li>○ <b>Multicast</b>. Die Verbindung besteht zwischen dem Scanner und mehreren Zielgeräten.</li> <li>○ <b>Punkt-zu-Punkt</b>. Die Verbindung besteht zwischen dem Scanner und einem einzelnen Zielgerät.</li> </ul> </li> <li>● <b>Fest/Variabel</b>. Bestimmt, ob der zu übertragende Datenumfang immer gleich (<b>Fest</b>) ist oder nur der exakte Umfang gepufferter Daten übertragen wird (<b>Variabel</b>).</li> <li>● <b>Übertragungsformat</b>. Das bei der Verbindung zu verwendende Echtzeit-Datenformat: <ul style="list-style-type: none"> <li>○ <b>32-Bit Betrieb/Ruhemodus</b>. Ein 32-Bit-Paket-Header enthält eine Betrieb/Ruhemodus-Benachrichtigung.</li> <li>○ <b>Reine Daten</b>. Keine Betrieb/Ruhemodus-Benachrichtigung</li> <li>○ <b>Heartbeat</b>. Keine Betrieb/Ruhemodus-Benachrichtigung</li> <li>○ <b>Ruhemodus mit Nulllänge</b>. Ein Nulllängen-Datenformat zeigt den Ruhemodus an.</li> </ul> </li> </ul> <p>Für weitere Informationen, siehe <a href="#">OVDA website</a>.</p> <ul style="list-style-type: none"> <li>● <b>Sperrzeit (ms)</b>: Mindestdauer zwischen 2 Datenaustauschvorgängen. Nur verfügbar, wenn als <b>Trigger-Typ</b> die Einstellung <b>Statusänderung</b> gewählt wurde. Der Wert muss ein Vielfaches von 2 ms sein. Der Maximalwert ist der <b>RPI (ms)</b>-Wert mit einem möglichen Maximalwert von 254 ms.</li> </ul>
10	<p>Konfigurieren der spezifischen Parameter für <b>Ziel zu Scanner (Eingang)</b>:</p> <ul style="list-style-type: none"> <li>● <b>T → O Größe (Byte)</b>: Anzahl der zu übertragenden Byte.</li> <li>● <b>Verbindungstyp</b>. Zu verwendender Verbindungstyp: <ul style="list-style-type: none"> <li>○ <b>Multicast</b>. Die Verbindung besteht zwischen dem Ziel und mehreren Scannern.</li> <li>○ <b>Punkt-zu-Punkt</b>. Die Verbindung besteht zwischen dem Ziel und einem einzelnen Scannern.</li> </ul> </li> <li>● <b>Fest/Variabel</b>. Bestimmt, ob der zu übertragende Datenumfang immer gleich (<b>Fest</b>) ist oder nur der exakte Umfang gepufferter Daten übertragen wird (<b>Variabel</b>).</li> <li>● <b>Verbraucher Assembly (O → T)</b></li> <li>● <b>Sperrzeit (ms)</b>: Mindestdauer zwischen 2 Datenaustauschvorgängen. Verfügbar, wenn als <b>Trigger-Typ</b> die Einstellung <b>Statusänderung</b> gewählt wurde. Der Wert muss ein Vielfaches von 2 ms sein. Der Maximalwert ist der <b>RPI (ms)</b>-Wert mit einem möglichen Maximalwert von 254 ms.</li> </ul>
11	Klicken Sie auf <b>OK</b> .
(1) Die Klassen-ID, Instanz-ID und Attribut-ID können in der Gerätedokumentation gefunden werden.	

Detaillierte Informationen zu den unterstützten Assemblies finden Sie in der Dokumentation des Geräts.

**HINWEIS:** Aufgrund der Beschränkungen bei **O → T Größe (Byte)** und **T → O Größe (Byte)** sowie der maximalen Anzahl an Ein-/Ausgangswörtern (*siehe Seite 135*) des Scanners sollten Sie die Scanner-Ressourcen-Last nach dem Hinzufügen einer Verbindung prüfen.

### Vordefinierte Verbindung hinzufügen

Vordefinierte Verbindungen (siehe Unterstützte Geräte (*siehe Seite 135*)) sind verfügbar für:

- Vordefinierte Geräte
- Andere Geräte, die mit einer EDS-Datei ausgeliefert werden.

Per Definition haben generische Slavegeräte keine vordefinierte Verbindung.

Gehen Sie wie folgt vor, um eine vordefinierte Verbindung hinzuzufügen:

Schritt	Aktion
1	Doppelklicken Sie in der <b>Gerätebaumstruktur</b> auf ein EtherNet/IP-Gerät.
2	Wählen Sie die Registerkarte <b>Verbindungen</b> aus.
3	Klicken Sie auf <b>Verbindung hinzufügen</b> .
4	Wählen Sie <b>Vordefinierte Verbindung (EDS-Datei)</b> aus:

Schritt	Aktion																																								
5	<p data-bbox="360 199 875 224">Wählen Sie eine der vordefinierten Verbindungen aus:</p> <div data-bbox="360 232 1245 1187" style="border: 1px solid blue; padding: 5px;"> <p data-bbox="388 248 536 269"><b>Neue Verbindung</b></p> <p data-bbox="381 285 701 334"> <input type="radio"/> Generische Verbindung (frei konfigurierbar)  <input checked="" type="radio"/> Vordefinierte Verbindung (EDS-Datei)         </p> <p data-bbox="388 337 552 358">Verbindung auswählen</p> <table border="1" data-bbox="395 367 1218 646"> <thead> <tr> <th>Verbindungsname</th> <th>RPI (ms)</th> <th>O-&gt; T Größe (Byte)</th> <th>T-&gt; O Größe (Byte)</th> <th>Konfig#1 Größe (Byte)</th> </tr> </thead> <tbody> <tr> <td>20-70 Exklusiver Eigentümer</td> <td>10</td> <td>4</td> <td>4</td> <td></td> </tr> <tr style="background-color: #e0f0ff;"> <td>20-70 Nur-Eingang</td> <td>10</td> <td>4</td> <td>4</td> <td></td> </tr> <tr> <td>20-70 Nur Überwachung</td> <td>10</td> <td>4</td> <td>4</td> <td></td> </tr> <tr> <td>21-71 Exklusiver Eigentümer</td> <td>10</td> <td>4</td> <td>4</td> <td></td> </tr> <tr> <td>21-71 Nur-Eingang</td> <td>10</td> <td>4</td> <td>4</td> <td></td> </tr> <tr> <td>21-71 Nur Überwachung</td> <td>10</td> <td>4</td> <td>4</td> <td></td> </tr> <tr> <td>22-72 Exklusiver Eigentümer</td> <td>10</td> <td>6</td> <td>6</td> <td></td> </tr> </tbody> </table> <p data-bbox="395 662 550 683">Allgemeine Parameter</p> <p data-bbox="395 691 1184 712">Verbindungsname: 20-70 Nur-Eingang</p> <p data-bbox="395 721 1184 742">Verbindungspfad: 20 04 24 06 2C FF 2C 46</p> <p data-bbox="395 750 1184 771">Trigger-Typ: Zyklisch RPI (ms): 16</p> <p data-bbox="395 779 1184 800">Transporttyp: Nur Eingang Timeout-Multiplizierer: 4</p> <div style="display: flex; justify-content: space-between;"> <div data-bbox="395 841 779 1133"> <p data-bbox="395 841 577 862">Scanner zu Ziel (Ausgang)</p> <p data-bbox="395 870 738 891">O→T Größe (Byte): 4</p> <p data-bbox="395 899 738 920">Konfig#1 Größe (Byte): 0</p> <p data-bbox="395 928 738 950">Konfig#2 Größe (Byte): 0</p> <p data-bbox="395 990 738 1011">Verbindungstyp: Punkt-zu-Punkt</p> <p data-bbox="395 1019 738 1040">Fest/Variabel: Festgelegt</p> <p data-bbox="395 1049 769 1070">Übertragungsformat: 32-Bit Betrieb/Ruhemodus</p> <p data-bbox="395 1078 701 1099">Sperrzeit (ms): 0</p> </div> <div data-bbox="806 841 1184 1133"> <p data-bbox="806 841 1002 862">Ziel zu Scanner (Eingang)</p> <p data-bbox="806 870 1184 891">T→O Größe (Byte): 4</p> <p data-bbox="806 990 1184 1011">Verbindungstyp: Multicast</p> <p data-bbox="806 1019 1184 1040">Fest/Variabel: Festgelegt</p> <p data-bbox="806 1049 1184 1070">Übertragungsformat: Reine Daten</p> <p data-bbox="806 1078 1140 1099">Sperrzeit (ms): 0</p> </div> </div> <p data-bbox="937 1141 1177 1162" style="text-align: right;"> <input type="button" value="OK"/> <input type="button" value="Abbrechen"/> </p> </div>	Verbindungsname	RPI (ms)	O-> T Größe (Byte)	T-> O Größe (Byte)	Konfig#1 Größe (Byte)	20-70 Exklusiver Eigentümer	10	4	4		20-70 Nur-Eingang	10	4	4		20-70 Nur Überwachung	10	4	4		21-71 Exklusiver Eigentümer	10	4	4		21-71 Nur-Eingang	10	4	4		21-71 Nur Überwachung	10	4	4		22-72 Exklusiver Eigentümer	10	6	6	
Verbindungsname	RPI (ms)	O-> T Größe (Byte)	T-> O Größe (Byte)	Konfig#1 Größe (Byte)																																					
20-70 Exklusiver Eigentümer	10	4	4																																						
20-70 Nur-Eingang	10	4	4																																						
20-70 Nur Überwachung	10	4	4																																						
21-71 Exklusiver Eigentümer	10	4	4																																						
21-71 Nur-Eingang	10	4	4																																						
21-71 Nur Überwachung	10	4	4																																						
22-72 Exklusiver Eigentümer	10	6	6																																						
6	<p data-bbox="360 1232 799 1253">Konfigurieren Sie die <b>Generischen Parameter</b>:</p> <ul style="list-style-type: none"> <li data-bbox="360 1261 1245 1334">● <b>RPI (ms)</b> Angefordertes Paketintervall (Requested Packet Interval). Die Zeitdauer zwischen vom Scanner angeforderten zyklischen Datenübertragungen. Der Standardwert ist in der EDS-Datei definiert.</li> <li data-bbox="360 1343 1245 1416">● Wählen Sie den <b>Timeout-Multiplizierer</b> aus: 4 (Standardwert) / 8 / 16 / 32 / 64 / 128 / 256 / 512. Der ausgewählte Wert wird mit dem RPI-Wert multipliziert, um den Wert für die Zeitüberschreitung (Timeout) der Verbindung zu erhalten.</li> </ul>																																								

Schritt	Aktion
7	Konfigurieren Sie <b>Scanner zu Ziel (Ausgang)</b> : <ul style="list-style-type: none"> <li>● <b>O --&gt; T Größe (Byte)</b>: Anzahl der zu übertragenden Byte.</li> </ul>
8	Konfigurieren Sie <b>Ziel zu Scanner (Eingang)</b> : <ul style="list-style-type: none"> <li>● <b>T --&gt; O Größe (Byte)</b>: Anzahl der zu übertragenden Byte (Anzahl der Kanäle der Assembly)</li> <li>● Wählen Sie den <b>Verbindungstyp: Multicast</b> (Standard), wenn die Verbindung zwischen dem Scanner und mehreren Zielgeräten besteht oder <b>Punkt-zu-Punkt</b>, wenn die Verbindung zwischen dem Scanner und einem einzelnen Zielgerät besteht. Nur für bestimmte Arten des <b>Transporttyps</b> bearbeitbar.</li> </ul>
9	Klicken Sie auf <b>OK</b> . <b>Ergebnis:</b> Die Verbindung wird der Registerkarte <b>Verbindungen</b> hinzugefügt.

### Konfigurieren einer Konfigurationsassembly

Einige Geräte unterstützen Konfigurationsassemblies.

Eine Konfigurationsassembly ist eine einfache Request, die beim Starten von EtherNet/IP Scanner gesendet wird, und alle Konfigurationsparameter an das Zielgerät sendet.

Gehen Sie zur Konfiguration einer Konfigurations-Assembly vor wie folgt:

Schritt	Aktion
1	Doppelklicken Sie in der <b>Gerätebaumstruktur</b> auf ein EtherNet/IP-Gerät.
2	Wählen Sie die Registerkarte <b>Verbindungen</b> aus.
3	Wählen Sie eine vorhandene Verbindung aus und klicken Sie auf <b>Verbindung bearbeiten</b> .
4	Wählen Sie <b>Generische Verbindung (frei konfigurierbar)</b> aus.
5	Wählen Sie <b>Konfigurations-Assembly</b> aus.
6	Konfigurieren Sie die <b>Konfigurations-Assembly</b> : <ul style="list-style-type: none"> <li>● <b>Klassen-ID</b> (Standardwert: 4): Klassenbezeichner<sup>(1)</sup></li> <li>● <b>Instanz-ID</b>: Instanzbezeichner<sup>(1)</sup></li> <li>● <b>Attribut-ID</b> (Standardwert: 3): Attributbezeichner<sup>(1)</sup></li> </ul>
7	Konfigurieren Sie <b>Scanner zu Ziel (Ausgang)</b> : <ul style="list-style-type: none"> <li>● <b>Konfig#1 Größe (Byte)</b>: Anzahl der Parameter im ersten Konfigurationsparametersatz.</li> <li>● <b>Konfig#2 Größe (Byte)</b>: Anzahl der Parameter im zweiten Konfigurationsparametersatz.</li> </ul>

Schritt	Aktion																																																																		
8	<p>Klicken Sie auf <b>OK</b>.</p> <p><b>Ergebnis:</b> Die Konfigurationsparameter werden auf der Registerkarte <b>Verbindungen</b> angezeigt:</p> <table border="1"> <thead> <tr> <th>Verbindung Nr.</th> <th>Verbindungsname</th> <th>RPI O-&gt;...</th> <th>RPI T-&gt;...</th> <th>O-&gt; T Größe...</th> <th>T-&gt; O Größe (...)</th> <th>Konfig#1 Größe (Byte)</th> <th>Konfig#2 Größe (Byte)</th> <th>Con</th> </tr> </thead> <tbody> <tr> <td>264</td> <td>Generische Verbindung</td> <td>10</td> <td>10</td> <td>0</td> <td>0</td> <td>4</td> <td>4</td> <td>20</td> </tr> </tbody> </table> <p>Verbindung hinzufügen...    Verbindung löschen...    Verbindung bearbeiten...</p> <p><b>Konfigurationsdaten</b></p> <p><input checked="" type="checkbox"/> Symbolische Werte</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Wert</th> <th>Datentyp</th> <th>Standardwert</th> </tr> </thead> <tbody> <tr> <td>[-] Generische Verbindung</td> <td></td> <td></td> <td></td> </tr> <tr> <td>[-] Konfig#1 Daten</td> <td></td> <td></td> <td></td> </tr> <tr> <td>    Parameter 1</td> <td>0</td> <td>BYTE</td> <td>0</td> </tr> <tr> <td>    Parameter 2</td> <td>0</td> <td>BYTE</td> <td>0</td> </tr> <tr> <td>    Parameter 3</td> <td>0</td> <td>BYTE</td> <td>0</td> </tr> <tr> <td>    Parameter 4</td> <td>0</td> <td>BYTE</td> <td>0</td> </tr> <tr> <td>[-] Konfig#2 Daten</td> <td></td> <td></td> <td></td> </tr> <tr> <td>    Parameter 5</td> <td>0</td> <td>BYTE</td> <td>0</td> </tr> <tr> <td>    Parameter 6</td> <td>0</td> <td>BYTE</td> <td>0</td> </tr> <tr> <td>    Parameter 7</td> <td>0</td> <td>BYTE</td> <td>0</td> </tr> <tr> <td>    Parameter 8</td> <td>0</td> <td>BYTE</td> <td>0</td> </tr> </tbody> </table>	Verbindung Nr.	Verbindungsname	RPI O->...	RPI T->...	O-> T Größe...	T-> O Größe (...)	Konfig#1 Größe (Byte)	Konfig#2 Größe (Byte)	Con	264	Generische Verbindung	10	10	0	0	4	4	20	Parameter	Wert	Datentyp	Standardwert	[-] Generische Verbindung				[-] Konfig#1 Daten				Parameter 1	0	BYTE	0	Parameter 2	0	BYTE	0	Parameter 3	0	BYTE	0	Parameter 4	0	BYTE	0	[-] Konfig#2 Daten				Parameter 5	0	BYTE	0	Parameter 6	0	BYTE	0	Parameter 7	0	BYTE	0	Parameter 8	0	BYTE	0
Verbindung Nr.	Verbindungsname	RPI O->...	RPI T->...	O-> T Größe...	T-> O Größe (...)	Konfig#1 Größe (Byte)	Konfig#2 Größe (Byte)	Con																																																											
264	Generische Verbindung	10	10	0	0	4	4	20																																																											
Parameter	Wert	Datentyp	Standardwert																																																																
[-] Generische Verbindung																																																																			
[-] Konfig#1 Daten																																																																			
Parameter 1	0	BYTE	0																																																																
Parameter 2	0	BYTE	0																																																																
Parameter 3	0	BYTE	0																																																																
Parameter 4	0	BYTE	0																																																																
[-] Konfig#2 Daten																																																																			
Parameter 5	0	BYTE	0																																																																
Parameter 6	0	BYTE	0																																																																
Parameter 7	0	BYTE	0																																																																
Parameter 8	0	BYTE	0																																																																
9	<p>Doppelklicken Sie in die Spalte <b>Wert</b>, um die Konfigurationsparameterwerte festzulegen.</p> <p><sup>(1)</sup> Die Klassen-ID, Instanz-ID und Attribut-ID können in der Gerätedokumentation gefunden werden. Siehe Finden von Assembly-Informationen (<i>siehe Seite 164</i>).</p>																																																																		

### Bearbeiten vordefinierter Verbindungen

Um eine vordefinierte Verbindung zu bearbeiten, wählen Sie die Verbindung in der Registerkarte **Verbindungen** aus und klicken auf **Verbindung bearbeiten**:

**Verbindung bearbeiten**

**Allgemeine Parameter**

Verbindungsname: 100-101 Exklusiver Eigentümer

Verbindungspfad: 20 04 24 06 2C 64 2C 65

Trigger-Typ: Zyklisch      RPI (ms): 16

Transporttyp: Exklusiver Eigentümer      Timeout-Multiplizierer: 4

**Scanner zu Ziel (Ausgang)**

O→T Größe (Byte): 16

Konfig#1 Größe (Byte): 0

Konfig#2 Größe (Byte): 0

Verbindungstyp: Punkt-zu-Punkt

Fest/Variabel: Festgelegt

Übertragungsformat: 32-Bit Betrieb/Ruhemodus

Sperrzeit (ms): 0

**Ziel zu Scanner (Eingang)**

T→O Größe (Byte): 16

Verbindungstyp: Punkt-zu-Punkt

Fest/Variabel: Festgelegt

Übertragungsformat: Reine Daten

Sperrzeit (ms): 0

Parameter	Beschreibung
<b>Allgemeine Parameter</b>	
<b>RPI (ms)</b>	<b>RPI (ms)</b> Angefordertes Paketintervall (Requested Packet Interval). Die Zeitdauer zwischen vom Scanner angeforderten zyklischen Datenübertragungen. Das Gerät stellt immer einen minimalen RPI bereit, während das Ziel der Steuerung ist, den höchsten RPI zu haben, um das System nicht zu überlasten. Überprüfen Sie die von den Zielgeräten verwendeten Ressourcen immer dann, wenn ein Gerät an den EtherNet/IP-Feldbus hinzugefügt oder der RPI-Wert geändert wird. Das Geräte-RPI kann in der Gerätedokumentation spezifiziert sein, ist aber normalerweise Teil der im Lieferumfang des Geräts inbegriffenen EDS-Datei.
<b>Timeout-Multiplizierer</b>	Wählen Sie den <b>Timeout-Multiplizierer</b> aus: 4 (Standardwert) / 8 / 16 / 32 / 64 / 128 / 256 / 512. Der ausgewählte Wert wird mit dem RPI-Wert multipliziert, um den Wert für die Zeitüberschreitung (Timeout) der Verbindung zu erhalten.

Parameter	Beschreibung
<b>Scanner zu Ziel (Ausgang)</b>	
<b>O --&gt; T Größe (Byte)</b>	Größe eines Kanals für eine Assembly. Die Speichergröße jedes Kanals ist 2 Byte, in denen der Wert des %IWx- oder %QWx-Objekts gespeichert wird, wobei x für die Kanalnummer steht.
<b>Ziel zu Scanner (Eingang)</b>	
<b>T --&gt; O Größe (Byte)</b>	<b>T --&gt; O Größe (Byte):</b> Anzahl der zu übertragenden Byte (Anzahl der Kanäle der Assembly) Die Speichergröße jedes Kanals ist 2 Byte, in denen der Wert des %IWx- oder %QWx-Objekts gespeichert wird, wobei x für die Kanalnummer steht.
<b>Verbindungstyp</b>	Verbindungstyp des Requests: <ul style="list-style-type: none"> <li>● <b>Multicast</b> (Standard), wenn die Verbindung zwischen dem Scanner und mehreren Zielgeräten besteht</li> <li>● <b>Punkt-zu-Punkt</b>, wenn die Verbindung zwischen dem Scanner und einem einzelnen Zielgerät besteht</li> </ul> Nur für bestimmte Arten des <b>Transporttyps</b> bearbeitbar.
<b>Sperrzeit (ms)</b>	Minstdauer zwischen 2 Datenaustauschvorgängen Verfügbar, wenn als <b>Trigger-Typ</b> die Einstellung <b>Statusänderung</b> gewählt wurde. Der Maximalwert der Sperrzeit ist RPI und auf 254 ms begrenzt.

### Bearbeiten generischer Verbindungen

Um eine generische Verbindung zu bearbeiten, wählen Sie die Verbindung in der Registerkarte **Verbindungen** aus und klicken auf **Verbindung bearbeiten**:

#### Verbindung bearbeiten

**Einstellungen für Verbindungspfad**

Pfad automatisch generieren

Konfigurations-Assembly  
 Klassen-ID 16#  Instanz-ID 16#  Attribut-ID 16#

Verbraucher-Assembly (O → T)  
 Klassen-ID 16#  Instanz-ID 16#  Attribut-ID 16#

Produktions-Assembly (T → O)  
 Klassen-ID 16#  Instanz-ID 16#  Attribut-ID 16#

Benutzerdefinierter Pfad

**Allgemeine Parameter**

Verbindungsname

Verbindungspfad

Trigger-Typ  RPI (ms)

Transporttyp  Timeout-Multiplizierer

**Scanner zu Ziel (Ausgang)**

O→T Größe (Byte)

Konfig#1 Größe (Byte)

Konfig#2 Größe (Byte)

Verbindungstyp

Fest/Variabel

Übertragungsformat

Sperrzeit (ms)

**Ziel zu Scanner (Eingang)**

T→O Größe (Byte)

Verbindungstyp

Fest/Variabel

Übertragungsformat

Sperrzeit (ms)

Gehen Sie wie folgt vor:

Parameter	Werte	Beschreibung	
<b>Einstellungen für Verbindungspfad</b>			
<b>Pfad automatisch generieren</b>	Ja/Nein	Ermöglicht das Konfigurieren der Parameter der Assembly	
	<b>Konfigurations-Assembly</b>	Aktiviert/Nicht aktiviert	Ermöglicht das Konfigurieren einer Konfigurations-Assembly ( <i>siehe Seite 164</i> ).
	<b>Klassen-ID</b>	2 Byte (Standardwert: 04h)	Klassenbezeichner <sup>(1)</sup>
	<b>Instanz-ID</b>	2 Byte (Standardwert: 0)	Instanzbezeichner <sup>(1)</sup>
	<b>Attribut-ID</b>	2 Byte (Standardwert: 03h)	Attributbezeichner <sup>(1)</sup>
	<b>Verbraucher Assembly (O --&gt; T)</b>		
	<b>Klassen-ID</b>	2 Byte (Standardwert: 04h)	Klassenbezeichner <sup>(1)</sup>
	<b>Instanz-ID</b>	2 Byte (Standardwert: 0)	Instanzbezeichner <sup>(1)</sup>
	<b>Attribut-ID</b>	2 Byte (Standardwert: 03h)	Attributbezeichner <sup>(1)</sup>
	<b>Produktions-Assembly (T --&gt; O)</b>		
	<b>Klassen-ID</b>	2 Byte (Standardwert: 04h)	Klassenbezeichner <sup>(1)</sup>
<b>Instanz-ID</b>	2 Byte (Standardwert: 0)	Instanzbezeichner <sup>(1)</sup>	
<b>Attribut-ID</b>	2 Byte (Standardwert: 03h)	Attributbezeichner <sup>(1)</sup>	
<b>Benutzerdefinierter Pfad</b>	Ja/Nein	Entfernen Sie die Markierung bei <b>Pfad automatisch generieren</b> . Dadurch wird das Feld <b>Verbindungspfad</b> aktiviert..	
<b>Allgemeine Parameter</b>			
<b>Verbindungsname</b>	Textzeichenfolge	Geben Sie den Namen der generischen Verbindung ein. Der Standardwert ist <b>generische Verbindung</b> .	
<b>Verbindungspfad</b>	Byte-Array	Kodierte Transkription des physischen Verbindungsobjekts, wenn <b>Pfad automatisch erzeugen</b> ausgewählt ist. Bearbeitbar, wenn <b>benutzerdefinierter Pfad</b> ausgewählt ist.	
<sup>(1)</sup> Die Klassen-ID, Instanz-ID und Attribut-ID können in der Gerätedokumentation gefunden werden. Siehe Finden von Assembly-Informationen ( <i>siehe Seite 164</i> ).			

Parameter	Werte	Beschreibung
Trigger-Typ	<ul style="list-style-type: none"> <li>● <b>Zyklisch</b> (Standard)</li> <li>● <b>Statusänderung</b></li> <li>● <b>Anwendung</b></li> </ul>	<p>Wählen Sie aus, wie der Datenaustausch initiiert wird:</p> <ul style="list-style-type: none"> <li>● <b>Zyklisch:</b> Endpunkte tauschen Daten in vorab festgelegten regelmäßigen Zeitintervallen aus.</li> <li>● <b>Statusänderung:</b> Endpunkte tauschen Daten nur bei Datenänderungen aus. Um eine Zeitüberschreitung der Verbindung zu verhindern, werden Daten außerdem in einem zyklischen Intervall in Hintergrund ausgetauscht (siehe RPI unten), wenn keine Datenänderungen vorkommen. Wenn die <b>Statusänderung</b> ausgewählt ist, dann sind die Felder für die <b>Sperrzeit (ms)</b> in den Verbindungseigenschaften bei Scanner-zu-Ziel und Ziel-zu-Scanner aktiviert.</li> <li>● <b>Anwendung:</b> Der Datenaustausch wird durch die Applikation initiiert.</li> </ul>
Transporttyp	<ul style="list-style-type: none"> <li>● <b>Exklusiver Eigentümer</b> (Standardwert)</li> <li>● <b>Redundanter Eigentümer</b></li> </ul>	<p><b>Exklusiver Eigentümer:</b> Dies ist eine bidirektionale Verbindung zu einem Ausgangsverbindungspunkt (in der Regel ein Assembly-Objekt), bei der die Daten dieser Assembly nur von einem Scanner gesteuert werden können. Es kann eine Verbindung zu einer Eingangs-Assembly vorhanden sein; diese Daten werden an den Scanner gesendet. Wenn die Eingangsdatenlänge null ist, wird diese Richtung zur Heartbeat-Richtung.</p> <p><b>Redundanter Eigentümer.</b> Ermöglicht es mehreren separaten Urheberanwendungen jeweils eine unabhängige, identische Verbindung zum Transport eines Zielgeräts herzustellen.</p>
RPI (ms)	In ms (Standardwert: 10 ms)	<p>Intervall für angeforderte Pakete (Requested Packet Interva): Die Zeitdauer zwischen vom Scanner angeforderten zyklischen Datenübertragungen.</p> <p>Der Geräte-RPI kann in der Gerätedokumentation angegeben werden. Normalerweise wird diese Information jedoch als Teil der mit dem Gerät gelieferten EDS-Datei zur Verfügung gestellt.</p>
<p>(1) Die Klassen-ID, Instanz-ID und Attribut-ID können in der Gerätedokumentation gefunden werden. Siehe Finden von Assembly-Informationen (<i>siehe Seite 164</i>).</p>		

Parameter	Werte	Beschreibung
<b>Timeout-Multiplizierer</b>	4 (Standardwert) / 8 / 16 / 32 / 64 / 128 / 256 / 512	Die Zeitüberschreitung des Scanners wird pro Verbindung verwaltet, indem der RPI-Wert und der Timeout-Multiplikator-Wert multipliziert werden.
<b>Scanner zu Ziel (Ausgang)</b>		
<b>O --&gt; T Größe (Byte)</b>	0 bis XX => gerätespezifisch	Größe eines Kanals für eine Assembly. Die Speichergröße jedes Kanals ist 2 Byte, in denen der Wert des %IWx- oder %QWx-Objekts gespeichert wird, wobei x für die Kanalnummer steht.
<b>Konfig#1 Größe (Byte)</b>	0 bis XX => gerätespezifisch	Verfügbar, wenn der Verbindungspfad eine Konfigurations-Assembly enthält Anzahl der zu übertragenden Parameter (1 Byte). Die Konfigurationswerte werden beim Start des Scanners an das Gerät gesendet.
<b>Konfig#2 Größe (Byte)</b>	0 bis XX => gerätespezifisch	
<b>Verbindungstyp</b>	Punkt-zu-Punkt	Verbindungstyp des Requests
<b>Fest/Variabel</b>	Fest	Die Requestlänge ist fest.
<b>Übertragungsformat</b>	<ul style="list-style-type: none"> <li>● 32-Bit Betrieb/Ruhemodus (Standardwert)</li> <li>● Reine Daten</li> <li>● Heartbeat</li> </ul>	Übertragungsformat des Requests. Weitere Informationen finden Sie unter <a href="#">ODVA website</a> .
<b>Sperrzeit</b>	0 ms	Mindestdauer zwischen 2 Datenaustauschvorgängen
<b>Ziel zu Scanner (Eingang)</b>		
<b>T --&gt; O Größe (Byte)</b>	0 bis XX => gerätespezifisch	Größe eines Kanals einer Assembly. Die Speichergröße jedes Kanals ist 2 Byte, in denen der Wert des %IWx- oder %QWx-Objekts gespeichert wird, wobei x für die Kanalnummer steht.
<b>Verbindungstyp</b>	<ul style="list-style-type: none"> <li>● Multicast (Standard)</li> <li>● Punkt-zu-Punkt</li> </ul>	Verbindungstyp des Requests
<b>Fest/Variabel</b>	Fest	Die Requestlänge ist fest.
<b>Übertragungsformat</b>	<ul style="list-style-type: none"> <li>● 32-Bit Betrieb Ruhemodus</li> <li>● Reine Daten (Standardwert)</li> <li>● Heartbeat</li> <li>● Ruhemodus mit Nulllänge</li> </ul>	Übertragungsformat des Requests. Weitere Informationen finden Sie unter <a href="#">ODVA website</a> .
(1) Die Klassen-ID, Instanz-ID und Attribut-ID können in der Gerätedokumentation gefunden werden. Siehe Finden von Assembly-Informationen ( <a href="#">siehe Seite 164</a> ).		

Parameter	Werte	Beschreibung
<b>Sperrzeit (ms)</b>	Als Vielfaches von 2 ms (Standardwert: 2 ms)	Mindestdauer zwischen 2 Datenaustauschvorgängen Verfügbar, wenn als <b>Trigger-Typ</b> die Einstellung <b>Statusänderung</b> gewählt wurde. Der Maximalwert der Sperrzeit ist RPI und auf 254 ms begrenzt.
(1) Die Klassen-ID, Instanz-ID und Attribut-ID können in der Gerätedokumentation gefunden werden. Siehe Finden von Assembly-Informationen ( <i>siehe Seite 164</i> ).		

### Finden von Assembly-Informationen.

Assembly-Informationen werden in der Gerätedokumentation zur Verfügung gestellt. Es ist normalerweise Teil der Beschreibung von Assembly-Objekten.

Um eine Assembly zu konfigurieren, müssen die folgenden Informationen identifiziert werden:

#### 1. Klassen-ID

Die Klassen-ID des „Assembly-Objektes“ ist in der Gerätedokumentation enthalten und normalerweise gleich 04h.

#### 2. Instanz-ID

Wählen Sie die Assembly-Instanz abhängig von der App und der Geräteart aus. Die Auswahl der Assembly-Instanz wird eine spezielle Zustandsmaschine im Gerät erzeugen:

- **Konfigurationsassembly:** von einigen Geräten unterstützt; überprüfen Sie in der Gerätedokumentation, welche Assembly-Instanz unterstützt wird.
- **Verbraucher-Assembly:** in der Gerätedokumentation manchmal als „Geräteausgang“ bezeichnet (aus der Geräteperspektive).
- **Produzent-Assembly:** in der Gerätedokumentation manchmal als „Geräteeingang“ bezeichnet (aus der Geräteperspektive).

#### 3. Attribut-ID

Suchen Sie nach dem zu lesenden Attribut. Dies entspricht dem während der Verbindung ausgetauschten Datenpuffer.

Die Attributeigenschaft muss für die Produzent-Assembly einen Schreibzugriff und für die Verbraucher-Assembly einen Lesezugriff haben.

Die Attribut-ID ist dieselbe für beide Assemblys. Ihr Wert ist in der Gerätedokumentation enthalten und normalerweise gleich 03h. Es stimmt mit dem Attribut mit Zugriff *Get/Set* überein. Der Name ist oft „Daten“ (Data) und der Datentyp ist „Array des Byte“ (Array of Byte).

## Austausch von Geräten über Benutzerparameter

### Überblick

Sie **Benutzerparameter** konfigurieren, die an das Gerät gesendet werden, um das Ersetzen zu ermöglichen. Dies erfolgt kurz vor dem Starten der Scannerverbindung nach:

- Anwendungsdownload
- Warm-Reset/Kaltstart
- Manuellem Starten einer Verbindung

Bei einigen EtherNet/IP-Geräten sind **Benutzerparameter** vordefiniert.

Auf der Registerkarte **Benutzerparameter** können Sie außerdem weitere Parameter hinzufügen und verwalten.

Informationen zur Wartung finden Sie unter Anwenden der richtigen Gerätekonfiguration.

### Benutzerparameter

Doppelklicken Sie in der **Gerätebaumstruktur** auf ein EtherNet/IP-Gerät, und wählen Sie die Registerkarte **Benutzerparameter** aus:

Übersicht	Zieleinstellungen	Verbindungen	Benutzerparameter	EtherNet/IP-E/A-Abbild	Status	Informationen				
⌂ ⚙️ Neu... 🗑️ Löschen... ✎ Bearbeiten...										
Zeile	Name	Klasse...	Instanz...	Attribut...	Wert	Bitlänge	Abbruch bei Fehler	Bei Fehler Sprung zu Zeile	Nächste Zeile	Kommentar
1	Profil	8B	1	2	2	16	<input type="checkbox"/>	<input type="checkbox"/>	0	
2	Ref. 1 Kanal	8B	1	E	169	16	<input type="checkbox"/>	<input type="checkbox"/>	0	
3	Befehl Kanal 1	8B	1	18	30	16	<input type="checkbox"/>	<input type="checkbox"/>	0	
4	Befehlsumschaltung	8B	1	16	98	16	<input type="checkbox"/>	<input type="checkbox"/>	0	
5	Ref. 2 Umschaltung	8B	1	C	222	16	<input type="checkbox"/>	<input type="checkbox"/>	0	
6	OMA3	AE	1	4	3204	16	<input type="checkbox"/>	<input type="checkbox"/>	0	

Spalte	Beschreibung
<b>Linie</b>	Zeilennummer. Gibt die Reihenfolge der in das Gerät geladenen Parameter an
<b>Name</b>	Name des Parameters
<b>Klasse</b>	Klassen-ID <sup>(1)</sup> der zum Objekt gehörenden Klasse.
<b>Instanz</b>	Instanz-ID <sup>(1)</sup> der zum Objekt gehörenden Instanz.
<b>Attribut</b>	Attribut-ID <sup>(1)</sup> des zum Objekt gehörenden Attributs.
<b>Wert</b>	Wert des Parameters Doppelklicken Sie auf den Wert, um ihn zu ändern. Gegebenenfalls wird eine Liste mit möglichen Werten geöffnet.
<sup>(1)</sup> Die Klassen-ID, Instanz-ID und Attribut-ID können in der Gerätedokumentation gefunden werden. Siehe Finden von Informationen zu benutzerdefinierten Parametern ( <i>siehe Seite 168</i> ).	

Spalte	Beschreibung
<b>Bitlänge</b>	Anzahl Bits des Parameters Wird je nach dem ausgewählten Datentyp des Parameters automatisch geändert.
<b>Abbruch bei Fehler</b>	Wenn diese Option ausgewählt wird, wird die Übertragung der Parameter bei Erkennung eines Fehlers abgebrochen.
<b>Bei Fehler Sprung zu Zeile</b>	Wenn diese Option ausgewählt wird, setzt das Programm bei Erkennung eines Fehlers die Verarbeitung mit der in der Spalte <b>Nächste Zeile</b> angegebenen Zeile fort. Auf diese Weise kann eine Blockierung bei der Initialisierung vermieden oder ein Rücksprung definiert werden.  <b>HINWEIS:</b> Ein Rücksprung kann zu einer niemals endenden Schleife führen, wenn ein bestimmter Parameter nie geschrieben werden kann.
<b>Nächste Zeile</b>	Doppelklicken Sie, um die Zeile einzugeben, zu der gesprungen werden soll (wenn das Kontrollkästchen <b>Bei Fehler Sprung zu Zeile</b> aktiviert ist).
<b>Kommentar</b>	Doppelklicken Sie, um einen Kommentar einzugeben.
(1) Die Klassen-ID, Instanz-ID und Attribut-ID können in der Gerätedokumentation gefunden werden. Siehe Finden von Informationen zu benutzerdefinierten Parametern ( <i>siehe Seite 168</i> ).	

Symbol	Beschreibung
<b>Nach oben</b>	Den ausgewählten Parameter in der Parameterliste nach oben verschieben
<b>Nach unten</b>	Den ausgewählten Parameter in der Parameterliste nach unten verschieben
<b>Neu</b>	Neuen Parameter erstellen.
<b>Delete</b>	Ausgewählten Parameter löschen.
<b>Bearbeiten</b>	Ausgewählten Parameter bearbeiten.

## Erstellen oder Konfigurieren von Anwenderparametern

Klicken Sie auf **Neu**, oder wählen Sie einen Parameter aus und klicken Sie auf **Bearbeiten**:

Parameter auswählen
✕

Name	Klasse	Instanz	Attribut	Typ	Standard	Minimum	Maximum
(Empty table content)							

Parametergruppen anzeigen
  Allgemeine Parameter
 OK

Name:

Klasse:  ▼

Instanz:  ▼

Attribut:  ▼

Datentyp:  ▼

Bitlänge:  ▼

Wert:

Felder	Beschreibung
<b>Name</b>	Name des Parameters
<b>Klasse</b>	Klassen-ID <sup>(1)</sup> der zum Objekttyp gehörenden Klasse..
<b>Instanz</b>	Instanz-ID <sup>(1)</sup> der zur Implementation einer Klasse gehörenden Instanz..
<b>Attribut</b>	Attribut-ID <sup>(1)</sup> des zur Eigenschaft einer Instanz gehörenden Attributs..
<b>Datentyp</b>	Liste mit den möglichen Datentypen.
<b>Bitlänge</b>	Anzahl Bits des Parameters Wird je nach dem ausgewählten <b>Datentyp</b> automatisch geändert.
<b>Wert</b>	Wert des Parameters
<sup>(1)</sup> Die Klassen-ID, Instanz-ID und Attribut-ID können in der Gerätedokumentation gefunden werden. Siehe Finden von Informationen zu benutzerdefinierten Parametern ( <i>siehe Seite 168</i> ).	

### Finden von Informationen zu benutzerdefinierten Parametern

Konfigurierbare Informationen zu benutzerdefinierten Parametern werden in der Gerätedokumentation zur Verfügung gestellt. Es ist normalerweise Teil der Beschreibung von Applikationsobjekten, der expliziten Nachrichtenübertragung oder von Objekten, die der EtherNet/IP-Kategorie 3 angehören.

Der Schreibzugriff für benutzerdefinierte Parameter wird normalerweise für die Klasse oder Instanz definiert, der der Parameter angehört. Der Schreibvorgang wird normalerweise mithilfe eines Dienstes namens `Set_Attribute_Single` oder `Write one attribute` ausgeführt. Andernfalls kann auch ein Dienst-Bezeichner 0x10 (hexadezimal) oder 16 (dezimal) unterstützt werden.

Ein benutzerdefinierter Parameter hat immer die folgenden numerischen Eigenschaften:

- Klasse oder Klassen-ID, normalerweise als Hexadezimalwert ausgedrückt.
- Instanz oder Instanz-ID, normalerweise als Hexadezimalwert ausgedrückt.
- Attribut oder Attribut-ID, normalerweise als Hexadezimalwert ausgedrückt.

Ein benutzerdefinierter Parameter kann auch einen Bezeichner haben, der als dezimale Dreiergruppe (xx/yy/zz) oder als hexadezimale Dreiergruppe (16#xx/yy/zz) ausgedrückt wird.

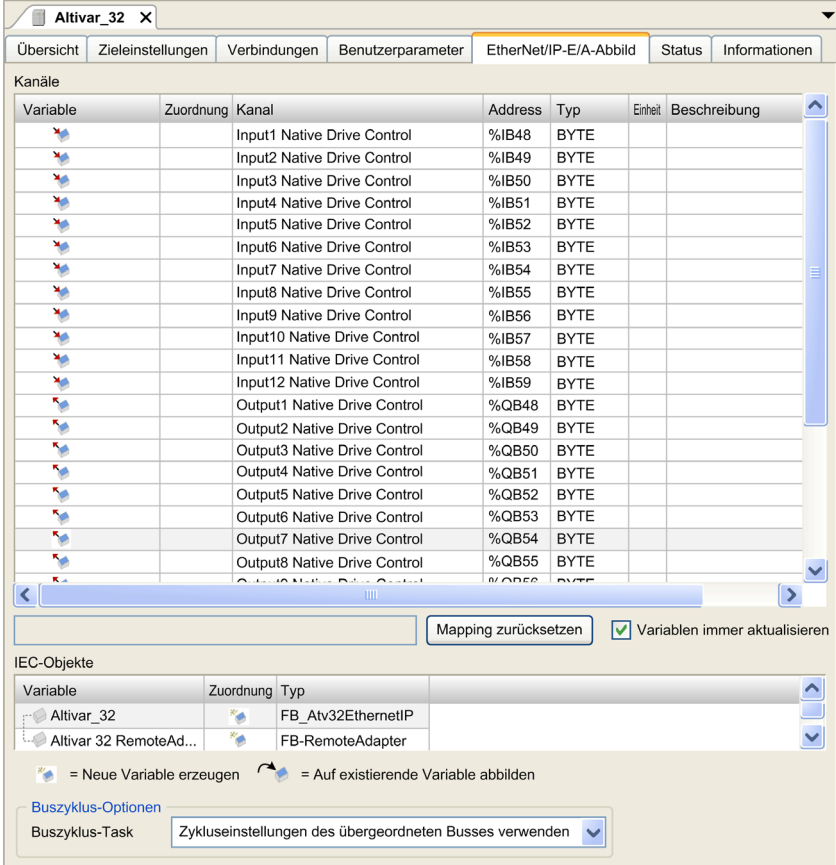
## EtherNet/IP-E/A-Abbild

### Übersicht

Nachdem der Datenaustausch konfiguriert ist, können Sie Variablen zuordnen, die die Applikation verwenden soll.

### Konfigurieren des E/A-Abbilds für ein EtherNet/IP-Zielgerät

Gehen Sie wie folgt vor, um das E/A-Abbild für ein EtherNet/IP-Zielgerät zu konfigurieren:

Schritt	Aktion
1	Doppelklicken Sie in der <b>Gerätebaumstruktur</b> auf ein EtherNet/IP-Gerät. <b>Ergebnis:</b> Das Konfigurationsfenster des Geräts wird angezeigt.
2	Wählen Sie die Registerkarte <b>EtherNet/IPE/A-Abbild</b> . 

Schritt	Aktion
3	Doppelklicken Sie in eine Zelle der Spalte <b>Variable</b> , um ein Textfeld zu öffnen. Geben Sie den Namen einer Variablen ein oder klicken Sie auf die Durchsuchen-Schaltfläche [...] und wählen Sie die Variable über die <b>Eingabehilfe</b> aus.
4	<b>IEC-Objekte.</b> Dieser Teil der Registerkarte listet die IEC-Objekte des Zielgeräts auf, auf die durch die Applikation zugegriffen werden kann (beispielsweise, um einen Bus neu zu starten oder Informationen abzufragen).
5	Der Parameter <b>Buszyklus-Task</b> definiert die Task zur Aktualisierung der E/A-Bilder (%IB und %QB). Diese E/A-Bilder entsprechen dem EtherNet/IP-Request, der an das EtherNet/IP-Zielgerät und die Funktionsfähigkeitsbits gesendet wird. Wählen Sie in der Liste den Eintrag <b>Buszyklus-Task</b> aus: <ul style="list-style-type: none"><li>● <b>Zykluseinstellungen des übergeordneten Busses verwenden</b> (Standardeinstellung). Verwenden Sie die in der Registerkarte SPS-Einstellungen (<i>siehe Seite 91</i>) der Steuerung spezifizierte Task.</li><li>● <b>MAST.</b> Verwenden Sie die MAST-Task (<i>siehe Seite 51</i>).</li><li>● <b>Bewegung.</b> Verwenden Sie die Bewegungs-Task (<i>siehe Seite 44</i>).</li></ul>

---

# Kapitel 11

## Konfiguration der Ethernet-Verbindung

---

### Einführung

In diesem Kapitel wird die Konfiguration der Ethernet-Netzwerkschnittstelle des Modicon LMC078 Motion Controller beschrieben.

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Abschnitte:

Abschnitt	Thema	Seite
11.1	Ethernet-Dienste	172
11.2	Firewallkonfiguration	191

# Abschnitt 11.1

## Ethernet-Dienste

---

### Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Beschreibung	173
Konfiguration der IP-Adresse	175
Modbus TCP Client/Server	180
FTP-Server	182
FTP-Client	185
LMC078 Motion Controller als IOScanner-Slave-Gerät in Modbus TCP-Netzwerken	186

## Beschreibung

### Ethernet-Dienste

Die Steuerung unterstützt die folgenden Dienste:

- Modbus TCP-Server (*siehe Seite 180*)
- Modbus TCP-Client (*siehe Seite 180*)
- WebVisualisation (*siehe Seite 174*)
- FTP-Server (*siehe Seite 182*)
- FTP-Client (*siehe Seite 185*)
- Steuerung als E/A-Scanner-Slave-Gerät in Modbus TCP (*siehe Seite 186*)
- IEC VAR ACCESS (*siehe Seite 174*)

### Ethernet-Protokoll

Die Steuerung unterstützt die folgenden Protokolle:

- SoMachine Protokoll
- IP (Internet Protocol)
- UDP (User Datagram Protocol)
- TCP (Transmission Control Protocol)
- ARP (Address Resolution Protocol)
- ICMP (Internet Control Messaging Protocol)
- IGMP (Internet Group Management Protocol)

### Anschlüsse

Diese Tabelle gibt die maximale Anzahl von Verbindungen an:

Anschlussart	Maximale Anzahl Verbindungen
Modbus-Server	8
Modbus-Client	2
EtherNet/IP-Ziel	64
FTP-Server	4
WebVisualisation	10

Jeder TCP-basierte Server verwaltet seine eigenen Verbindungen.

Wenn ein Client versucht, eine Verbindung herzustellen, die die Abfragegröße überschreiten würde, schließt die Steuerung die älteste Verbindung.

Wenn ein Client eine neuen Verbindung zu öffnen versucht und alle Verbindungen belegt sind (ein Datenaustausch stattfindet), wird die neue Verbindung zurückgewiesen.

Alle Serververbindungen bleiben offen, solange sich die Steuerung in einem der Betriebszustände befindet (RUN, STOP, HALT).

Alle Serververbindungen werden geschlossen, wenn der Server aus dem bzw. in den jeweiligen Betriebszustand (RUN, STOP, HALT), wechselt, außer bei einem Stromausfall (da der Steuerung in diesem Fall keine Zeit bleibt, die Verbindungen zu schließen).

### Verfügbare Dienste

Bei Ethernet-Kommunikation wird der Dienst **IEC VAR ACCESS** von der Steuerung unterstützt. Mit dem Dienst **IEC VAR ACCESS** können Daten zwischen der Steuerung und einer HMI ausgetauscht werden.

Der Dienst **NetWork variables** wird ebenfalls von der Steuerung unterstützt. Über den **NetWork variables**-Dienst können Daten zwischen Steuerungen ausgetauscht werden.

**HINWEIS:** Weitere Informationen finden Sie im SoMachine-Programmierhandbuch.

### WebVisualisation

Die Funktion der WebVisualisation wird in der Online-Hilfe zu SoMachine im Kapitel *Programmieren mit SoMachine/Visualisierung* beschrieben.

## Konfiguration der IP-Adresse

### Einführung

Es gibt verschiedene Methoden, um die IP-Adresse der Steuerung zuzuweisen:

- Adressenzuweisung über den DHCP-Server
- Adressenzuweisung über den BOOTP-Server
- Feste IP-Adresse

Die IP-Adresse kann dynamisch geändert werden:

- über die Registerkarte Steuerungsauswahl (*siehe SoMachine, Programmierhandbuch*) in SoMachine.
- über den **Funktionsbaustein** changelPAddress (*siehe Seite 261*).

**HINWEIS:** Sollten die verwendeten Adressierungsmethoden fehlschlagen, verwendet die Steuerung eine Standard-IP-Adresse (*siehe Seite 178*).

Sie müssen die IP-Adressen sorgfältig verwalten, da jedes Gerät im Netzwerk eine eindeutige Adresse benötigt. Wenn mehrere Geräte dieselbe IP-Adresse besitzen, kann dies ein unbeabsichtigtes Betriebsverhalten Ihres Netzwerks und der zugehörigen Geräte zur Folge haben.

### **WARNUNG**

#### **UNBEABSICHTIGTER GERÄTEBETRIEB**

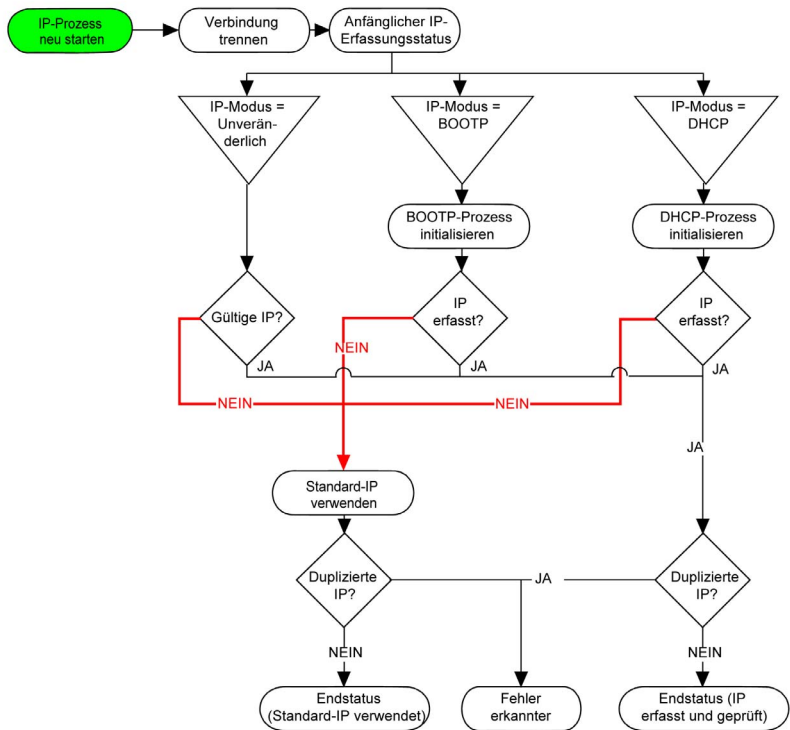
- Vergewissern Sie sich, dass im Netzwerk oder auf der dezentralen Verbindung nur eine Master-Steuerung konfiguriert ist.
- Stellen Sie sicher, dass alle Geräte über eindeutige Adressen verfügen.
- Erfragen Sie Ihre IP-Adresse bei Ihrem Systemadministrator.
- Vergewissern Sie sich, dass die IP-Adresse des Geräts eindeutig ist, bevor Sie das System in Betrieb nehmen.
- Weisen Sie dieselbe IP-Adresse keinem anderen Gerät im Netzwerk zu.
- Aktualisieren Sie die IP-Adresse nach dem Klonen einer Anwendung, die auf eine Ethernet-Kommunikation mit eindeutigen Adressen zurückgreift.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

**HINWEIS:** Stellen Sie sicher, dass der Systemadministrator über alle zugewiesenen IP-Adressen im Netzwerk und im Subnetz Buch führt und dass er über alle durchgeführten Konfigurationsänderungen unterrichtet wird.

**Adressverwaltung**

Dieses Diagramm zeigt die verschiedenen Arten von Adresssystemen für die Steuerung:



**HINWEIS:** Wenn ein Gerät für die Verwendung der Adressierungsmethoden DHCP oder BOOTP programmiert wurde, kann es keine Verbindung zum jeweiligen Server aufnehmen, da die Steuerung die Standard-IP-Adresse verwendet. Es wird diese Anforderung jedoch ständig erneut senden.

Der IP-Prozess wird in den folgenden Fällen neu gestartet:

- Neustart der Steuerung
- Erneuter Anschluss des Ethernet-Kabels
- Anwendungsdownload (falls sich IP-Parameter ändern)
- Nach einer gescheiterten Adressierung wurde der DHCP- bzw. der BOOTP-Server gefunden.

## Konfiguration der Ethernet-Verbindung

Doppelklicken Sie in der **Gerätebaumstruktur** auf **Ethernet**.

The screenshot shows the configuration window for an Ethernet connection. It is divided into two main sections: 'Konfigurierte Parameter' and 'Sicherheitsparameter'.

**Konfigurierte Parameter:**

- Schnittstellenname: EthernetPort0
- Netzwerkname: my\_Device
- IP-Adresse nach DHCP:
- IP-Adresse nach BOOTP:
- Feste IP-Adresse: 
  - IP-Adresse: 0 . 0 . 0 . 0
  - Subnetzmaske: 0 . 0 . 0 . 0
  - Gateway-Adresse: 0 . 0 . 0 . 0
- Ethernet-Protokoll: Ethernet 2
- Übertragungsrage: Auto

**Sicherheitsparameter:**

- SoMachine-Protokoll aktiv
- Modbus-Server aktiv
- FTP-Server aktiv
- Erkennungsprotokoll aktiv
- WebVisualisation-Protokoll aktiv

Die konfigurierten Parameter werden im Folgenden erläutert:

Konfigurierte Parameter	Beschreibung
<b>Schnittstellenname</b>	Name der Netzwerkverbindung
<b>Netzwerkname</b>	Dient als Gerätename zum Abrufen der IP-Adresse über DHCP, max. 16 Zeichen.
<b>IP-Adresse nach DHCP</b>	IP-Adresse wird über DHCP bezogen.
<b>IP-Adresse nach BOOTP</b>	IP-Adresse wird über BOOTP bezogen.

Konfigurierte Parameter	Beschreibung
<b>Feste IP-Adresse</b>	Verwenden Sie zur Konfiguration einer festen IP-Adresse die Registerkarte Steuerungsauswahl ( <i>siehe Seite 90</i> ) im <b>Geräteeditor</b> .
<b>Ethernet-Protokoll</b>	Verwendeter Protokolltyp: Ethernet2
<b>Übertragungsrate</b>	Übertragungsgeschwindigkeit und -richtung werden auf dem Bus automatisch konfiguriert.

### Standard-IP-Adresse

Die Standard-IP-Adresse lautet 190.201.100.100.

Die Standardsubnetzmaske lautet 255.255.255.0.

### Adressklassen

Die IP-Adresse ist mit Folgendem verknüpft:

- mit einem Gerät (dem Host)
- mit dem Netzwerk, an das das Gerät angeschlossen ist

Eine IP-Adresse besteht immer aus einem Code mit 4 Bytes.

Die Verteilung dieser Bytes auf die Netzwerkadresse und die Geräteadresse kann variieren. Diese Verteilung wird durch die Adressklassen definiert.

Die verschiedenen Klassen für IP-Adressen werden in der folgenden Tabelle beschrieben:

Adressklasse	Byte 1			Byte 2	Byte 3	Byte 4
Klasse A	0	Netzwerk-ID		Host-ID		
Klasse B	1	0	Netzwerk-ID		Host-ID	
Klasse C	1	1	0	Netzwerk-ID		Host-ID
Klasse D	1	1	1	0	Multicast-Adresse	
Klasse E	1	1	1	1	0	Adresse für die zukünftige Verwendung reserviert

### Subnetzmaske

Die Subnetzmaske wird verwendet, um mehrere physische Netzwerke mit einer einzigen Netzwerkadresse zu adressieren. Durch die Maske werden das Subnetz und die Geräteadresse in der Host-ID getrennt.

Man erhält die Subnetzadresse, indem die Bits der IP-Adresse, die den Positionen der Maske entsprechen, die 1 enthalten, beibehalten und die restlichen durch 0 ersetzt werden.

Umgekehrt erhält man die Subnetzmaske des Host-Geräts, indem die Bits der IP-Adresse, die den Positionen der Maske entsprechen, die 0 enthalten, beibehalten und die restlichen durch 1 ersetzt werden.

Beispiel für eine Subnetzadresse:

IP-Adresse	192 (11000000)	1 (00000001)	17 (00010001)	11 (00001011)
Subnetzmaske	255 (11111111)	255 (11111111)	240 (11110000)	0 (00000000)
Subnetzadresse	192 (11000000)	1 (00000001)	16 (00010000)	0 (00000000)

**HINWEIS:** Wenn kein Gateway vorhanden ist, kommuniziert das Gerät nicht in seinem Subnetz.

### Gateway-Adresse

Das Gateway ermöglicht, dass eine Nachricht an ein Gerät geleitet wird, das sich nicht im aktuellen Netzwerk befindet.

Wenn kein Gateway vorhanden ist, lautet die Gateway-Adresse 0.0.0.0.

### Sicherheitsparameter

Sicherheitsparameter	Beschreibung
<b>SoMachine-Protokoll aktiv</b>	Über diesen Parameter können Sie das SoMachine-Protokoll an Ethernet-Schnittstellen deaktivieren. Wenn dieser Parameter deaktiviert ist, werden alle SoMachine-Requests von allen Geräten zurückgewiesen, einschließlich Requests über die UDP oder TCP-Verbindung. Das bedeutet, dass ein PC mit SoMachine, ein HMI-Ziel, das Variablen mit dieser Steuerung austauschen möchte, ein OPC-Server und Controller Assistant keine Verbindung über Ethernet herstellen kann.
<b>Modbus-Server aktiv</b>	Über diesen Parameter können Sie den Modbus-Server des Logic Controller deaktivieren. Wenn dieser Parameter deaktiviert ist, werden alle an den Logic Controller gerichteten Modbus-Requests ignoriert.
<b>FTP-Server aktiv</b>	Über diesen Parameter können Sie den FTP-Server des Logic Controller deaktivieren. Wenn dieser Parameter deaktiviert ist, werden FTP-Requests ignoriert.
<b>Erkennungsprotokoll aktiv</b>	Über diesen Parameter können Sie das Discovery-Protokoll deaktivieren. Wenn dieser Parameter deaktiviert ist, werden Discovery-Requests ignoriert.
<b>WebVisualisation-Protokoll aktiv</b>	Über diesen Parameter können Sie die WebVisualisation-Seiten der Steuerung deaktivieren. Wenn dieser Parameter deaktiviert ist, werden alle an das WebVisualisation-Protokoll des Logic Controller gerichteten HTTP-Requests ignoriert.

### Knotenname

Der **Knotenname** wird zum Identifizieren eines Geräts während einer Netzwerkabfrage verwendet. Jedes Gerät in Ihrem Netzwerk muss einen eindeutigen **Knotenname** haben.

Der **Knotenname** einer LMC078-Bewegungssteuerung ist der Steuerungsname, der in der **Gerätebaumstruktur** erscheint.

## Modbus TCP Client/Server

### Einführung

Im Gegensatz zu seriellen Modbus-Verbindungen basiert Modbus TCP nicht auf einer hierarchischen Struktur, sondern auf einem Client/Server-Modell.

Der Modicon LMC078 Motion Controller implementiert sowohl Client- als auch Server-Dienste, so dass er eine Kommunikation mit anderen Steuerungen und E/A-Geräten initiieren und auf Anforderungen von anderen Steuerungen, SCADA, HMIs und anderen Geräten antworten kann.

Der integrierte Ethernet-Port der Steuerung unterstützt Modbus Server ohne jegliche Konfiguration.

Der Modbus-Client/Server ist in der Firmware enthalten und erfordert keinerlei Programmierung durch den Benutzer. Aus diesem Grund kann im Status RUNNING, STOPPED und EMPTY darauf zugegriffen werden.

### Modbus TCP-Client

Der Modbus TCP-Client unterstützt folgende Funktionsbausteine aus der Bibliothek PLCCommunication ohne jegliche Konfiguration:

- ADDM
- READ\_VAR
- SEND\_RECV\_MSG
- SINGLE\_WRITE
- WRITE\_READ\_VAR
- WRITE\_VAR

Weitere Informationen hierzu finden Sie in der Beschreibung der Funktionsbausteine (*siehe SoMachine, Modbus- und ASCII-Lese-/Schreibfunktionen, PLCCommunication-Bibliothekshandbuch*).

### Modbus TCP-Server

Der Modbus Server unterstützt die Modbus-Anforderungen:

Funktionscode Dez (Hex)	Unterfunktion Dez (Hex)	Funktion
1 (1)	–	Digitalausgänge lesen (%Q)
2 (2)	–	Digitaleingänge lesen (%I)
3 (3)	–	Halteregister (%MW) lesen
6 (6)	–	Einzelnes Register schreiben (%MW)
8 (8)	–	Diagnose
15 (F)	–	Mehrere digitale Ausgänge schreiben (%Q)
16 (10)	–	Mehrere Register schreiben (%MW)

Funktionscode Dez (Hex)	Unterfunktion Dez (Hex)	Funktion
23 (17)	–	Mehrere Register lesen/schreiben (%MW)
43 (2B)	14 (E)	Geräteidentifikation lesen

**HINWEIS:** Der integrierte Modbus-Server gewährleistet Zeitkonsistenz nur für ein einziges Wort (2 Byte). Wenn für Ihre Anwendung Zeitkonsistenz für mehrere Wörter gewährleistet werden muss, müssen Sie ein **Modbus TCP-Slave-Gerät** hinzufügen und konfigurieren (*siehe Seite 186*), damit der Inhalt der Puffer %IW und %QW in der zugeordneten IEC-Task (standardmäßig die MAST-Task) zeitkonsistent ist.

### Diagnose-Anforderung

Diese Tabelle enthält die Liste der Datenauswahlcodes:

Datenauswahlcode (hex)	Beschreibung
00	Reserviert
01	Standardnetzwerkdiagnose
02	Ethernet-Port-Diagnose
03	Modbus TCP/Port 502-Diagnose
04	Modbus TCP/Port 502-Verbindungstabelle
05 - 7E	Reserviert für andere öffentliche Codes
7F	Offsets von Datenstrukturen

## FTP-Server

### Einführung

Zur Übertragung von Dateien an und von dem Datenspeicherbereich der Steuerung kann ein beliebiger FTP-Client verwendet werden, der auf einem PC installiert ist, der über Ethernet mit der Steuerung verbunden ist, ohne dass SoMachine installiert sein muss.

**HINWEIS:** Schneider Electric operiert unter den Industriestandards bei der Entwicklung und Implementierung von Steuerungssystemen. Dies beinhaltet ein „Defense-in-Depth-Konzept“ zum Schutz industrieller Steuerungssysteme. Bei diesem Verfahren werden die Steuerungen hinter einer oder mehreren Firewalls platziert, um den Zugriff auf autorisierte Personen und Protokolle zu beschränken.

### **WARNUNG**

#### **UNBERECHTIGTER ZUGRIFF MIT UNBERECHTIGTEM MASCHINENBETRIEB**

- Beurteilen Sie, ob Ihre Betriebsumgebung bzw. Ihre Maschinen mit Ihrer kritischen Infrastruktur verbunden sind. Ist das der Fall, dann ergreifen Sie angemessene Präventivmaßnahmen auf der Basis des Defense-in-Depth-Konzepts, bevor Sie das Automatisierungssystem mit einem Netzwerk verbinden.
- Begrenzen Sie die Anzahl der mit einem Netzwerk verbundenen Geräte auf das strikte Minimum.
- Isolieren Sie Ihr Industrienetzwerk von anderen Netzwerken in Ihrer Firma.
- Schützen Sie alle Netzwerke vor unberechtigtem Zugriff mithilfe von Firewalls, VPNs oder anderen bewährten Schutzmaßnahmen.
- Überwachen Sie die Aktivität in Ihren Systemen.
- Verhindern Sie jeden direkten Zugriff bzw. jede direkte Verbindung von Fachgeräten durch unberechtigte Personen oder nicht autorisierte Vorgänge.
- Stellen Sie einen Wiederherstellungsplan für den Notfall auf. Dazu gehört ebenfalls der Backup Ihrer System- und Prozessdaten.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

**HINWEIS:** Nutzen Sie die sicherheitsbezogenen Befehle (*siehe SoMachine, Menübefehle, Online-Hilfe*), die eine Möglichkeit darstellen, einen Benutzer in der Online-Benutzerverwaltung des Zielgeräts, auf dem Sie momentan angemeldet sind, hinzuzufügen, zu entfernen und zu bearbeiten.

Der FTP-Server ist auch bei leerer Steuerung (keine Benutzeranwendung und keine aktivierten Benutzerrechte) verfügbar.

## FTP-Zugriff

Der Zugriff auf den FTP-Server wird über die in der SPS aktivierten Benutzerrechte gesteuert. Weitere Informationen finden Sie in der Beschreibung der Registerkarte *(siehe Seite 76)* **Benutzer und Gruppen**.

Wenn die Benutzerrechte in der Steuerung nicht aktiviert wurden, werden Sie zur Eingabe eines eindeutigen Benutzernamens und Passworts aufgefordert. Der Standardbenutzername lautet USER, ebenso wie das Standardpasswort USER.

**HINWEIS:** Es wird empfohlen, die Benutzerrechte zu verwenden, um einen globalen Schutz der Steuerung zu gewährleisten.

Aus Gründen der Sicherheit für Ihre Installation müssen Sie das Standardpasswort direkt nach der ersten Anmeldung ändern, wenn die Benutzerrechte in der Steuerung nicht aktiviert wurden. Das Passwort kann mit der Funktion FC\_UserChangePassword *(siehe Modicon LMC078 Motion Controller, Systemfunktionen und -Variablen, PLCSystem Bibliothekshandbuch)* geändert werden.

### **WARNUNG**

#### **SCHUTZ VOR UNBEFUGTEM ZUGRIFF**

- Ändern Sie das Standard-Passwort sofort auf ein neues, sicheres Passwort.
- Geben Sie das Passwort nicht an unbefugtes oder nicht qualifiziertes Personal weiter.
- Deaktivieren Sie den FTP-Server, um unerwünschten und unbefugten Zugriff auf die Daten in Ihrer Anwendung zu verhindern.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

**HINWEIS:** Ein sicheres Passwort ist ein Passwort, das nicht an unbefugtes Personal weitergegeben wurde und das nicht aus persönlichen oder anderweitig offensichtlichen Informationen besteht. Darüber hinaus bietet eine Kombination aus Klein- und Großbuchstaben und Ziffern mehr Sicherheit. Sie sollten ein mindestens 7 Zeichen langes Passwort wählen.

**HINWEIS:** Die einzige Möglichkeit, auf eine Steuerung mit aktivierten Benutzerrechten zuzugreifen, wenn Sie nicht über die entsprechenden Passwörter verfügen, besteht in der Durchführung einer Firmwareaktualisierung. Sie können die Benutzerrechte in der Steuerung auch durch die Ausführung eines Skripts löschen (weitere Informationen hierzu finden Sie im SoMachine-Programmierhandbuch). Dabei wird die vorhandene Anwendung aus dem Speicher der Steuerung entfernt, gleichzeitig aber wird die Möglichkeit des Zugriffs auf die Steuerung wiederhergestellt.

Wenn Sie die Benutzerrechte nicht aktiviert und Ihr Passwort verloren oder vergessen haben, müssen Sie über SoMachine eine direkte Verbindung mit der Steuerung herstellen und einen Ursprungsreset ausführen, um das Standardpasswort wiederherzustellen. Anschließend müssen Sie ein neues, sicheres Passwort einrichten.

**Dateizugriff**

Siehe Dateioorganisation (*siehe Seite 35*).

## FTP-Client

### Einführung

Die Bibliothek FTPRemoteFileHandling stellt die folgenden FTP-Client-Funktionen für die dezentrale Dateiverwaltung zur Verfügung:

- Lesen der Dateien
- Schreiben der Dateien
- Löschen der Dateien
- Auflisten des Inhalts dezentraler Verzeichnisse
- Hinzufügen von Verzeichnissen
- Entfernen von Verzeichnissen

**HINWEIS:** Schneider Electric operiert unter den Industriestandards bei der Entwicklung und Implementierung von Steuerungssystemen. Dies beinhaltet ein „Defense-in-Depth-Konzept“ zum Schutz industrieller Steuerungssysteme. Bei diesem Verfahren werden die Steuerungen hinter einer oder mehreren Firewalls platziert, um den Zugriff auf autorisierte Personen und Protokolle zu beschränken.

### **WARNUNG**

#### **UNBERECHTIGTER ZUGRIFF MIT UNBERECHTIGTEM MASCHINENBETRIEB**

- Beurteilen Sie, ob Ihre Betriebsumgebung bzw. Ihre Maschinen mit Ihrer kritischen Infrastruktur verbunden sind. Ist das der Fall, dann ergreifen Sie angemessene Präventivmaßnahmen auf der Basis des Defense-in-Depth-Konzepts, bevor Sie das Automatisierungssystem mit einem Netzwerk verbinden.
- Begrenzen Sie die Anzahl der mit einem Netzwerk verbundenen Geräte auf das strikte Minimum.
- Isolieren Sie Ihr Industrienetzwerk von anderen Netzwerken in Ihrer Firma.
- Schützen Sie alle Netzwerke vor unberechtigtem Zugriff mithilfe von Firewalls, VPNs oder anderen bewährten Schutzmaßnahmen.
- Überwachen Sie die Aktivität in Ihren Systemen.
- Verhindern Sie jeden direkten Zugriff bzw. jede direkte Verbindung von Fachgeräten durch unberechtigte Personen oder nicht autorisierte Vorgänge.
- Stellen Sie einen Wiederherstellungsplan für den Notfall auf. Dazu gehört ebenfalls der Backup Ihrer System- und Prozessdaten.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

Weitere Informationen finden Sie im FTPRemoteFileHandling-Bibliothekshandbuch.

## LMC078 Motion Controller als IOScanner-Slave-Gerät in Modbus TCP-Netzwerken

### Übersicht

In diesem Abschnitt wird die Konfiguration des LMC078 Motion Controller als **Modbus TCP-Servergerät** beschrieben.

Das **Modbus TCP-Slave-Gerät** richtet einen speziellen E/A-Bereich in der Steuerung ein, auf den über das Modbus TCP-Protokoll zugegriffen werden kann. Dieser Bereich wird verwendet, wenn ein externer E/A-Scanner (Master) Zugriff auf die Objekte %IW und %QW der Steuerung benötigt.

Der Hauptvorteil der Verwendung eines **Modbus TCP-Slave-Geräts** besteht darin, dass die Steuerungsobjekte gesammelt werden und über einen einzigen Modbus-Request auf sie zugegriffen werden kann.

Das Modbus-Slave-Gerät fügt der Steuerung eine weitere Modbus-Serverfunktion hinzu. Auf diesen Server kann die Modbus-Clientanwendung über die konfigurierte Unit-ID (nicht 255) zugreifen. Der integrierte Modbus-Server der Slave-Steuerung erfordert keine Konfiguration; er wird über die Unit-ID = 255 adressiert.

Eingänge/Ausgänge aus Sicht der Slave-Steuerung: Eingänge werden vom Master geschrieben und Ausgänge werden vom Master gelesen.

Das TCP-Slave-Gerät kann eine privilegierte Modbus-Clientanwendung definieren, deren Verbindung nicht abgebrochen wird (die integrierten Modbus-Verbindungen werden unter Umständen beendet, wenn mehr als 8 Verbindungen benötigt werden).

Weitere Informationen zu Modbus TCP finden Sie auf der Website unter [www.odva.org](http://www.odva.org).

### Hinzufügen eines Modbus TCP-Slave-Geräts

Zum Konfigurieren Ihres LMC078 Motion Controller als ein Modbus TCP-Slave-Gerät:

Schritt	Aktion
1	Wählen Sie den Knoten <b>Ethernet-Netzwerk</b> aus und klicken Sie auf die Plus-Schaltfläche. <b>Ergebnis:</b> Das Dialogfeld <b>Gerät hinzufügen</b> wird angezeigt.
2	Wählen Sie <b>Modbus TCP-Slave-Gerät</b> .
3	Klicken Sie auf <b>Gerät hinzufügen</b>
4	Klicken Sie auf <b>Schließen</b> .

## Modbus TCP-Konfiguration

Um das **Modbus TCP-Slave-Gerät** zu konfigurieren, doppelklicken Sie auf **Ethernet-Netzwerk** → **Modbus TCP-Slave-Gerät** in der **Gerätebaumstruktur**.

Das folgende Dialogfeld wird angezeigt:

Element	Beschreibung
<b>Master-IP-Adresse</b>	IP-Adresse des Modbus-Masters Die Verbindungen sind auf dieser Adresse nicht geschlossen.
<b>Timeout</b>	Timeout in 500-ms-Inkrementen. <b>HINWEIS:</b> Das Timeout gilt für die Master-IP-Adresse, sofern sie nicht 0.0.0.0. lautet.
<b>Slave-Anschluss</b>	Modbus-Kommunikationsport (502)
<b>Unit-ID</b>	Sendet die Requests an das Modbus TCP-Slave-Gerät (1 bis 247) anstatt an den integrierten Modbus-Server (255).
<b>Halteregister (%IW)</b>	Anzahl der %IW-Register, die beim Austausch verwendet werden sollen (2 bis 40) (jedes Register umfasst 2 Byte).
<b>Eingangsregister (%QW)</b>	Anzahl der %QW-Register, die beim Austausch verwendet werden sollen (2 bis 40) (jedes Register umfasst 2 Byte).

### Registerkarte ModbusTCP Slave Gerät E/A-Abbild

Die E/A werden aus der Sicht des Masters wie folgt den Modbus-Registern zugeordnet:

- %IW-Register werden von Register 0 bis n-1 zugeordnet und können gelesen/geschrieben werden (n = Anzahl Haltereister, jedes %IW-Register umfasst 2 Byte).
- %QW-Register werden von Register n bis n+m-1 zugeordnet und sind schreibgeschützt (n = Anzahl Eingangsregister, jedes %QW-Register umfasst 2 Byte).





































































Wenn ein **Modbus TCP-Slavegerät** konfiguriert wurde, greifen Modbus-Befehle, die an die Unit-ID (Modbus-Adresse) dieses Geräts gesendet werden, auf die %IW- und %QW-Objekte der Steuerung zu und nicht auf die Modbus-Standardwörter (auf die zugegriffen wird, wenn die Unit-ID = 255). Dies ermöglicht Lese-/Schreibvorgänge durch eine Modbus TCP-IOScanner-Anwendung.

Das **Modbus TCP-Slave-Gerät** antwortet auf eine Untergruppe der Modbus-Befehle und dient dem Austausch von Daten mit dem externen E/A-Scanner. Die folgenden Modbus-Befehle werden vom Modbus TCP-Slave-Gerät unterstützt:

Funktionscode dez. (in Hex-Darstellung)	Funktion	Kommentar
3 (3)	Haltereister lesen	Ermöglicht dem Master das Lesen der %IW- und %QW-Objekte des Geräts.
6 (6)	Einzelnes Register schreiben	Ermöglicht dem Master das Schreiben der %IW-Objekte des Geräts.
16 (10)	Mehrere Register schreiben	Ermöglicht dem Master das Schreiben der %IW-Objekte des Geräts.
23 (17)	Mehrere Register lesen/schreiben	Ermöglicht dem Master das Lesen der %IW- und %QW-Objekte des Geräts und das Schreiben der %IW-Objekte des Geräts
Sonstige	Nicht unterstützt	–

**HINWEIS:** Bei Modbus-Requests, die auf Register oberhalb von n+m-1 zugreifen, wird der Ausnahmecode 02 – ILLEGAL DATA ADDRESS zurückgegeben.

Zur Verknüpfung der E/A mit Variablen wählen Sie die Registerkarte **ModbusTCP Slave Gerät I/O Abbild** aus:

Modbus TCP		ModbusTCP Slave Gerät I/O Abbild		Informationen			
Kanäle							
Variable	Zuordnun	Kanal	Address	Typ	Standardwert	Einheit	Beschreibung
 		Eingänge	%IW0	ARRAY[0 BIS 9] AUS...			Modbus-Halterregister...
  Application.POU.tata		Eingänge[0]	%IW0	WORD			
  iwModbusTCT_Sla...		Eingänge[1]	%IW1	WORD			
  iwModbusTCT_Sla...		Eingänge[2]	%IW2	WORD			
  iwModbusTCT_Sla...		Eingänge[3]	%IW3	WORD			
  iwModbusTCT_Sla...		Eingänge[4]	%IW4	WORD			
  iwModbusTCT_Sla...		Eingänge[5]	%IW5	WORD			
  iwModbusTCT_Sla...		Eingänge[6]	%IW6	WORD			
  iwModbusTCT_Sla...		Eingänge[7]	%IW7	WORD			
  iwModbusTCT_Sla...		Eingänge[8]	%IW8	WORD			
  iwModbusTCT_Sla...		Eingänge[9]	%IW9	WORD			
 		Ausgänge	%QW0	ARRAY[0 BIS 9] AUS...			Modbus-Eingangsregister...
  qwModbusTCP_Sl...		Ausgänge[0]	%QW0	WORD			
  qwModbusTCP_Sl...		Ausgänge[1]	%QW1	WORD			
  qwModbusTCP_Sl...		Ausgänge[2]	%QW2	WORD			
  qwModbusTCP_Sl...		Ausgänge[3]	%QW3	WORD			
  qwModbusTCP_Sl...		Ausgänge[4]	%QW4	WORD			
  qwModbusTCP_Sl...		Ausgänge[5]	%QW5	WORD			
  qwModbusTCP_Sl...		Ausgänge[6]	%QW6	WORD			
  qwModbusTCP_Sl...		Ausgänge[7]	%QW7	WORD			
  qwModbusTCP_Sl...		Ausgänge[8]	%QW8	WORD			
  qwModbusTCP_Sl...		Ausgänge[9]	%QW9	WORD			
<input type="button" value="Mapping zurücksetzen"/> <input checked="" type="checkbox"/> Variablen immer aktualisieren							
IEC-Objekte							
Variable	Zuordnung	Typ					
 Modbus TCP_Slave_De		IoDrvModbusTCPSlave					
 = Neue Variable erzeugen  = Auf existierende Variable abbilden							
Buszyklus-Optionen Buszyklus-Task <input type="text" value="Zykluseinstellungen des übergeordneten Busses verwenden"/>							

Kanal		Typ	Beschreibung
Eingang	IW0	WORD	Halteregister 0
	...	...	...
	IWx	WORD	Halteregister x
Ausgang	QW0	WORD	Eingangsregister 0
	...	...	...
	QWy	WORD	Eingangsregister y

Die Anzahl der Wörter ist von den Parametern **Halteregister (%IW)** und **Eingangsregister (%QW)** auf der Registerkarte **Modbus TCP** abhängig.

**HINWEIS:** Ausgang bedeutet AUSGANG der Ursprungssteuerung (= %IW für die Steuerung). Eingang bedeutet EINGANG der Ursprungssteuerung (= %QW für die Steuerung).

**HINWEIS:** Das **Modbus TCP-Slave-Gerät** aktualisiert die %IW- und %QW-Register als eine zeitkonsistente Einheit, die mit den IEC-Tasks (standardmäßig die MAST-Task) synchronisiert wird. Im Gegensatz dazu gewährleistet der Modbus TCP-Server Zeitkonsistenz nur für 1 Wort (2 Byte). Wenn für Ihre Anwendung Zeitkonsistenz für mehr als 1 Wort (2 Byte) erforderlich ist, verwenden Sie das **Modbus TCP-Slave-Gerät**.

### Buszyklus-Optionen

Wählen Sie die **Buszyklus-Task** aus, um Folgendes verwenden zu können:

- **Zykluseinstellungen des übergeordneten Busses verwenden** (Standardeinstellung)
- **MAST**
- **Eine vorhandene Task des Projekts**

**HINWEIS:** Im E/A-Abbild-Editor des Geräts, das das **Modbus TCP-Slave-Gerät** enthält, ist der entsprechende Parameter **Buszyklus-Task** vorhanden. Dieser Parameter verweist auf die Task, die für die Aktualisierung der %IW- und %QW-Register zuständig ist.

---

# Abschnitt 11.2

## Firewallkonfiguration

---

### Einführung

In diesem Abschnitt wird die Konfiguration der Firewall des Modicon LMC078 Motion Controller beschrieben.

### Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Einführung	192
Verhalten der Firewall	194
Skriptbefehle für die Firewall	195
Skriptdateien	199

## Einführung

### Firewall – Beschreibung

Im Allgemeinen dienen Firewalls dem Schutz der Netzwerksicherheitszone, indem Sie jeden unbefugten Zugriff verhindern und ausschließlich autorisierten Zugriff gewähren. Bei einer Firewall handelt es sich um ein Gerät bzw. um eine Gruppe von Geräten, die für die Genehmigung, Verweigerung, Verschlüsselung, Entschlüsselung oder Umleitung über Proxy des Datenverkehrs zwischen verschiedenen Sicherheitszonen auf der Grundlage einer Reihe von Regeln und anderen Kriterien konfiguriert wurden.

Geräte zur Prozesssteuerung und Maschinen zur Hochgeschwindigkeitsproduktion benötigen einen hohen Datendurchsatz und tolerieren in vielen Fällen die Latenz nicht, die bei einer aggressiven Sicherheitsstrategie innerhalb des Steuerungsnetzwerks gegeben ist. Aus diesem Grund spielen Firewalls eine bedeutende Rolle in jeder Sicherheitsstrategie, da sie bestimmte Schutzniveaus am Netzwerkperimeter bereitstellen. Firewalls sind ein wichtiger Bestandteil einer globalen Strategie auf Systemebene.

**HINWEIS:** Schneider Electric operiert unter den Industriestandards bei der Entwicklung und Implementierung von Steuerungssystemen. Dies beinhaltet ein „Defense-in-Depth-Konzept“ zum Schutz industrieller Steuerungssysteme. Bei diesem Verfahren werden die Steuerungen hinter einer oder mehreren Firewalls platziert, um den Zugriff auf autorisierte Personen und Protokolle zu beschränken.

### **WARNUNG**

#### **UNBERECHTIGTER ZUGRIFF MIT UNBERECHTIGTEM MASCHINENBETRIEB**

- Beurteilen Sie, ob Ihre Betriebsumgebung bzw. Ihre Maschinen mit Ihrer kritischen Infrastruktur verbunden sind. Ist das der Fall, dann ergreifen Sie angemessene Präventivmaßnahmen auf der Basis des Defense-in-Depth-Konzepts, bevor Sie das Automatisierungssystem mit einem Netzwerk verbinden.
- Begrenzen Sie die Anzahl der mit einem Netzwerk verbundenen Geräte auf das strikte Minimum.
- Isolieren Sie Ihr Industrienetzwerk von anderen Netzwerken in Ihrer Firma.
- Schützen Sie alle Netzwerke vor unberechtigtem Zugriff mithilfe von Firewalls, VPNs oder anderen bewährten Schutzmaßnahmen.
- Überwachen Sie die Aktivität in Ihren Systemen.
- Verhindern Sie jeden direkten Zugriff bzw. jede direkte Verbindung von Fachgeräten durch unberechtigte Personen oder nicht autorisierte Vorgänge.
- Stellen Sie einen Wiederherstellungsplan für den Notfall auf. Dazu gehört ebenfalls der Backup Ihrer System- und Prozessdaten.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

## Firewallkonfiguration

Es gibt zwei Möglichkeiten, die Konfiguration der Steuerungsfirewall zu verwalten:

- Statische Konfiguration
- Anwendungseinstellungen

Für die statische Konfiguration wird eine Skriptdatei verwendet.

## Statische Konfiguration

Die statische Konfiguration wird geladen, wenn die Steuerung gestartet wird.

Die Firewall der Steuerung kann statisch konfiguriert werden, indem eine auf der Steuerung befindliche Standardskriptdatei verwaltet wird. Speicherort dieser Datei:

*/Usr/Cfg/FirewallDefault.cmd*

## Anwendungseinstellungen

Siehe Ethernet-Konfiguration (*siehe Seite 177*).

## Verhalten der Firewall

### Einführung

Die Firewallkonfiguration richtet sich nach der Aktion, die an der Steuerung durchgeführt wird, und nach dem Anfangszustand der Konfiguration. Es gibt 4 mögliche Anfangszustände:

- In der Steuerung ist keine Standardskriptdatei vorhanden.
- Eine gültige Datei ist vorhanden.
- Eine ungültige Skriptdatei ist vorhanden.
- Es ist keine Standardskriptdatei vorhanden und die Anwendung hat die Firewall konfiguriert.

### Keine Standardskriptdatei

Aktion	Folge
Steuerung wird gestartet	Firewall wird nicht konfiguriert. Es ist kein Schutz aktiviert.
Anwendung wird heruntergeladen	Firewall wird entsprechend den Anwendungseinstellungen konfiguriert.

### Standardskriptdatei vorhanden

Aktion	Folge
Steuerung wird gestartet	Firewall wird entsprechend der Standardskriptdatei konfiguriert.
Anwendung wird heruntergeladen	Die gesamte Konfiguration der Anwendung wird ignoriert. Firewall wird entsprechend der Standardskriptdatei konfiguriert.

### Inkorrekte Standardskriptdatei vorhanden

Aktion	Folge
Steuerung wird gestartet	Firewall wird nicht konfiguriert. Es ist kein Schutz aktiviert.
Anwendung wird heruntergeladen	Firewall wird entsprechend den Anwendungseinstellungen konfiguriert.

### Anwendungseinstellungen ohne Standardskriptdatei

Aktion	Folge
Steuerung wird gestartet	Firewall wird entsprechend den Anwendungseinstellungen konfiguriert.
Anwendung wird heruntergeladen	Die gesamte Konfiguration der vorherigen Anwendung wird gelöscht. Firewall wird entsprechend den neuen Anwendungseinstellungen konfiguriert.

## Skriptbefehle für die Firewall

### Überblick

In diesem Abschnitt wird beschrieben, wie Skriptdateien (Standardskriptdatei oder dynamische Skriptdatei) geschrieben werden müssen, damit sie beim Start der Steuerung bzw. bei einem bestimmten ausgelösten Befehl korrekt ausgeführt werden können.

### Syntax einer Skriptdatei

Die Syntax von Skriptdateien werden in den Richtlinien für die Skriptsyntax (*siehe Seite 199*) beschrieben.

### Allgemeine Firewallbefehle

Für die Verwaltung der Ethernet-Firewall des LMC078 Motion Controller sind folgende Befehle verfügbar:

Befehl	Beschreibung
<code>FireWall Enable</code>	Blockiert die Frames von den Ethernet-Schnittstellen. Wenn keiner bestimmte IP-Adresse entsprechende Berechtigungen zugewiesen werden, ist keine Datenübertragung über die Ethernet-Schnittstellen möglich. <b>HINWEIS:</b> Standardmäßig werden die Frames bei aktivierter Firewall abgewiesen.
<code>FireWall Disable</code>	IP-Adressen sind zum Zugriff auf die Steuerung über die Ethernet-Schnittstellen berechtigt.
<code>FireWall Eth1 Default Allow</code>	Frames werden von der Steuerung angenommen.
<code>FireWall Eth1 Default Reject</code>	Frames werden von der Steuerung abgewiesen. <b>HINWEIS:</b> Wenn diese Zeile nicht vorhanden ist, wird standardmäßig der Befehl <code>FireWall Eth1 Default Reject</code> verwendet.

## Spezifische Firewallbefehle

Für die Konfiguration der Firewallregeln für bestimmte Ports und Adressen sind folgende Befehle verfügbar:

Befehl	Bereich	Beschreibung
Firewall Eth1 Allow IP ..... .....	* = 0 bis 255	Die Frames von den genannten IP-Adressen sind für alle Portnummern und Porttypen zugelassen.
Firewall Eth1 Reject IP ..... .....	* = 0 bis 255	Die Frames von den genannten IP-Adressen werden für alle Portnummern und Porttypen abgewiesen.
Firewall Eth1 Allow IPs ..... to ..... .....	* = 0 bis 255	Die Frames von den IP-Adressen im genannten Bereich sind für alle Portnummern und Porttypen zugelassen.
Firewall Eth1 Reject IPs ..... to ..... .....	* = 0 bis 255	Die Frames von den IP-Adressen im genannten Bereich werden für alle Portnummern und Porttypen abgewiesen.
Firewall Eth1 Allow port_type port Y	Y = (Zielportnummern (siehe Seite 198))	Die Frames mit der genannten Zielportnummer sind zugelassen.
Firewall Eth1 Reject port_type port Y	Y = (Zielportnummern (siehe Seite 198))	Die Frames mit der genannten Zielportnummer sind zugelassen.
Firewall Eth1 Allow port_type ports Y1 to Y2	Y = (Zielportnummern (siehe Seite 198))	Die Frames mit einer Zielportnummer im genannten Bereich sind zugelassen.
Firewall Eth1 Reject port_type ports Y1 to Y2	Y = (Zielportnummern (siehe Seite 198))	Die Frames mit einer Zielportnummer im genannten Bereich werden abgewiesen.
Firewall Eth1 Allow IP ..... on port_type port Y	* = 0 bis 255 Y = (Zielportnummern (siehe Seite 198))	Die Frames von der genannten IP-Adresse und mit der genannten Zielportnummer sind zugelassen.
Firewall Eth1 Reject IP ..... on port_type port Y	* = 0 bis 255 Y = (Zielportnummern (siehe Seite 198))	Die Frames von der genannten IP-Adresse und mit der genannten Zielportnummer werden abgewiesen.
Firewall Eth1 Allow IP ..... on port_type ports Y1 to Y2	* = 0 bis 255 Y = (Zielportnummern (siehe Seite 198))	Die Frames von der genannten IP-Adresse und mit einer Zielportnummer im genannten Bereich sind zugelassen.
Firewall Eth1 Reject IP ..... on port_type ports Y1 to Y2	* = 0 bis 255 Y = (Zielportnummern (siehe Seite 198))	Die Frames von der genannten IP-Adresse und mit einer Zielportnummer im genannten Bereich werden abgewiesen.

Befehl	Bereich	Beschreibung
<pre>Firewall Eth1 Allow IPs •1.1.1.1 to •2.2.2.2 on port_type port Y</pre>	<ul style="list-style-type: none"> <li>• = 0 bis 255</li> <li>Y = (Zielportnummern <i>(siehe Seite 198)</i>)</li> </ul>	Die Frames von einer IP-Adresse im genannten Bereich und mit der genannten Zielportnummer werden abgewiesen.
<pre>Firewall Eth1 Reject IPs •1.1.1.1 to •2.2.2.2 on port_type port Y</pre>	<ul style="list-style-type: none"> <li>• = 0 bis 255</li> <li>Y = (Zielportnummern <i>(siehe Seite 198)</i>)</li> </ul>	Die Frames von einer IP-Adresse im genannten Bereich und mit der genannten Zielportnummer werden abgewiesen.
<pre>Firewall Eth1 Allow IPs •1.1.1.1 to •2.2.2.2 on port_type ports Y1 to Y2</pre>	<ul style="list-style-type: none"> <li>• = 0 bis 255</li> <li>Y = (Zielportnummern <i>(siehe Seite 198)</i>)</li> </ul>	Die Frames von einer IP-Adresse im genannten Bereich und mit einer Zielportnummer im genannten Bereich sind zugelassen.
<pre>Firewall Eth1 Reject IPs •1.1.1.1 to •2.2.2.2 on port_type ports Y1 to Y2</pre>	<ul style="list-style-type: none"> <li>• = 0 bis 255</li> <li>Y = (Zielportnummern <i>(siehe Seite 198)</i>)</li> </ul>	Die Frames von einer IP-Adresse im genannten Bereich und mit einer Zielportnummer im genannten Bereich werden abgewiesen.
<pre>Firewall Eth1 Allow MAC ••:••:••:••:••:••</pre>	<ul style="list-style-type: none"> <li>• = 0..F</li> </ul>	Die Frames von der genannten MAC-Adresse ••:••:~••:~••:~••:~•• sind zugelassen.
<pre>Firewall Eth1 Reject MAC ••:~••:~••:~••:~••:~••</pre>	<ul style="list-style-type: none"> <li>• = 0..F</li> </ul>	Die Frames mit der genannten MAC-Adresse ••:~••:~••:~••:~••:~•• werden abgewiesen.

### Beispiel für ein Skript

```
; Enable firewall on Ethernet 1. All frames are rejected;
FireWall Enable;

; Block all Modbus Requests on all IP address
Firewall Eth1 Reject tcp port 502;

; Allow FTP active connection for IP address 85.16.0.17
Firewall Eth1 Allow IP 85.16.0.17 on tcp port 20 to 21;
```

## Verwendete Ports

Protokoll	Zielportnummern
SoMachine	UDP 1740, 1741, 1742, 1743 TCP 1105
FTP	TCP 21, 20
HTTP	TCP 80
Modbus	TCP 502
Discovery	UDP 27126, 27127
NVL	UDP-Standardwert: 1202
EtherNet/IP	UDP 2222 TCP 44818

## Skriptdateien

### Überblick

Im Folgenden wird beschrieben, wie Skriptdateien zur Konfiguration der Ethernet-Firewall (*siehe Seite 195*) geschrieben werden müssen.

### Richtlinien für die Skriptsyntax

Jede Zeile eines Befehls im Skript endet mit einem ";".

Wenn eine Zeile mit einem ";" beginnt, handelt es sich um einen Kommentar.

Eine Skriptdatei darf maximal 50 Zeilen lang sein.

Bei der Syntax wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Wenn die Syntax in einer Skriptdatei nicht eingehalten wird, wird die Skriptdatei nicht ausgeführt. Das bedeutet, dass die Firewallkonfiguration im vorherigen Zustand verbleibt.

**HINWEIS:** Wenn die Skriptdatei nicht ausgeführt wird, wird eine Protokolldatei generiert. Speicherort der Protokolldatei in der Steuerung: */usr/Syslog/FWLog.txt*.



---

# Kapitel 12

## CANopen-Konfiguration

---

### Einführung

In diesem Kapitel wird die Konfiguration der CAN-Schnittstelle in der Steuerung beschrieben.

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
CANopen-Schnittstellenkonfiguration	202
Konfiguration des CANopen-Masters	203
Konfiguration eines CANopen-Slaves	205

## CANopen-Schnittstellenkonfiguration

### Hinzufügen des CAN-Bus-Knotens

Wenn Sie Ihrer Steuerung den Knoten **CAN\_Layer** hinzufügen möchten, wählen Sie **CANbus** im **Hardwarekatalog** aus, ziehen Sie ihn auf die **Gerätebaumstruktur**, und legen Sie ihn auf dem Steuerungsknoten ab.

Weitere Informationen zum Hinzufügen von Geräten in einem Projekt finden Sie unter:

- Verwenden der Methode Drag&Dop (*siehe SoMachine, Programmierhandbuch*) (Ziehen und Ablegen)
- Verwenden der Kontextmenüs oder Plus-Schaltflächen (*siehe SoMachine, Programmierhandbuch*)

### CAN-Buskonfiguration

So konfigurieren Sie den **CAN-Bus** Ihrer Steuerung:

Schritt	Aktion
1	Doppelklicken Sie in der <b>Gerätebaumstruktur</b> auf <b>CAN_Layer</b> .
2	Konfigurieren Sie die Baudrate (Standardwert: 250000 Bit/s).

### Hinzufügen eines CANopen-Managers

Die Steuerung unterstützt die folgenden CANopen-Manager:

- **CANopen\_Manager** für den als CANopen-Master konfigurierten CAN-Port
- **CAN\_Local\_Device** für den als CANopen-Slave konfigurierten CAN-Port

Um der Steuerung einen CANopen-Manager hinzuzufügen, wählen Sie Folgendes im **Hardwarekatalog** aus:

- Für einen CANopen-Master: **CANopen\_Manager**
- Für einen CANopen-Slave: **CAN\_Local\_Device**

Ziehen Sie das ausgewählte Element in die **Gerätebaumstruktur** und legen Sie es auf einem der optisch hervorgehobenen Knoten ab.

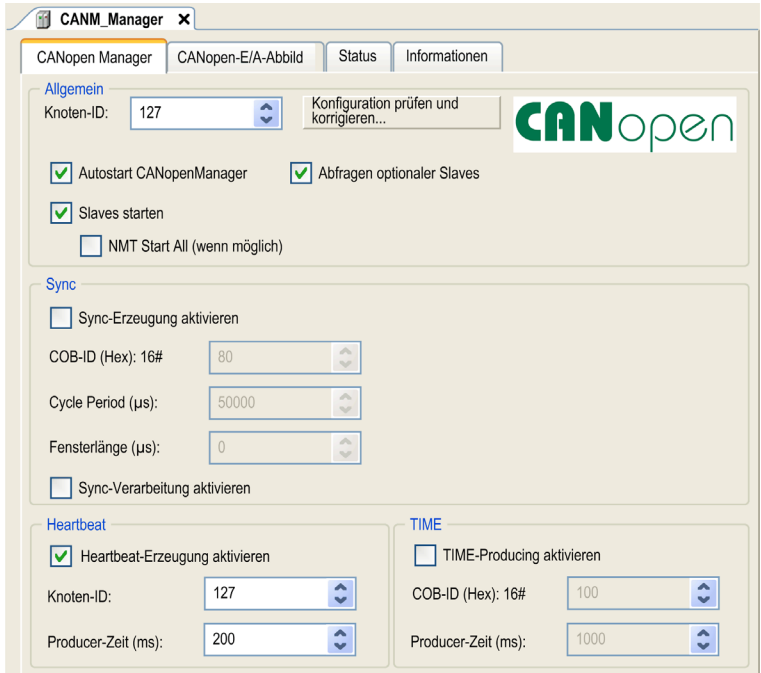
Weitere Informationen zum Hinzufügen von Geräten in einem Projekt finden Sie unter:

- Verwenden der Methode Drag&Dop (*siehe SoMachine, Programmierhandbuch*) (Ziehen und Ablegen)
- Verwenden der Kontextmenüs oder Plus-Schaltflächen (*siehe SoMachine, Programmierhandbuch*)

## Konfiguration des CANopen-Masters

### Konfiguration des CANopen-Managers

Gehen Sie wie folgt vor, um den **CANopen\_Manager** zu konfigurieren:

Schritt	Aktion
1	<p>Doppelklicken Sie auf <b>CANopen_Manager</b> in der <b>Gerätebaumstruktur</b>.  <b>Ergebnis:</b> Das Konfigurationsfenster <b>CANopen Manager</b> wird angezeigt:</p> 
2	<p>Weitere Informationen zur CANopen-Manager-Konfiguration finden Sie in der Online-Hilfe zu SoMachine im Kapitel <i>Programmieren mit SoMachine/Geräteeditoren/CANopen-Manager</i>.</p>

### Hinzufügen eines CANopen-Geräts

Weitere Informationen zum Hinzufügen eines Kommunikations-Managers und zum Hinzufügen von Slave-Geräten zu einem Kommunikations-Manager finden Sie im SoMachine-Programmierhandbuch.

**CANopen-Betriebseinschränkungen**

Für den Modicon LMC078 Motion Controller-CANopen-Master gelten folgende Betriebseinschränkungen:

Maximale Anzahl von Slave-Geräten	63
Maximale Anzahl an von Empfangs-PDO (RPDO)	252
Maximale Anzahl von Sende-PDO (TPDO)	252

** WARNUNG****UNBEABSICHTIGTER GERÄTEBETRIEB**

- Schließen Sie nicht mehr als 63 CANopen-Slave-Geräte an die Steuerung an.
- Programmieren Sie Ihre Anwendung für eine Verwendung von maximal 252 Sende-PDO (TPDO).
- Programmieren Sie Ihre Anwendung für eine Verwendung von maximal 252 Empfangs-PDO (RPDO).

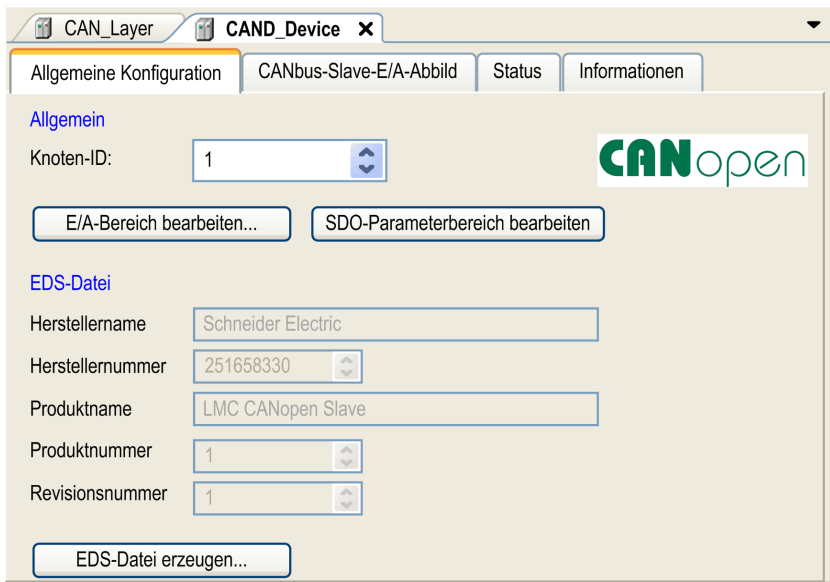
**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

## Konfiguration eines CANopen-Slaves

### Konfiguration eines CANopen-Slaves

Gehen Sie wie folgt vor, um die Steuerung als CANopen-Slave zu konfigurieren:

Schritt	Aktion
1	Wählen Sie den Steuerungsknoten in der <b>Gerätebaumstruktur</b> aus, klicken Sie auf das grüne Pluszeichen des Knotens, oder klicken Sie mit der rechten Maustaste auf den Knoten, und wählen Sie aus dem Kontextmenü <b>Gerät hinzufügen</b> aus. <b>Ergebnis:</b> Das Dialogfeld <b>Gerät hinzufügen</b> wird geöffnet.
2	Wählen Sie im Dialogfeld <b>Gerät hinzufügen CANbus</b> aus und klicken sie auf die Schaltfläche <b>Gerät hinzufügen</b> . <b>Ergebnis:</b> Das Gerät wird der Steuerung hinzugefügt.
3	Klicken Sie auf die Schaltfläche <b>Schließen</b> , um das Dialogfeld <b>Gerät hinzufügen</b> zu schließen.
4	Wählen Sie den <b>CAN_Layer</b> -Knoten in der <b>Gerätebaumstruktur</b> aus, klicken Sie auf das grüne Pluszeichen des Knotens, oder klicken Sie mit der rechten Maustaste auf den Knoten, und wählen Sie aus dem Kontextmenü <b>Gerät hinzufügen</b> aus.
5	Wählen Sie im Dialogfeld <b>Gerät hinzufügen CAN Lokales_Gerät</b> aus und klicken sie auf die Schaltfläche <b>Gerät hinzufügen</b> . <b>Ergebnis:</b> Das Gerät wird dem Knoten <b>CAN_Layer</b> hinzugefügt.
6	Klicken Sie auf die Schaltfläche <b>Schließen</b> , um das Dialogfeld <b>Gerät hinzufügen</b> zu schließen.

Schritt	Aktion
7	<p>Doppelklicken Sie auf <b>CAND_Device</b> in der <b>Gerätebaumstruktur</b>. <b>Ergebnis:</b> Das Konfigurationsfenster <b>CAND_Device</b> wird angezeigt:</p> 
8	<p>Die Konfiguration des CANopen-Managers wird in der Online-Hilfe zu SoMachine im Kapitel <i>Programmieren mit SoMachine/Geräteeditoren/CANbus-Slave-Gerät</i> beschrieben.</p>

---

# Kapitel 13

## SercosKonfiguration

---

### Einführung

In diesem Kapitel wird die Konfiguration der Sercos-Schnittstelle von einem Modicon LMC078 Motion Controller beschrieben.

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Überblick über den Sercos-Standard	208
Sercos Schnittstellenkonfiguration	211
Sercos-Geräte	215
Editor für die Geräteadressierung	216
Lexium LXM32S-Antriebskonfiguration	220
TM5NS31 Sercos Interface Module	223
Sercos Fehlercodes	224

## Überblick über den Sercos-Standard

### Einführung

Die Sercos-Schnittstelle ist eine standardisierte Schnittstelle (IEC 6149) für die Echtzeitkommunikation zwischen Steuerungen, Antrieben und E/A-Antrieben.

Sie beschreibt die international standardisierte digitale Schnittstelle für die Kommunikation zwischen einer Steuereinheit und zugewiesenen Servoantrieben, die in einem Netzwerk ein Bewegungssteuerungssystem bilden. Sie definiert die Standardisierung von Betriebsdaten, Parametern und Skalierungen für Geräte mit mehreren Antrieben, die in Betriebsmodi auf Grundlage von Drehmoment, Geschwindigkeit oder Position betrieben werden können.

Die Hauptfunktionen der Sercos-Schnittstelle lauten:

- Ringtopologie (Redundanz)
- Master/Slave-System
- Baudrate: 100 MBaud
- Minimale Synchronisierungszeit von 1 ms (8 Achsen), 2 ms (16 Achsen) oder 4 ms (24 Achsen)
- Synchronisierung (Jitter < 1 µs)

### Datenaustausch

Die Kommunikation mit der Sercos-Schnittstelle ist in zwei Typen unterteilt:

- Zyklische Kommunikation mit Telegrammen:

Die zyklische Kommunikation wird für den Austausch von Echtzeitdaten verwendet (z. B. Position) und wird einmal pro Kommunikationszyklus ausgeführt (*CycleTime*). Bestimmte festgelegte Daten werden in jedem Zyklus von der Steuerung an alle Antriebe und von allen Antrieben an die Steuerung übertragen.

- Nicht-zyklische Kommunikation mit Funktionsbausteinen der LMC078\_Sercos3-Bibliothek (*siehe Seite 281*).

Nicht-zyklische Kommunikation wird verwendet, um Daten wie Parameter für die Kommunikationskonfiguration, die Antriebsparameter, Zustände usw. auszutauschen, für die die Zeit keine große Rolle spielt. Die Steuerung steuert die nicht-zyklische Kommunikation. Über diesen Kanal können alle Parameter im System kontaktiert werden, auch zyklisch konfigurierte Parameter.

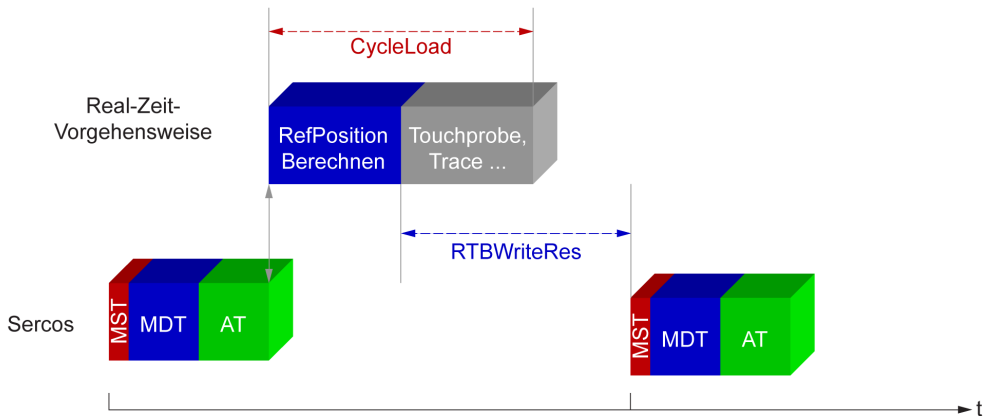
**HINWEIS:** Die zwei Kommunikationstypen können gleichzeitig verwendet werden.

## Zyklischer Datenaustausch

Der Austausch von Informationen zwischen der Bewegungssteuerung (Sercos-Master) und den Servoantrieben (Slaves) geschieht über eine Meldungsstruktur, die als Telegramm bezeichnet wird. Dabei handelt es sich um drei Telegramme, die in IEC 61491 definiert sind:

- MST (Master Synchronization Telegram): Ein MST-Telegramm wird zu Beginn jedes Übertragungszyklus gesendet, um die Zykluszeit zu synchronisieren.
- MDT (Master Data Telegram): Ein MDT-Telegramm wird während jedes Übertragungszyklus gesendet, um Daten (Befehlswerte) an die Servoantriebe (Slaves) zu übertragen.
- AT (Acknowledge Telegram): AT-Telegramme werden von den Slaves an den Master gesendet (Feedbackwerte).

In dieser Abbildung werden die drei Telegrammtypen dargestellt:



In diesem Thema werden nur die Telegrammtypen MDT und AT besprochen. Im Folgenden wird eine allgemeine Telegrammstruktur gezeigt:

Telegramm Begrenzer	Adresse Feld	Konfigurierbares Datenfeld	Frame- Überprüfung Sequenz	Telegramm Begrenzer
------------------------	-----------------	----------------------------	----------------------------------	------------------------

Das administrative Segment des Telegramms, das die Telegrammsprungmarke (Start), das Adressfeld, die Frame-Prüfsequenz und die Telegrammsprungmarke (Ende) enthält, ist für die Übertragung aller Telegramme erforderlich.

Innerhalb des Telegramms werden während jedes Kommunikationszyklus Echtzeitdaten (Betriebsdaten) im Feld für konfigurierbare Daten übertragen. Die Spezifikation dieser Daten wird von einer IDN (Identifizierungsnummer) bereitgestellt.

Über den Sercos kann der Verarbeitungszyklus der Steuerung mit dem Datenaustausch und dem Steuerungszyklus des Antriebs synchronisiert werden. Deshalb gibt es keine Interferenz zwischen diesen individuellen Zyklen, und die Steuerungsschleifen haben eine minimale, konstante Verlustzeit. Zudem wird der neue Referenzwert in allen Antrieben zur gleichen Zeit wirksam, und alle Bus-Slaves erfassen die gemessenen Werte, die sie gleichzeitig als Ist-Werte an die Steuerung weiterleiten.

### Beschreibung der IDN

IEC 61491 weist allen Betriebsdaten in einem Sercos-Antrieb Identifizierungsnummern (IDNs) zu. Betriebsdaten umfassen Parameter, Schnittstellenprozedurbefehle und Befehls- und Feedbackwerte.

Es gibt zwei IDN-Kategorien:

- Standard-IDNs (S): Diese IDNs werden vom Sercos-Standard IEC 61491 definiert. Standard-IDNs verhalten sich, falls sie von einem Sercos-Antrieb unterstützt werden, immer gleich, unabhängig vom Antriebshersteller.
- Proprietäre IDNs (P): Diese IDNs sind für produktspezifische Daten reserviert, die von den Herstellern von Steuereinheiten und Servoantrieben definiert werden können.

**HINWEIS:** Weitere Informationen zu den in den LXM32S-Antrieben implementierten Sercos-IDNs finden Sie im Produkthandbuch zu Lexium LXM32S.

Bei IDNs handelt es sich üblicherweise um 16-Bit- oder 32-Bit-Binärparameter eines Sercos-Antriebs. IDNs werden wie folgt identifiziert:

- S-0-0047.0.0 Standard-IDN: Positionsbefehlswert
- P-0-3017.0.12 Vom Hersteller definierte IDN: Aktuelle Beschränkung

Die -0- bezieht sich auf Parametersätze. Viele Sercos-Antriebe (einschließlich Lexium) unterstützen keine Parametersätze.

## Sercos Schnittstellenkonfiguration

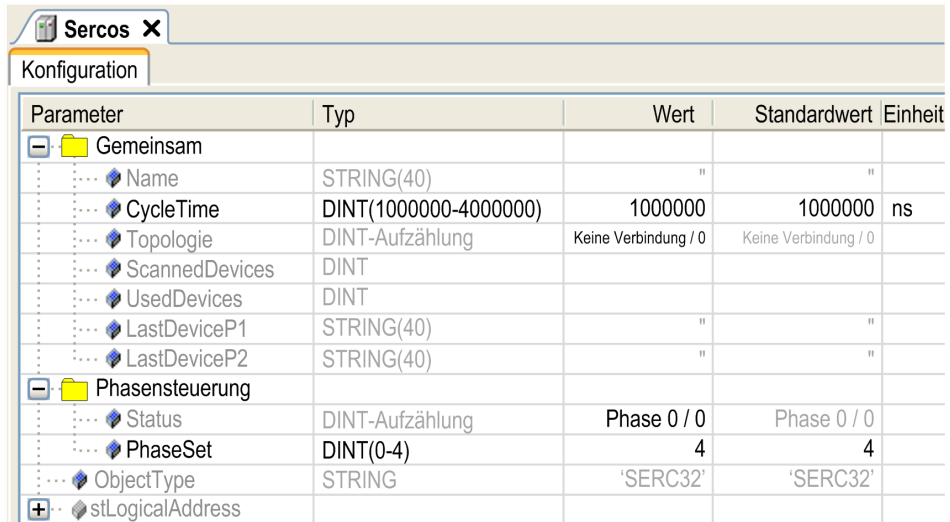
### Einführung

Im Sercos-Konfigurationsfenster können Sie die Sercos-Schnittstellenparameter konfigurieren und anzeigen.

### Sercos-Schnittstellenkonfiguration

Um auf das Sercos-Konfigurationsfenster zuzugreifen, doppelklicken Sie in der **Gerätebaumstruktur** auf **Sercos III**.

Das unten dargestellte Fenster **Konfiguration** wird angezeigt:



Parameter	Typ	Wert	Standardwert	Einheit
Gemeinsam				
Name	STRING(40)	"	"	
CycleTime	DINT(1000000-4000000)	1000000	1000000	ns
Topologie	DINT-Aufzählung	Keine Verbindung / 0	Keine Verbindung / 0	
ScannedDevices	DINT			
UsedDevices	DINT			
LastDeviceP1	STRING(40)	"	"	
LastDeviceP2	STRING(40)	"	"	
Phasensteuerung				
Status	DINT-Aufzählung	Phase 0 / 0	Phase 0 / 0	
PhaseSet	DINT(0-4)	4	4	
ObjectType	STRING	'SERC32'	'SERC32'	
stLogicalAddress				

In der folgenden Tabelle werden die Parameter der Sercos-Schnittstelle beschrieben.

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
<b>Gemeinsam</b>						
Name	R/W(*)	EF	STRING	"	"	Symbolischer Name des Konfigurationsobjekts.
CycleTime	R/W	ER	DINT	1000000 2000000 4000000	1000000	Definiert die Sercos-Buszykluszeit in ns. Die CycleTime kann auf 1 ms, 2 ms oder 4 ms gesetzt werden. Wenn die CycleTime geändert wird, muss die Steuerung zurückgesetzt werden.

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
Topology	R	AF	DINT Enum	no link / 0 line P1 / 1 line P2 / 2 double line / 3 ring / 4 defect ring / 8	no link / 0	Beschreibt die Topologie des Sercos-Systems: <ul style="list-style-type: none"> <li>● 0 = Kein Sercos-Gerät verbunden.</li> <li>● 1 = Alle Sercos-Geräte sind mit Port 1 verbunden.</li> <li>● 2 = Alle Sercos-Geräte sind mit Port 2 verbunden.</li> <li>● 3 = Alle Sercos-Geräte sind mit Port 1 und Port 2 verbunden.</li> <li>● 4 = Die Verbindung von Port 1 zu Port 2 ist geschlossen (der Ring ist geschlossen).</li> <li>● 8 = Ungültige Ringtopologie, Topologiewechsel noch nicht abgeschlossen.</li> </ul>
ScannedDevices	R	AD	DINT	-	-	Anzahl der physisch gescannten Sercos-Geräte.
UsedDevices	R	AD	DINT	-	-	Anzahl der konfigurierten und physisch zugewiesenen Sercos-Geräte.
LastDeviceP1	R	AD	STRING	"	"	Name des letzten physischen Sercos-Geräts auf Port 1.
LastDeviceP2	R	AD	STRING	"	"	Name des letzten physischen Sercos-Geräts auf Port 2.

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
<b>Phasensteuerung</b>						
State	R	AD	DINT Enum	Phase 0 / 0 Phase 1 / 1 Phase 2 / 2 Phase 3 / 3 Phase 4 / 4 Firmware download / 5 Phase 6 / 6 Bus scan / 7 Reinit Sercos / 8 Init / 10 Error / 11 Continuous light / 12 Zero bit stream / 13	Phase 0 / 0	Zeigt den Sercos-Systemzustand an: <ul style="list-style-type: none"> <li>● 0 = Konfiguration und Start</li> <li>● 1 = Normaler Betrieb, Sercos Phase 1.</li> <li>● 2 = Normaler Betrieb, Sercos Phase 2.</li> <li>● 3 = Normaler Betrieb, Sercos Phase 3.</li> <li>● 4 = Normaler Betrieb, Sercos Phase 4.</li> <li>● 5 = Firmware wird heruntergeladen.</li> <li>● 6 = Reserviert.</li> <li>● 7 = Bus wird gescannt.</li> <li>● 8 = Sercos wird neu initialisiert.</li> <li>● 10 = Tritt nur kurzzeitig während des Systemstarts auf.</li> <li>● 11 = In der Betriebsphase wurde ein Sercos-Fehler erkannt.</li> <li>● 12 = Nicht zutreffend (für optische Verbindung).</li> <li>● 13 = Nicht zutreffend (für optische Verbindung).</li> </ul>
PhaseSet	R/W	EF	DINT	0 bis 4	4	Mit diesem Parameter können Sie die erforderliche Kommunikationsphase des Sercos-Bus voreinstellen: <sup>(1)</sup> <ul style="list-style-type: none"> <li>● 0 bis 3 = Die Tasks des Programms werden gleichzeitig mit dem Sercos-Start gestartet.</li> <li>● 4 = Der Start der Programmtasks wird bis zu folgendem Zeitpunkt verzögert: <ul style="list-style-type: none"> <li>○ Der Sercos-Start hat Phase 4 erreicht (<i>State</i> = 4).</li> <li>○ Der Sercos-Start wird wegen eines Startfehlers abgebrochen (<i>State</i> = 11).</li> </ul> </li> </ul>
—						
ObjectType	R	AD	STRING	SERC32	SERC32	Objekttyp.

Parameter	Zugriff	Param. Typ	Datentyp	Wert	Standardwert	Beschreibung
stLogicalAddress	R	AD	ST_LogicalAddress	-	-	Logische Adresse der Sercos-Parameter. LogicalAddress = STRUCT (udiType, udiInstance, udiParameterId)

(\*) Weitere Informationen zu den Parameter-Zugriffsrechten finden Sie unter Parametertypen (*siehe Seite 29*).

(1) Sobald der Parameter 'PhaseSet' den aktiven Austausch von Echtzeitdaten angibt, ist eine Neuinitialisierung der Achsen für die Positionssynchronisierung erforderlich.

## **HINWEIS**

### **POSITIONSVERLUST AUFGRUND DER SERCOS-BUSPHASENÄNDERUNG**

Stellen Sie die Neuinitialisierung oder das Homing des Bewegungssystems programmatisch sicher, wenn zum ersten Mal Sercos-Phase 4 erreicht wird.

**Die Nichtbeachtung dieser Anweisungen kann Sachschäden zur Folge haben.**

### Sercos-Betriebsphasen

In der folgenden Tabelle werden die Sercos-Betriebsphasen (Phase 0 bis 4) erläutert:

Betriebsphase	Beschreibung
Phase 0	Stellt sicher, dass die Anzahl der verbundenen Geräte konstant bleibt und welche Topologie verwendet wird. Dann wird mit der festgelegten Topologie ein Wechsel in Phase 1 ausgeführt.
Phase 1	Stellt sicher, ob die Slaves kontaktiert werden können. Dafür werden alle konfigurierten Slaves kurz adressiert. Wenn alle konfigurierten Bus-Slaves im Ring kontaktiert werden können, wird Kommunikationsphase 2 aktiv.
Phase 2	Der Master tauscht nacheinander mit den Slaves wichtige Kommunikationsparameter und Daten zu allgemeinen Geräteeigenschaften aus. Dadurch wird die Konfiguration des zyklischen Kanals definiert und festgelegt. Wenn alle Bus-Slaves konfiguriert sind, wird Kommunikationsphase 3 aktiv.
Phase 3	<b>Parameterphase:</b> Der Master tauscht Parameter mit den Slaves aus. Alle Slaves können gleichzeitig adressiert werden. Es sind noch keine Echtzeitdaten verfügbar.
Phase 4	<b>Betriebsphase:</b> Zyklischer Austausch von Echtzeitdaten (zyklische Kommunikation). Über den Servicekanal kann eine beliebige Anzahl an Parametern gelesen und geschrieben werden (nicht-zyklische Kommunikation).

## Sercos-Geräte

### Einführung

Der Modicon LMC078 Motion Controller unterstützt folgende Sercos-Geräte:

- Lexium LXM32S-Antriebe
- TM5NS31 Sercos-Schnittstellenmodule
- Sercos-Geräte von Drittanbietern

### Hinzufügen eines Sercos-Geräts

Es gibt zwei Möglichkeiten, Sercos-Geräte hinzuzufügen:

- Verwenden des Editors für die Geräteadressierung (*siehe Seite 216*)
- Verwenden der im Folgenden beschriebenen Methode.

Um ein Sercos-Gerät hinzuzufügen, wählen Sie das Element **Lexium 32 S** oder **TM5NS31-Schnittstelle** im Hardwarekatalog aus, ziehen Sie das Element in die **Gerätebaumstruktur** und legen Sie es auf dem **SERCOSIII-Knoten** ab.

Weitere Informationen zum Hinzufügen von Geräten in einem Projekt finden Sie unter:

- Verwenden der Methode Drag&Dop (*siehe SoMachine, Programmierhandbuch*) (Ziehen und Ablegen)
- Verwenden der Kontextmenüs oder Plus-Schaltflächen (*siehe SoMachine, Programmierhandbuch*)

### Hinzufügen eines Sercos-Geräts eines Drittanbieters

Sie können auch Sercos-Geräte von Drittanbietern hinzufügen (*siehe SoMachine, Programmierhandbuch*).

**HINWEIS:** Abhängig vom Drittanbieter müssen Sie den Parameter `Producer-CycleTime` des Sercos-Geräts gegebenenfalls auf einen anderen Wert als den Standardwert (1 ms) einstellen. Für weitere Informationen, siehe die spezifischen Angaben in der Gerätedokumentation des Drittanbieters.

## Editor für die Geräteadressierung

### Einführung

Der Editor für die **Geräteadressierung** unterstützt die folgenden Funktionen:

- Scannen der mit dem Sercos-Bus verbundenen Geräte
- Hinzufügen neuer Sercos-Geräte
- Definieren von Identifizierungsparametern der Sercos-Geräte

Um auf das Tool **Geräteadressierung** zuzugreifen, wählen Sie die Registerkarte **Tools-Baumstruktur** aus, und doppelklicken Sie auf **Geräteadressierung**:

**Geräteadressierung**

1 Neu LXM32S Hinzufügen Sercos-Scan starten

Geräte der Steuerungskonfiguration				Gescannte Geräte	
IEC-ID	Typ	SERCOS-Adresse	Betriebsmodus	2 gescannte, 2 zugewiesene Geräte	
DRV_Lexium32S	LXM32S	2	Optio...	1   LXM32S   2324035233	LEXIUM_SERVO   2
BC_TM5NS31	TM5NS31	3	Real	2   TM5NS31   B37C0170632	Modulares E/A-Gerät   3

**Farblgende**

- Login fehlerfrei möglich
- Login nicht ohne Fehler möglich
- Kein gescanntes Gerät zugeordnet
- Login fehlerfrei möglich; Unterschiede in irrelevanten Werten werden jedoch hervorgehoben
- Login fehlerfrei möglich; gescanntes Gerät wird jedoch nicht zugeordnet

Werte aller zugeordneten Geräte übernehmen

Bezeichnung	Beschreibung
<b>Geräte der SPS-Konfiguration</b>	<p>Im linken Teil des Editorfensters werden die LXM32S-Antriebe und die TM5NS31-Schnittstellenmodule der Sercos-Konfiguration im Steuerungsprojekt angezeigt.</p> <p><b>HINWEIS:</b> Wenn die Sercos-Objekte automatisch zugewiesen werden, werden sie in der Reihenfolge ihrer topologischen Adressen aufgeführt, beginnend mit dem niedrigsten Wert. Wenn bereits Sercos-Objekte von einem vorigen Scan aufgeführt werden, werden alle weiteren Sercos-Objekte am Ende der Liste hinzugefügt, in der Reihenfolge ihrer topologischen Adressen und beginnend mit dem niedrigsten Wert.</p>
<b>Betriebsmodus</b> <i>(siehe Seite 219)</i>	<p>Der Betriebsmodus wird verwendet, um den Betrieb des Geräts festzulegen.</p> <p><b>HINWEIS:</b> Wenn Sie einen Wert in dieser Spalte ändern, während Sie die Umschalttaste gedrückt halten, werden alle Werte in dieser Spalte auf diesen Wert gesetzt.</p>

Bezeichnung	Beschreibung
<b>Sercos-Scan starten</b>	Klicken Sie auf diese Schaltfläche, um die Suche nach realen LXM32S-Antrieben und den TM5NS31-Schnittstellenmodulen zu starten, die mit dem Sercos-Bus verbunden sind. So führen Sie die Suche durch: <ul style="list-style-type: none"> <li>● Der Sercos-Bus wechselt in Phase 0 (<i>siehe Seite 214</i>).</li> <li>● Alle Anwendungen werden gestoppt.</li> <li>● Alle Diagnosemeldungen müssen bestätigt werden.</li> </ul>
<b>Gescannte Geräte</b>	Nach Durchführen eines Scans werden im rechten Teil des Editorfensters die LXM32S-Antriebe und die TM5NS31-Schnittstellenmodule angezeigt, die mit dem Sercos-Bus verbunden sind. In der Kopfzeile der Spalte wird die Anzahl der gescannten Geräte und die Anzahl der automatisch zugewiesenen Geräte angezeigt. Alle Geräte, die automatisch zugewiesen wurden, werden in einer anderen Farbe als Weiß angezeigt. Sie können die Zuweisung später manuell über eine Dropdownliste in der Spalte ganz rechts ändern.
<b>&lt;--</b>	Klicken Sie auf diese Schaltfläche, um die in dieser Zeile zugewiesenen Werte des Sercos-Geräts hinzuzufügen.
<b>Werte aller zugeordneten Geräte übernehmen</b>	Wenn Sie alle LXM32S-Antriebe und TM5NS31-Schnittstellenmodule hinzugefügt haben, klicken Sie auf diese Schaltfläche, um die Gerätedaten der gescannten Sercos-Geräte in den zugewiesenen Objekten im Sercos-Knoten der Steuerungskonfiguration anzuwenden.
<b>Hinzufügen</b>	Klicken Sie auf diese Schaltfläche, um dem Sercos-Knoten der Steuerungskonfiguration neue Geräte hinzuzufügen.

**HINWEIS:** Die Funktionen des Editors für die **Geräteadressierung** sind nur im Offlinemodus verfügbar.

Die Farbcodierung einer Zeile weist darauf hin, ob eine Anmeldung für das jeweilige Sercos-Objekt in der Steuerungskonfiguration durchgeführt werden kann und möglich ist.

Die Farben bedeuten Folgendes:

Farbe	Beschreibung
Grün	Mit diesem Objekt kann eine Anmeldung durchgeführt werden.
Rot	Mit diesem Objekt kann keine Anmeldung durchgeführt werden.
Weiß	Diesem Objekt ist kein Sercos-Gerät zugewiesen.
Gelb	Die Anmeldung für dieses Objekt ist unsicher. Es gibt Abweichungen bei den Werten des zugewiesenen Sercos-Geräts. Die nicht übereinstimmenden Werte sind <b>fett</b> markiert.
Rosa	Die Anmeldung für dieses Objekt ist unsicher. Es wurde jedoch kein Sercos-Gerät zugewiesen, nachdem der Sercos-Scan ("Sercos-Scan starten") durchgeführt wurde, obwohl der <b>Betriebsmodus</b> auf <b>Real</b> gesetzt ist.


## Sercos-Scan

Nach dem Scan (Schaltfläche **SERCOS-Scan starten**) versucht das Programm, Sercos-Objekte aus der Steuerungskonfiguration den Geräten zuzuweisen, die über die topologische Adresse mit dem Sercos-Bus verbunden sind. Alle Geräte, die automatisch zugewiesen wurden, werden in einer anderen Farbe als Weiß angezeigt.

In der Kopfzeile der Spalte wird die Anzahl der gescannten Geräte und die Anzahl der automatisch zugewiesenen Geräte angezeigt:

Gescannte Geräte	
	2 gescannte, 2 zugewiesene Geräte
<--	1   LXM32S   2324035233     LEXIUM_SERVO   2
<--	2   TM5NS31   B37C0170632     Modulares E/A-Gerät   3

Sie können die automatische Zuweisung später manuell über eine Dropdownliste in der Spalte ganz rechts ändern.

- Klicken Sie auf die Schaltfläche  in der Zeile, die Sie ändern möchten. In der Dropdownliste werden alle Sercos-Geräte dieses Typs angezeigt, die noch nicht zugewiesen sind.
- Wählen Sie das gewünschte Gerät aus der Liste aus.

**HINWEIS:** Sie können die leere Zeile unten in der Liste verwenden, um eine Zuweisung zurückzusetzen.

Jede Zeile in der Auswahlliste enthält eine kurze Beschreibung des Sercos-Geräts.

Die Werte sind durch einen vertikalen Strich (|") getrennt und entsprechen den folgenden Parametern:

- TopologyAddress
- ObjectType
- SerialNumber
- ConfiguredApplicationType
- SercosAddress

Sie haben zwei Möglichkeiten, die Parameterwerte zu übernehmen:

- Sie können die Werte eines zugewiesenen Geräts anwenden, indem Sie auf die Schaltfläche <-- klicken.
- Sie können die Werte aller zugewiesenen Geräte anwenden, indem Sie auf **Werte aller zugeordneten Geräte übernehmen** klicken.

Wenn Sie diese Werte annehmen, schreibt das Programm die Werte der zugewiesenen und gescannten Sercos-Geräte, die für die Bereitstellung erforderlich sind, in die verbundenen Sercos-Objekte in der Steuerungskonfiguration.

**HINWEIS:** Wenn die Werte angewendet wurden, wird die entsprechende Zeile in Grün angezeigt.

## Betriebsmodus

Der **Betriebsmodus** wird verwendet, um den Betrieb eines Sercos-Geräts festzulegen.

Wählen Sie den gewünschten Betriebsmodus in der Spalte **Betriebsmodus** aus der Dropdownliste aus:

Betriebsmodus	Beschreibung
Virtual	Das Sercos-Gerät ist physisch nicht vorhanden.
Real	Das Sercos-Gerät muss physisch vorhanden sein.
Deactivated	Das Sercos-Gerät wird nicht verwendet. Es ist evtl. physisch vorhanden.
Optional	Das Sercos-Gerät ist evtl. physisch vorhanden, doch dies ist keine Voraussetzung.

**HINWEIS:** Wenn Sie einen Wert in dieser Spalte ändern, während Sie die Umschalttaste gedrückt halten, werden alle Werte in dieser Spalte auf diesen Wert gesetzt.

## Manuelles Hinzufügen von Geräten

Im Editor für die **Geräteadressierung** können Sie der Steuerungskonfiguration manuell neue Geräte hinzufügen.

Gehen Sie wie folgt vor, um ein Gerät manuell hinzuzufügen:

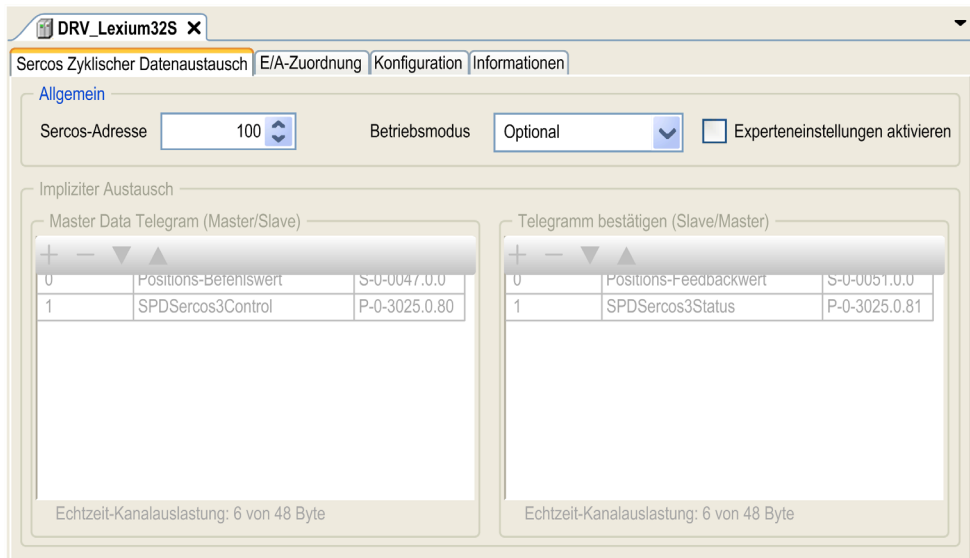
Schritt	Aktion
1	Geben Sie die gewünschte Anzahl der neuen Geräte ein.
2	Wählen Sie das gewünschte Gerät aus der Liste aus: <ul style="list-style-type: none"> <li>● <b>LXM32S:</b> Lexium 32S-Antrieb</li> <li>● <b>TM5NS31:</b> TM5 Sercos-Schnittstellenmodul</li> </ul>
3	Klicken Sie auf <b>Hinzufügen</b> . <b>Ergebnis:</b> Die Geräte werden der Steuerungskonfiguration hinzugefügt.

**HINWEIS:** Wenn beim Hinzufügen eines Geräts ein Fehler erkannt wird, werden die Geräte, die nicht hinzugefügt werden könnten, im Meldungsfenster mit einer Erklärung angezeigt.

## Lexium LXM32S-Antriebskonfiguration

### Beschreibung

Um auf das Fenster des Geräteeditors zuzugreifen, doppelklicken Sie in der **Gerätebaumstruktur** auf den Antriebsknoten.



Das Fenster des Geräteeditors des LXM32S-Antriebs enthält die folgenden Registerkarten:

- **Sercos Zyklischer Datenaustausch:**
  - Konfiguration der Sercos-Adresse des Antriebs.
  - Konfiguration des Betriebsmodus (*siehe Seite 219*) des Antriebs.
  - Konfiguration der Sercos-impliziten Austauschvorgänge (IDN-Konfiguration von MDT- und AT-Telegrammen).
- **E/A-Abbild:** Auf dieser Registerkarte können Sie IEC-Variablen erstellen und der für zyklische Austauschvorgänge ausgewählten IDN zuweisen.
- **Konfiguration:** Die Konfigurationsparameter des Antriebs (Konfigurieren Sie die Antriebsparameter über die Registerkarte **Sercos Zyklischer Datenaustausch**).
- **Informationen:** Diese Registerkarte zeigt allgemeine Informationen zum Gerät an (Name, Beschreibung, Hersteller, Version, Abbild).

**HINWEIS:**

Die Standardkonfiguration des zyklischen Austauschs umfasst vier IDN, die nicht bearbeitbar sind:

- MDT-Telegramm (Steuerung zu Antrieb):
  - Position command value: S-0-0047.0.0
  - SPDSercos3Control: P-0-3025.0.80
- AT-Telegramm (Antrieb zu Steuerung):
  - Position feedback value: S-0-0051.0.0
  - SPDSercos3Status: P-0-3025.0.81

Weitere Informationen zu den in den LXM32S-Antrieben implementierten IDNs finden Sie im Produkthandbuch zu Lexium LXM32S.

**Konfiguration der Experteneinstellungen**

Mit der Option für Experteneinstellungen können Sie die Liste der Sercos IDN bearbeiten, die zyklisch zwischen Steuerung und Antrieb ausgetauscht werden.

**HINWEIS:** Die IDN-Gesamtlänge der MDT- und AT-Telegramme ist auf 48 Byte beschränkt.

Um die Experteneinstellung zu aktivieren, aktivieren Sie das Kontrollkästchen **Experten-Einstellungen aktivieren**:

DRV\_Lexium32S X

Sercos Zyklischer Datenaustausch | E/A-Zuordnung | Konfiguration | Informationen

Allgemein

Sercos-Adresse: 100      Betriebsmodus: Optional       Experteneinstellungen aktivieren

Impliziter Austausch

Master Data Telegram (Master/Slave)

0	Positions-Befehlswert	S-0-0047.0.0
1	SPDSercos3Control	P-0-3025.0.80
2	Position am Referenzpunkt	P-0-3040.0.11
3	Überwachung des Geschwindigkeitsschwellenwerts	P-0-3006.0.27

Echtzeit-Kanalauslastung: 14 von 48 Byte

Telegramm bestätigen (Slave/Master)

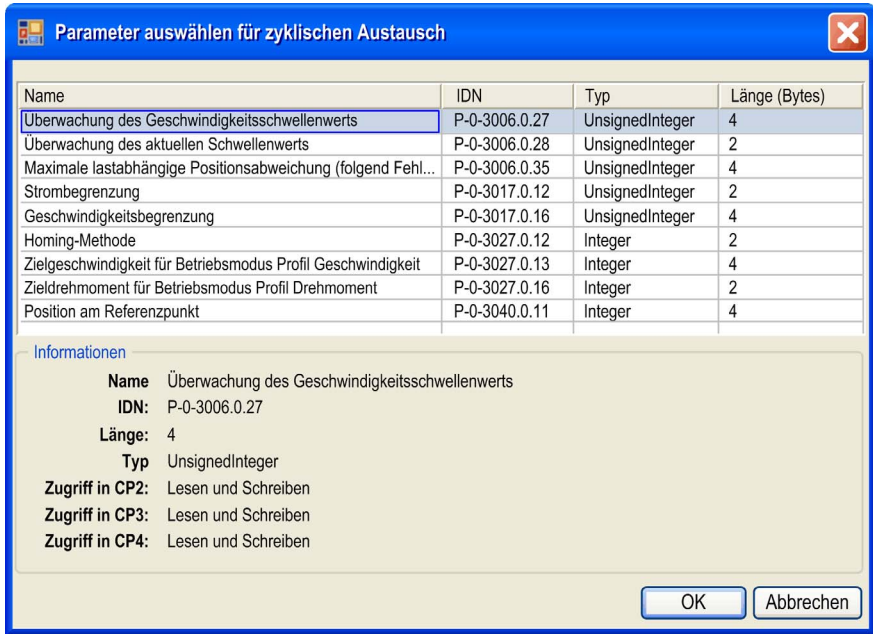
0	Positions-Feedbackwert	S-0-0051.0.0
1	SPDSercos3Status	P-0-3025.0.81
2	Physischer Status der Digitaleingänge und -ausgänge	P-0-3008.0.1
3	Aktiver Betriebsmodus	P-0-3027.0.4
4	Aktuelle Geschwindigkeit	P-0-3030.0.32

Echtzeit-Kanalauslastung: 14 von 48 Byte

Nun sind die folgenden Schaltflächen verfügbar:

Schaltfläche	Beschreibung
+	Klicken Sie auf diese Schaltfläche, um der Liste eine IDN hinzuzufügen (Beschreibung im Folgenden).
-	Wählen Sie aus der Liste eine IDN aus, und klicken Sie auf diese Schaltfläche, um eine IDN aus der Liste zu entfernen.
NACH-OBEN	Wählen Sie aus der Liste eine IDN aus, und klicken Sie auf diese Schaltfläche, um eine IDN in der Liste nach oben zu verschieben.
Nach-unten-Pfeil	Wählen Sie aus der Liste eine IDN aus, und klicken Sie auf diese Schaltfläche, um eine IDN in der Liste nach unten zu verschieben.

Gehen Sie wie folgt vor, um eine IDN dem MDT- oder AT-Telegramm hinzuzufügen:

Schritt	Aktion
1	<p>Klicken Sie auf +.</p> <p><b>Ergebnis:</b> Das folgende Dialogfeld wird angezeigt:</p> 
2	Wählen Sie die IDN aus, die Sie der Liste hinzufügen möchten.
3	<p>Klicken Sie auf <b>OK</b>.</p> <p><b>Ergebnis:</b> Die IDN, die dem MDT- oder AT-Telegramm hinzugefügt wird, und die Anzahl der verwendeten Bytes wird aktualisiert (<b>Echtzeit-Kanalauslastung</b>).</p>

## TM5NS31 Sercos Interface Module

### Stromaufnahme

Gehen Sie wie folgt vor, um den geschätzten Stromverbrauch der Erweiterungmodule anzuzeigen:

Schritt	Aktion
1	Klicken Sie mit der rechten Maustaste auf den Knoten <b>TM5NS31 Interface</b> in der <b>Gerätebaumstruktur</b> .
2	Wählen Sie <b>Stromverbrauch</b> aus.

Segment	Erstes Modul	Letztes Modul	Stromverbrauch	Verbleibender Strom (mA)
TM5-Bussegment 1			0 %	400
24V E/A-Segment 1	Module_9	Module_11	21 %	7833

**HINWEIS:** Der von der Funktion **Stromverbrauch** angegebene aktuelle Stromverbrauch basiert auf geschätzten und nicht auf tatsächlich gemessenen Werten. Die geschätzten Werte für die Ausgänge basieren auf typischen Lasten, können jedoch mithilfe der Einstellung "24 V E/A-Segment externer Strom" auf der Registerkarte E/A-Konfiguration eines jeden Moduls angepasst werden. Die Schätzungen für die Eingangssignale basieren auf bekannten internen Belastungen und können daher nicht verändert werden. Die Verwendung der Funktion **Stromverbrauch** ist zwar zum Prüfen der Strombilanz erforderlich, ersetzt jedoch keinesfalls die eigentliche und vollständige Prüfung und Inbetriebnahme des Systems. Die entsprechenden Informationen entnehmen Sie dem TM5 / TM7 System Planning and Installation Guide.

## Sercos Fehlercodes

### Fehlermeldungen von Sercos-Slaves

4-stellige Codes identifizieren von Sercos-Slaves an den Master berichtete Fehlermeldungen. Die folgenden sind standardmäßige Sercos-Fehlercodes:

Fehlercode	Beschreibung	Kommentar
0x0nnn	Allgemeiner Fehler	–
0x0000	Kein Fehler im Service-Kanal	–
0x0001	Service-Kanal nicht offen	–
0x0009	Ungültiger Zugriff zum Schließen des Service-Kanals	–
0x1nnn	Element 1	Identifikationsnummer
0x1001	IDN nicht unterstützt	–
0x1009	Ungültiger Zugriff auf Element 1	–
0x2nnn	Element 2	Name
0x2001	Name nicht unterstützt	–
0x2002	Übertragung des Namens zu kurz	Zu frühe Einstellung der „letzten Übertragung“ durch Master
0x2003	Übertragung des Namens zu lang	Keine Einstellung der „letzten Übertragung“ durch Master
0x2004	Name kann nicht geändert werden	Name ist schreibgeschützt
0x2005	Name ist schreibgeschützt	–
0x3nnn	Element 3	Attribut
0x3002	Übertragung des Attributs zu kurz	Zu frühe Einstellung der „letzten Übertragung“ durch Master
0x3003	Übertragung des Attributs zu lang	Keine Einstellung der „letzten Übertragung“ durch Master
0x3004	Attribut kann nicht geändert werden	Attribut ist schreibgeschützt
0x3005	Attribut ist schreibgeschützt	–
0x4nnn	Element 4	Einheit
0x4001	Einheit nicht unterstützt	–
0x4002	Übertragung der Einheit zu kurz	Zu frühe Einstellung der „letzten Übertragung“ durch Master
0x4003	Übertragung der Einheit zu lang	Keine Einstellung der „letzten Übertragung“ durch Master
0x4004	Einheit kann nicht geändert werden	Einheit ist schreibgeschützt
0x4005	Einheit ist schreibgeschützt	–

Fehlercode	Beschreibung	Kommentar
0x5nnn	Element 5	Minimaler Eingangswert
0x5001	Minimaler Eingangswert nicht unterstützt	–
0x5002	Übertragung des minimalen Eingangswerts zu kurz	Zu frühe Einstellung der „letzten Übertragung“ durch Master
0x5003	Übertragung des minimalen Eingangswerts zu lang	Keine Einstellung der „letzten Übertragung“ durch Master
0x5004	Minimaler Eingangswert kann nicht geändert werden	Minimaler Eingangswert ist schreibgeschützt
0x5005	Minimaler Eingangswert ist schreibgeschützt	–
0x6nnn	Element 6	Maximaler Eingangswert
0x6001	Maximaler Eingangswert nicht unterstützt	–
0x6002	Übertragung des maximalen Eingangswerts zu kurz	Zu frühe Einstellung der „letzten Übertragung“ durch Master
0x6003	Übertragung des maximalen Eingangswerts zu lang	Keine Einstellung der „letzten Übertragung“ durch Master
0x6004	Maximaler Eingangswert kann nicht geändert werden	Maximaler Eingangswert ist schreibgeschützt
0x6005	Maximaler Eingangswert ist schreibgeschützt	–
0x7nnn	Element 7	Betriebsdaten
0x7002	Übertragung der Betriebsdaten zu kurz	Zu frühe Einstellung der „letzten Übertragung“ durch Master
0x7003	Übertragung der Betriebsdaten zu lang	Keine Einstellung der „letzten Übertragung“ durch Master
0x7004	Betriebsdaten können nicht geändert werden	Betriebsdaten sind schreibgeschützt
0x7005	Betriebsdaten sind in dieser Kommunikationsphase schreibgeschützt	–
0x7006	Betriebsdaten sind kleiner als der minimale Eingangswert	–
0x7007	Betriebsdaten sind größer als der maximale Eingangswert	–
0x7008	Ungültige Betriebsdaten	Die ungültigen Betriebsdaten kann einer der folgenden nicht unterstützen Werte sein: <ul style="list-style-type: none"> <li>● Bitnummer oder Bitkombination,</li> <li>● ein Wert, Code oder</li> <li>● konfigurierte IDN sein</li> </ul>
0x7009	Betriebsdaten ist durch ein Passwort schreibgeschützt	–

Fehlercode	Beschreibung	Kommentar
0x700A	Betriebsdaten sind schreibgeschützt, da sie zyklisch konfiguriert sind	IDN wird in MDT oder AT konfiguriert. Aus diesem Grund ist das Schreiben über den Service-Kanal nicht erlaubt
0x700B	Ungültige indirekte Adressierung	z.B. Datencontainer, Listenhandhabung usw.
0x700C	Betriebsdaten sind wegen anderen Einstellungen schreibgeschützt	z.B. Betriebsmodus, Untergerät ist aktiviert, Einstellung der Kommunikationsversion usw.
0x700D	Ungültige Gleitkommanummer	–
0x700E	Betriebsdaten sind auf der Parametrierungsebene schreibgeschützt	–
0x700F	Betriebsdaten sind auf der Betriebsebene schreibgeschützt	–
0x7010	Vorgangsbefehl bereits aktiv	–
0x7011	Vorgangsbefehl kann nicht unterbrochen werden	–
0x7012	Vorgangsbefehl kann nicht ausgeführt werden	z.B. Der Vorgangsbefehl kann in dieser Phase nicht aktiviert werden
0x7013	Vorgangsbefehl kann nicht ausgeführt werden	Die entsprechenden Parameter sind ungültig
0x7014	Die empfangene aktuelle Länge der Listenparameter entspricht nicht den Erwartungen	–
0x7015	Betriebsdaten sind noch nicht vollständig erstellt	Versuchen Sie es später erneut, wenn mehr Zeit zur Erstellung der Betriebsdaten benötigt wird
0x71nn	Übertragung der Listenparameter über SVC per Segment	–
0x7101	IDN in S-0-0394 ungültig	–
0x7102	Leere Liste in S-0-0397 für Schreibzugriff nicht erlaubt	–
0x7103	Maximallänge der Liste in S-0-0394 wird durch die Übernahme des Listensegments überschritten	–
0x7104	Nur Lesezugriff (schreibgeschützt): die Lage des Listensegments vom Listenindex überschreitet die aktuelle Länge der Liste in S-0-0394	–
0x7105	IDN in S-0-0394 ist schreibgeschützt	–
0x7106	Betriebsdaten im Listensegment sind kleiner als der minimale Eingangswert	–
0x7107	Betriebsdaten im Listensegment sind größer als der maximale Eingangswert	–
0x7108	Ungültiger Listenindex in IDN S-0-0395	–

Fehlercode	Beschreibung	Kommentar
0x7109	Parameter in IDN S-0-0394 hat keine variable Länge	–
0x710A	IDN S-0-0397 nicht als Betriebsdaten in S-0-0394 erlaubt	–
0x8nnn	Reserviert für interne Master-Fehlercodes	Fehlercodes können durch den Hersteller der Steuerungseinheiten (z.B. NC, SPS) definiert sein
0xAAnnn	Reserviert	–
0xBnnn	Reserviert	–
0xCnnn	Reserviert für spezifische Slave-Fehlercodes	Verwendet für Fehleranalyse und Traceaufzeichnung (Problembhebung)
0xDnnn	Fehlercodes werden nicht über SVC erstellt und übermittelt	Fehlercodes werden durch ein TWG an Sercos definiert
0xD000	Kein Fehler	–
0xD001	Service-Kanal (vorübergehend) nicht verfügbar	–
0xD002	Service-Kanal durch Applikation beschäftigt	–
0xD003	Service-Kanal beschäftigt, Slave verarbeitet vorherigen Request	–
0xD004	Sercos-Slave nicht erreichbar	–
0xD005	Service-Kanal-Transaktion abgebrochen	–
0xD006	Schreiben dieses Elements wird durch den Service-Kanal nicht unterstützt	–
0xEAnnn	Reserviert für interne Master-Fehlercodes	–
0xFnnn	Reserviert für interne Master-Fehlercodes	–

**HINWEIS:** Alle anderen Fehlercodes sind reserviert.

### Fehlermeldungen vom Sercos-Master

Wenn der Sercos-Master Fehler erkennt, werden diese entweder als 5-stelliger Hexadezimalwert oder als negativer Dezimalwert spezifiziert.

Fehlercode	Beschreibung
20001	SVC: Neuer Request mit höherer Priorität während eines aktiven internen Requests
20002	SVC: Neuer interner Request während eines aktiven internen Requests
20003	SVC: Übertragung durch anderen Funktionsaufruf mit höherer Priorität abgebrochen
20004	SVC: Neuer Übertragsungsrequest, MBusy ist jedoch nicht eingestellt
20005	SVC: Ungültiger Status: AHS != MHS während der Einstellung von BusyAT
20006	SVC: Zeitüberschreitung, da Slave keine BusyAT-Flag eingestellt hat
20007	SVC: Zeitüberschreitung, da Slave die BusyAT-Flag zu lang eingestellt hat

<b>Fehlercode</b>	<b>Beschreibung</b>
20008	SVC: Schreiben mit nicht unterstütztem Element (erlaubt: 1 oder 7)
20009	SVC: Schreiben mit Datenlänge = Null
-1	Anderer Fehler
-431	Service-Request-Fehler (z.B. Zeitüberschreitung/Timeout)
-445	Dienst-Timeout
-467	Interner Statusfehler der Maschine

---

# Kapitel 14

## Konfiguration der seriellen Leitung

---

### Einführung

In diesem Kapitel wird die Konfiguration der SL-Kommunikation mit dem Modicon LMC078 Motion Controller beschrieben.

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Konfiguration der seriellen Leitung	230
ASCII-Manager	232
SoMachine-Netzwerkmanager	234
Modbus Serial-E/A-Scanner	235
Hinzufügen eines Geräts auf dem Modbus Serial-E/A-Scanner	237
Modbus-Manager	244
Hinzufügen eines Modems zu einem Manager	249

## Konfiguration der seriellen Leitung

### Einführung

Das Fenster zur Konfiguration der seriellen Leitung ermöglicht das Konfigurieren der physischen Parameter der seriellen Leitung (Baudrate, Parität usw).

### Konfiguration der seriellen Leitung

Um eine serielle Leitung zu konfigurieren, doppelklicken Sie auf **Serielle Leitung** in der **Gerätebaumstruktur**.

Das unten dargestellte Fenster **Konfiguration** wird angezeigt:

The screenshot shows a configuration window titled 'Konfiguration'. It is divided into two sections: 'Serielle Leitung' and 'Physikalisches Medium'.  
 Under 'Serielle Leitung', there are four dropdown menus:  
 - Baudrate: 19200  
 - Parität: Gerade  
 - Datenbits: 8  
 - Stoppbits: 1  
 Under 'Physikalisches Medium', there are two radio buttons for 'RS-485' (selected) and 'RS232'. To the right, there is a dropdown menu for 'Polarisationwiderstand' set to 'Nein'.

Die folgenden Parameter müssen für alle an den Port angeschlossenen seriellen Geräte identisch sein.

Element	Beschreibung
<b>Baudrate</b>	Übertragungsgeschwindigkeit in Bit/s
<b>Parität</b>	Dient zur Fehlererkennung
<b>Datenbits</b>	Anzahl der Bits zur Übertragung von Daten
<b>Stoppbits</b>	Anzahl der Stoppbits
<b>Physikalisches Medium</b>	Geben Sie das zu verwendende Medium an: <ul style="list-style-type: none"> <li>● RS485 (mit oder ohne Verwendung des Polarisierungswiderstands)</li> <li>● RS232</li> </ul>
<b>Polarisationswiderstand</b>	In die Steuerung sind Polarisationswiderstände integriert. Dieser Parameter schaltet sie ein bzw. aus.

Die SL-Ports (serielle Leitung) der Steuerung sind werkseitig oder bei Aktualisierung der Firmware der Steuerung standardmäßig für das SoMachine-Protokoll konfiguriert. Das SoMachine-Protokoll ist mit anderen Protokollen, wie z. B. Modbus Serial Line, nicht kompatibel. Wenn an eine aktive, für Modbus konfigurierte serielle Leitung eine neue Steuerung angeschlossen oder die Firmware einer daran angeschlossenen Steuerung aktualisiert wird, kann dies dazu führen, dass die anderen Geräte auf der Leitung die Kommunikation einstellen. Vergewissern Sie sich vor dem Herunterladen einer gültigen Anwendung, bei der die entsprechenden Ports für das jeweilige Protokoll konfiguriert sind, dass die Steuerung nicht mit einem aktiven Modbus SL-Netzwerk verbunden ist.

## ***HINWEIS***

### **UNTERBRECHUNG DER KOMMUNIKATION ÜBER DIE SERIELLE LEITUNG**

Vergewissern Sie sich, dass die SL-Ports in Ihrer Anwendung vorschriftsmäßig für Modbus konfiguriert wurden, bevor Sie die Steuerung physisch an ein aktives Modbus Serial Line-Netzwerk anschließen.

**Die Nichtbeachtung dieser Anweisungen kann Sachschäden zur Folge haben.**

Diese Tabelle gibt die maximale Baudrate der Manager an:

<b>Manager</b>	<b>Maximale Baudrate (Bit/s)</b>
SoMachine-Netzwerkmanager	115200
Modbus-Manager	
ASCII-Manager	
Modbus-IOScanner	

## ASCII-Manager

### Einführung

Der ASCII-Manager dient zum Senden und/oder Empfangen von Daten mittels eines einfachen Geräts.

### Hinzufügen des Managers

Wenn Sie Ihrer Steuerung einen ASCII-Manager hinzufügen möchten, wählen Sie den **ASCII-Manager** im **Hardware-Katalog** aus, ziehen Sie ihn in die **Gerätebaumstruktur** und legen Sie ihn dann auf einem der optisch hervorgehobenen Knoten ab.

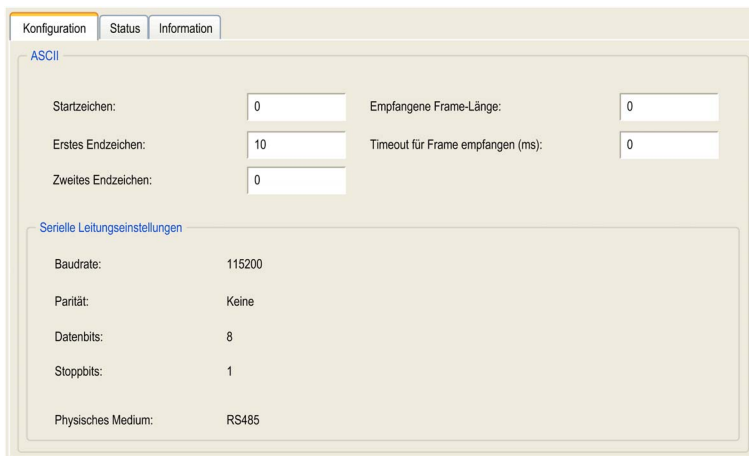
Weitere Informationen zum Hinzufügen von Geräten in einem Projekt finden Sie unter:

- Verwenden der Methode Drag&Dop (*siehe SoMachine, Programmierhandbuch*) (Ziehen und Ablegen)
- Verwenden der Kontextmenüs oder Plus-Schaltflächen (*siehe SoMachine, Programmierhandbuch*)

### ASCII-Manager-Konfiguration

Um den ASCII-Manager Ihrer Steuerung zu konfigurieren, doppelklicken Sie auf **ASCII-Manager** in der **Gerätebaumstruktur**.

Das Konfigurationsfenster für den ASCII-Manager wird angezeigt:



Konfiguration		Status	Information
<b>ASCII</b>			
Startzeichen:	<input type="text" value="0"/>	Empfangene Frame-Länge:	<input type="text" value="0"/>
Erstes Endzeichen:	<input type="text" value="10"/>	Timeout für Frame empfangen (ms):	<input type="text" value="0"/>
Zweites Endzeichen:	<input type="text" value="0"/>		
<b>Serielle Leitungseinstellungen</b>			
Baudrate:	115200		
Parität:	Keine		
Datenbits:	8		
Stoppbits:	1		
Physisches Medium:	RS485		

Legen Sie die Parameter wie in dieser Tabelle beschrieben fest:

Parameter	Beschreibung
<b>Startzeichen</b>	Bei 0 wird kein Startzeichen im Frame verwendet. Andernfalls wird im <b>Empfangsmodus</b> das entsprechende Zeichen im ASCII-Format verwendet, um den Beginn eines Frames zu erkennen. Im <b>Sendemodus</b> wird dieses Zeichen zu Beginn eines Frames eingefügt.
<b>Erstes Endzeichen</b>	Bei 0 wird kein Endzeichen im Frame verwendet. Andernfalls wird im <b>Empfangsmodus</b> das entsprechende Zeichen im ASCII-Format verwendet, um das Ende eines Frames zu erkennen. Im <b>Sendemodus</b> wird dieses Zeichen am Ende eines Frames eingefügt.
<b>Zweites Endzeichen</b>	Bei 0 wird kein zweites Endzeichen im Frame verwendet. Andernfalls wird im <b>Empfangsmodus</b> das entsprechende Zeichen im ASCII-Format verwendet, um das Ende eines Frames zu erkennen. Im <b>Sendemodus</b> wird dieses Zeichen am Ende eines Frames eingefügt.
<b>Empfangene Frame-Länge</b>	Bei 0 wird dieser Parameter nicht verwendet. Aus diesem Parameter kann das System beim Empfang folgern, dass das Ende eines Frames erreicht wurde, wenn die Steuerung eine bestimmte Anzahl von Zeichen empfangen hat. <b>Hinweis:</b> Dieser Parameter kann nicht gleichzeitig mit <b>Timeout für Frame empfangen (ms)</b> verwendet werden.
<b>Timeout für Frame empfangen (ms)</b>	Bei 0 wird dieser Parameter nicht verwendet. Aus diesem Parameter kann das System beim Empfang folgern, dass das Ende eines Frames erreicht wurde, nachdem eine Stille von einer bestimmten Anzahl von ms eingehalten wurde.
<b>Serielle Leitungseinstellungen</b>	Die im Konfigurationsfenster der seriellen Leitung ( <i>siehe Seite 230</i> ) eingestellten Parameter.

**HINWEIS:** Falls mehrere Bedingungen zur Beendigung eines Frames verwendet werden, wird der Austausch durch die erste Bedingung, die TRUE ist, beendet.

### Hinzufügen eines Modems

Weitere Informationen zum Hinzufügen eines Modems zum ASCII-Manager finden Sie unter Hinzufügen eines Modems zu einem Manager (*siehe Seite 249*).

## SoMachine-Netzwerkmanager

### Einführung

Verwenden Sie den SoMachine-Netzwerkmanager zum Austauschen von Variablen mit einer erweiterten HMI (XBTGT/XBTGK) über das SoMachine-Softwareprotokoll oder wenn die serielle Leitung für die SoMachine-Programmierung verwendet wird.

### Hinzufügen des Managers

Wenn Sie Ihrer Steuerung einen SoMachineNetzwerkmanager hinzufügen möchten, wählen Sie den **SoMachine-Netzwerkmanager** im **Hardware-Katalog** aus, ziehen Sie ihn in die **Gerätebaumstruktur** und legen Sie ihn dann auf einem der optisch hervorgehobenen Knoten ab.

Weitere Informationen zum Hinzufügen von Geräten in einem Projekt finden Sie unter:

- Verwenden der Methode Drag&Dop (*siehe SoMachine, Programmierhandbuch*) (Ziehen und Ablegen)
- Verwenden der Kontextmenüs oder Plus-Schaltflächen (*siehe SoMachine, Programmierhandbuch*)

### Konfigurieren des Managers

Der SoMachine-Netzwerkmanager erfordert keine Konfiguration.

### Hinzufügen eines Modems

Informationen zum Hinzufügen eines Modems zum SoMachine-Netzwerkmanager finden Sie unter Hinzufügen eines Modems zu einem Manager (*siehe Seite 249*).

## Modbus Serial-E/A-Scanner

### Einführung

Der Modbus-E/A-Scanner erleichtert den Datenaustausch mit Modbus-Slave-Geräten.

### Hinzufügen eines Modbus-E/A-Scanners

Um einen Modbus-E/A-Scanner in Ihrem Projekt hinzuzufügen, wählen Sie den **Modbus\_IOScanner** im **Hardwarekatalog** aus, ziehen Sie ihn auf die **Gerätebaumstruktur**, und legen Sie ihn auf einem der optisch hervorgehobenen Knoten ab.

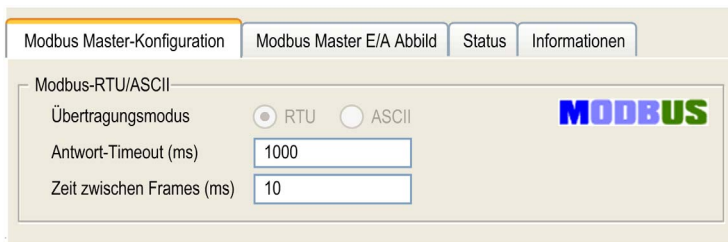
Weitere Informationen zum Hinzufügen von Geräten in einem Projekt finden Sie unter:

- Verwenden der Methode Drag&Dop (*siehe SoMachine, Programmierhandbuch*) (Ziehen und Ablegen)
- Verwenden der Kontextmenüs oder Plus-Schaltflächen (*siehe SoMachine, Programmierhandbuch*)

### Konfiguration des Modbus-E/A-Scanners

Um einen Modbus-E/A-Scanner für eine serielle Leitung zu konfigurieren, doppelklicken Sie auf **Modbus-E/A-Scanner** in der **Gerätebaumstruktur**.

Das Konfigurationsfenster wird angezeigt:



Legen Sie die Parameter wie in der folgenden Tabelle beschrieben fest:

Element	Beschreibung
<b>Übertragungsmodus</b>	Legen Sie den zu verwendenden Übertragungsmodus fest: <ul style="list-style-type: none"> <li>• RTU: Verwendet Binärcodierung und CRC-Fehlerprüfung (8 Datenbits).</li> <li>• ASCII: Meldungen sind im ASCII-Format, LRC-Fehlerprüfung (7 Datenbits)</li> </ul> Dieser Parameter muss für jedes Modbus-Gerät im Netzwerk identisch sein.
<b>Antwort-Timeout (ms)</b>	Bei Austauschvorgängen verwendetes Timeout.
<b>Zeit zwischen Frames (ms)</b>	Verzögerung, um Datenkollisionen auf dem Bus zu reduzieren. Dieser Parameter muss für jedes Modbus-Gerät im Netzwerk identisch sein.

**HINWEIS:** Verwenden Sie keine Funktionsbausteine aus der PLCCommunication-Bibliothek auf einer seriellen Leitung, für die einModbus-Modbus-E/A-Scanner konfiguriert ist. Dadurch wird der Austausch des Modbus-E/A-Scanners gestört.

### Auswahl der Buszyklus-Task

Der Modbus-E/A-Scanner und die Geräte tauschen in jedem Zyklus der ausgewählten Anwendungstask Daten aus.

Die Auswahl der Task erfolgt auf der Registerkarte **Modbus Master E/A-Abbild**. Das Konfigurationsfenster wird angezeigt:

Variable	Abbild	Typ
Modbus_EA/Scanner		IoDrvMo...

= Neue Variable erzeugen   
 = Auf existierende Variable abbilden

Buszyklus-Optionen  
 Buszyklus-Task: MAST

Über den Parameter **Buszyklus-Task** können Sie die Anwendungstask auswählen, die den Scanner verwaltet:

- **Zykluseinstellungen des übergeordneten Busses verwenden:** Ordnet dem Scanner die Anwendungstask zu, die die Steuerung verwaltet.
- **MAST:** Ordnet dem Scanner die MAST-Task zu.
- Eine weitere bestehende Task: Sie können eine vorhandene Task auswählen und mit dem Scanner verbinden. Weitere Informationen über die Anwendungstasks finden Sie im SoMachine-Programmierhandbuch.

Die Zykluszeit der dem Scanner zugeordneten Task muss unter 500 ms liegen.

## Hinzufügen eines Geräts auf dem Modbus Serial-E/A-Scanner

### Einführung

Dieser Abschnitt beschreibt, wie ein Gerät auf dem Modbus-E/A-Scanner hinzugefügt wird.

### Hinzufügen eines Geräts auf dem Modbus-E/A-Scanner

Um ein Gerät zum Modbus-E/A-Scanner hinzuzufügen, wählen Sie den **Allgemeinen Modbus-Slave** im **Hardware-Katalog**, ziehen ihn auf die **Gerätebaumstruktur** und legen ihn auf dem Knoten **Modbus\_IOScanner** der **Gerätebaumstruktur** ab.

Weitere Informationen zum Hinzufügen von Geräten in einem Projekt finden Sie unter:

- Verwenden der Methode Drag&Dop (*siehe SoMachine, Programmierhandbuch*) (Ziehen und Ablegen)
- Verwenden der Kontextmenüs oder Plus-Schaltflächen (*siehe SoMachine, Programmierhandbuch*)

**HINWEIS:** Daraufhin wird in den Feldern %IWx und %QWx der Registerkarte **Modbus serieller Master E/A-Abbild** automatisch die Variable für den Datenaustausch erstellt.

### Auf dem Modbus-E/A-Scanner hinzugefügtes Gerät konfigurieren

So konfigurieren Sie das auf dem Modbus-E/A-Scanner hinzugefügte Gerät:

Schritt	Aktion
1	Doppelklicken Sie in der <b>Gerätebaumstruktur</b> auf <b>Allgemeiner Modbus-Slave</b> . <b>Ergebnis:</b> Das Konfigurationsfenster wird angezeigt. 
2	Geben Sie einen Wert für die <b>Slave-Adresse</b> des Geräts ein (wählen Sie einen Wert zwischen 1 und 247).
3	Wählen Sie einen Wert für das <b>Timeout für Antwort</b> (in ms) aus.


So konfigurieren Sie die **Modbus-Kanäle**:

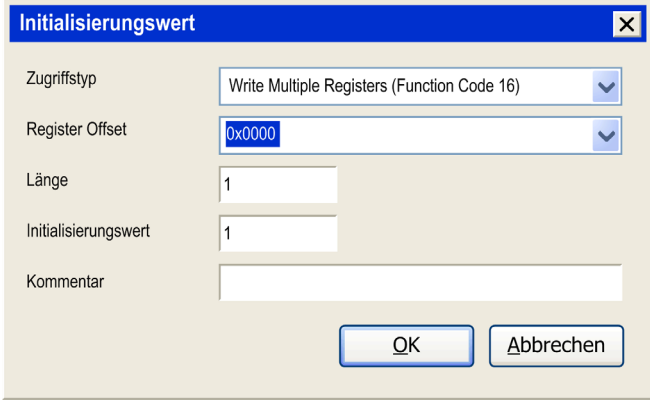
Schritt	Aktion
1	<p>Klicken Sie auf die Registerkarte <b>Modbus Slave-Kanal</b>:</p> 

Schritt	Aktion
2	<p>Klicken Sie auf die Schaltfläche <b>Kanal hinzufügen</b>:</p> <div data-bbox="378 233 1105 971"><p><b>Modbus Kanal</b> <span style="float: right;">✕</span></p><p><b>Kanal</b></p><p>Name <input type="text" value="Kanal 1"/></p><p>Zugriffstyp <input type="text" value="Read/Write Multiple Registers (Function Code 23)"/></p><p>Trigger <input type="text" value="ZYKLISCH"/> Zykluszeit (ms) <input type="text" value="100"/></p><p>Kommentar <input type="text"/></p><p><b>READ Register</b></p><p>Offset <input type="text" value="0x0000"/></p><p>Länge <input type="text" value="1"/></p><p>Fehlerbehandlung <input type="text" value="Letzten Wert beibehalten"/></p><p><b>WRITE Register</b></p><p>Offset <input type="text" value="0x0000"/></p><p>Länge <input type="text" value="1"/></p><p style="text-align: right;"><input type="button" value="OK"/> <input type="button" value="Abbrechen"/></p></div>

Schritt	Aktion
3	<p>Konfigurieren Sie einen Austausch:            Im Feld <b>Kanal</b> können Sie die folgenden Werte hinzufügen:</p> <ul style="list-style-type: none"> <li>● <b>Kanal:</b> Geben Sie einen Namen für den Kanal ein..</li> <li>● <b>Zugriffstyp:</b> Wählen Sie den Typ des Datenaustauschs aus: Lesen oder Schreiben oder Lesen/Schreiben mehrerer Register (d.h. %MW) (<i>siehe Seite 243</i>).</li> <li>● <b>Trigger:</b> Wählen Sie den Trigger für den Datenaustausch aus. Er kann entweder <b>ZYKLISCH</b> mit einem im Feld <b>Zykluszeit (ms)</b> definierten Zeitraum sein oder auf der <b>STEIGENDEN FLANKE</b> an einer booleschen Variablen starten (die boolesche Variable wird dann auf der Registerkarte <b>Modbus Master E/A-Abbild</b> definiert).</li> <li>● <b>Kommentar:</b> Fügen Sie einen Kommentar zu diesem Kanal ein.</li> </ul> <p>Im Feld <b>LESEN-Register</b> (falls es sich bei dem Kanal um einen Lesen- oder einen Lesen/Schreiben-Kanal handelt) können Sie die im Modbus-Slave zu lesenden %MW konfigurieren. Diese werden auf den %IW abgebildet (siehe Registerkarte „Modbus Master E/A-Abbild“):</p> <ul style="list-style-type: none"> <li>● <b>Offset:</b> Offset des zu lesenden %MW. 0 bedeutet, dass es sich bei dem ersten Objekt, das gelesen wird, um %MW0 handelt.</li> <li>● <b>Länge:</b> Anzahl der zu lesenden %MW. Beispiel: Wenn 'Offset' = 2 und 'Länge' = 3, liest der Kanal %MW2, %MW3 und %MW4.</li> <li>● <b>Fehlerbehandlung:</b> Wählen Sie das Verhalten verwandter %IW im Fall eines Kommunikationsverlusts.</li> </ul> <p>Im Feld <b>SCHREIBEN-Register</b> (falls es sich bei dem Kanal um einen Schreiben- oder einen Lesen/Schreiben-Kanal handelt) können Sie die in den Modbus-Slave zu schreibenden %MW konfigurieren. Diese werden auf den %QW abgebildet (siehe Registerkarte „Modbus Master E/A-Abbild“):</p> <ul style="list-style-type: none"> <li>● <b>Offset:</b> Offset der zu schreibenden %MW. 0 bedeutet, dass es sich bei dem ersten Objekt, das geschrieben wird, um %MW0 handelt.</li> <li>● <b>Länge:</b> Anzahl der zu schreibenden %MW. Beispiel: Wenn 'Offset' = 2 und 'Länge' = 3, schreibt der Kanal %MW2, %MW3 und %MW4.</li> </ul>
5	<p>Klicken Sie auf <b>OK</b>, um die Konfiguration des Kanals zu bestätigen.</p> <p><b>HINWEIS:</b> Sie können ebenfalls:</p> <ul style="list-style-type: none"> <li>● Klicken Sie auf <b>Löschen</b>, um den Kanal zu entfernen.</li> <li>● Klicken Sie auf <b>Bearbeiten</b>, um die Parameter eines Kanals zu ändern.</li> </ul>

So konfigurieren Sie den **Modbus-Initialisierungswert**:

Schritt	Aktion
1	<p>Klicken Sie auf die Registerkarte <b>Modbus Slave Init:</b></p>  <p>The screenshot shows a software interface for configuring a Modbus slave. It features a tabbed interface with the following tabs: 'Modbus Slave-Konfiguration', 'Modbus Slave-Kanal', 'Modbus Slave Init:' (which is highlighted with an orange border), and 'Modbus'. Below the tabs is a table with the following columns: 'Zeile', 'Zugriffstyp', 'WRITE-Offset', 'Standardwert', 'Länge', and 'Kommentar'. The table is currently empty. At the bottom of the window, there are four buttons: 'Nach oben', 'Neu...', 'Löschen...', and 'Bearbeiten...'.</p>

Schritt	Aktion
2	<p>Klicken Sie auf <b>Neu</b>, um einen neuen Initialisierungswert zu erstellen:</p>  <p>Das Fenster <b>Initialisierungswert</b> enthält die folgenden Parameter:</p> <ul style="list-style-type: none"> <li>● <b>Zugriffstyp:</b> Wählen Sie den Typ des Datenaustauschs aus: Lesen oder Schreiben oder Lesen/Schreiben mehrerer Register (d.h. %MW) (<i>siehe Seite 243</i>).</li> <li>● <b>Register Offset:</b> Registernummer des zu initialisierenden Registers.</li> <li>● <b>Länge:</b> Anzahl der zu lesenden %MW. Beispiel: Wenn 'Offset' = 2 und 'Länge' = 3, liest der Kanal %MW2, %MW3 und %MW4.</li> <li>● <b>Initialisierungswert:</b> Wert, mit dem die Register initialisiert werden.</li> <li>● <b>Kommentar:</b> Fügen Sie einen Kommentar zu diesem Kanal ein.</li> </ul>
4	<p>Klicken Sie auf <b>OK</b>, um einen neuen <b>Initialisierungswert</b> zu erstellen.</p> <p><b>HINWEIS:</b> Sie können ebenfalls:</p> <ul style="list-style-type: none"> <li>● Klicken Sie auf <b>Nach oben</b>, um die Position eines Werts in der Liste zu ändern.</li> <li>● Klicken Sie auf <b>Löschen</b>, um einen Wert aus der Liste zu entfernen.</li> <li>● Klicken Sie auf <b>Bearbeiten</b>, um die Parameter eines Werts zu ändern.</li> </ul>

So konfigurieren Sie das **Modbus Master E/A-Abbild**:

Schritt	Aktion
1	<p>Klicken Sie auf die Registerkarte <b>Modbus Master E/A-Abbild</b>:</p>
2	<p>Doppelklicken Sie in eine Zelle der Spalte <b>Variable</b>, um ein Textfeld zu öffnen. Geben Sie den Namen einer Variable ein, oder klicken Sie auf die Durchsuchen-Schaltfläche [...], und wählen Sie die Variable über die <b>Eingabehilfe</b> aus.</p>
3	<p>Weitere Informationen zum E/A-Abbild finden Sie im SoMachine - Programmierhandbuch.</p>

## Zugriffstypen

Diese Tabelle beschreibt die verfügbaren Zugriffstypen:

Funktion	Funktionscode	Verfügbarkeit
Read Coils	1	<b>Modbus-Kanal</b>
Read Discrete Inputs	2	<b>Modbus-Kanal</b>
Read Holding Registers (Standardeinstellung für die Kanalkonfiguration)	3	<b>Modbus-Kanal</b>
Read Input Registers	4	<b>Modbus-Kanal</b>
Write Single Coil	5	<b>Modbus-Kanal</b> <b>Initialisierungswert</b>
Write Single Register	6	<b>Modbus-Kanal</b> <b>Initialisierungswert</b>
Write Multiple Coils	15	<b>Modbus-Kanal</b> <b>Initialisierungswert</b>
Write Multiple Registers (Standardeinstellung für die Slave-Initialisierung)	16	<b>Modbus-Kanal</b> <b>Initialisierungswert</b>
Read/Write Multiple Registers	23	<b>Modbus-Kanal</b>

## Modbus-Manager

### Einführung

Der Modbus-Manager wird für das RTU- oder ASCII-Protokoll im Master- oder Slave-Modus verwendet.

### Hinzufügen des Managers

Wenn Sie Ihrer Steuerung einen Modbus-Manager hinzufügen möchten, wählen Sie den **Modbus-Manager** im **Hardwarekatalog** aus, ziehen Sie ihn in die **Gerätebaumstruktur**, und legen Sie ihn dann auf einem der optisch hervorgehobenen Knoten ab.

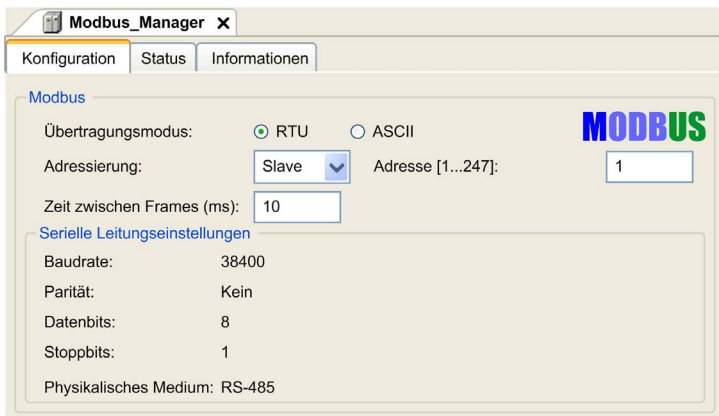
Weitere Informationen zum Hinzufügen von Geräten in einem Projekt finden Sie unter:

- Verwenden der Methode Drag&Dop (*siehe SoMachine, Programmierhandbuch*) (Ziehen und Ablegen)
- Verwenden der Kontextmenüs oder Plus-Schaltflächen (*siehe SoMachine, Programmierhandbuch*)

### Modbus-Manager-Konfiguration

Um den Modbus-Manager Ihrer Steuerung zu konfigurieren, doppelklicken Sie auf **Modbus-Manager** in der **Gerätebaumstruktur**.

Das Konfigurationsfenster für den Modbus-Manager wird angezeigt:



Legen Sie die Parameter wie in der folgenden Tabelle beschrieben fest:

Element	Beschreibung
<b>Übertragungsmodus</b>	Legen Sie den zu verwendenden Übertragungsmodus fest: <ul style="list-style-type: none"> <li>● RTU: Verwendet Binärcodierung und CRC-Fehlerprüfung (8 Datenbits)</li> <li>● ASCII: Meldungen sind im ASCII-Format, LRC-Fehlerprüfung (7 Datenbits)</li> </ul> Dieser Parameter muss für jedes Modbus-Gerät auf der Verbindung identisch sein.
<b>Adressierung</b>	Geben Sie den Gerätetyp an: <ul style="list-style-type: none"> <li>● Master</li> <li>● Slave</li> </ul>
<b>Adresse</b>	Modbus-Adresse des Geräts, wenn Slave ausgewählt ist.
<b>Zeit zwischen Frames (ms)</b>	Zeit zur Vermeidung einer Bus-Kollision. Dieser Parameter muss für jedes Modbus-Gerät auf der Verbindung identisch sein.
<b>Serielle Leitungseinstellungen</b>	Die im Konfigurationsfenster der seriellen Leitung eingestellten Parameter.

### Modbus-Master

Wenn die Steuerung als Modbus-Master konfiguriert ist, werden die folgenden Funktionsbausteine aus der PLCCommunication-Bibliothek unterstützt:

- ADDM
- READ\_VAR
- SEND\_RECV\_MSG
- SINGLE\_WRITE
- WRITE\_READ\_VAR
- WRITE\_VAR

Weitere Informationen finden Sie in den Funktionsbaustein-Beschreibungen (*siehe SoMachine, Modbus- und ASCII-Lese-/Schreibfunktionen, PLCCommunication-Bibliothekshandbuch*) der PLCCommunication-Bibliothek.

### Modbus-Slave

Wenn die Steuerung als Modbus-Slave konfiguriert ist, werden die folgenden Modbus-Anforderungen unterstützt:

Funktionscode Dez (Hex)	Unterfunktion Dez (Hex)	Funktion
1 (1 hex)	–	Digitalausgänge lesen (%Q)
2 (2 hex)	–	Digitaleingänge lesen (%I)
3 (3 hex)	–	Mehrere Register lesen (%MW)
5 (5 Hex)	–	Einzelne Spule schreiben (%M)
6 (6 hex)	–	Einzelnes Register schreiben (%MW)

Funktionscode Dez (Hex)	Unterfunktion Dez (Hex)	Funktion
8 (8 Hex)	–	Diagnose
15 (F hex)	–	Mehrere digitale Ausgänge schreiben (%Q)
16 (10 hex)	–	Mehrere Register schreiben (%MW)
23 (17 hex)	–	Mehrere Register lesen/schreiben (%MW)
43 (2B hex)	14 (E hex)	Geräteidentifikation lesen

Diese Tabelle enthält die Unterfunktionscodes, die von der Modbus-Diagnoseanforderung 08 unterstützt werden:

Unterfunktionscode		Funktion
Dez.	Hex.	
10	0A	Zähler und Diagnoseregister löschen
11	0B	Anzahl von Busnachrichten zurückgeben
12	0C	Anzahl von Buskommunikationsfehlern zurückgeben
13	0D	Anzahl von Busausnahmefehlern zurückgeben
14	0E	Anzahl von Slave-Nachrichten zurückgeben
15	0F	Anzahl von fehlenden Slave-Antworten zurückgeben
16	10	Slave-NAK-Zähler zurückgeben
17	11	Anzahl von ausgelasteten Slaves zurückgeben
18	12	Anzahl von Buszeichenüberläufen zurückgeben

In der folgenden Tabelle werden die Objekte beschrieben, die mit der Anforderung "Geräte-ID lesen" (Basis-ID-Stufe) gelesen werden können:

Objekt-ID	Objektname	Typ	Wert
00 hex	Herstellercode	ASCII-Zeichenfolge	Schneider Electric
01 hex	Produktcode	ASCII-Zeichenfolge	Referenz der Steuerung LMC078CECS20T
02 hex	Haupt-/Nebenrevision	ASCII-Zeichenfolge	aa.bb.cc.dd (entspricht einem Gerätedeskriptor)

Im folgenden Abschnitt werden die Unterschiede zwischen der Modbus-Speicherzuordnung der Steuerung und der Modbus-Zuordnung der HMI erläutert. Wenn Sie Ihre Anwendung nicht so programmieren, dass sie diese Zuordnungsunterschiede erkennt, kommunizieren Steuerung und HMI nicht ordnungsgemäß. Dadurch können falsche Werte in die für den Ausgangsbetrieb zuständigen Speicherbereiche geschrieben werden.

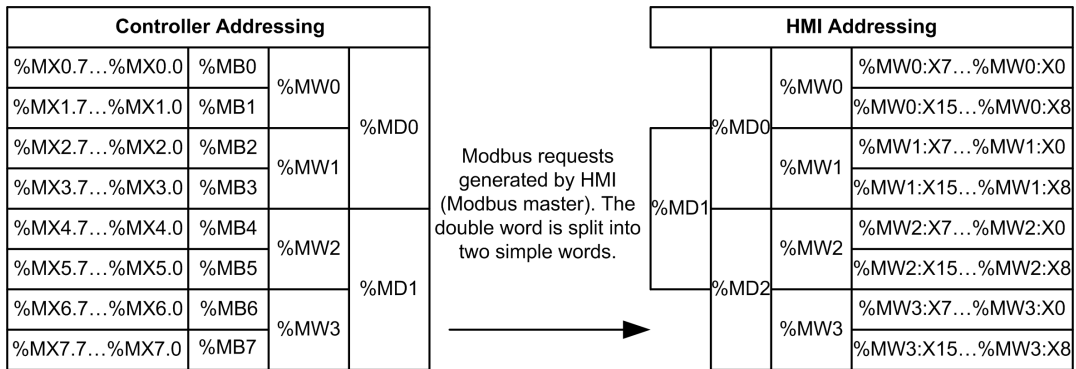
## ⚠️ WARNUNG

### UNBEABSICHTIGTER GERÄTEBETRIEB

Programmieren Sie die Anwendung für eine Übersetzung zwischen dem Modbus-Speicherabbild, das von der Steuerung verwendet wird, und dem Abbild, das von den zugeordneten HMI-Geräten verwendet wird.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

Wenn die Steuerung und die Magelis-HMI über Modbus verbunden sind (HMI ist Master der Modbus-Anforderungen), werden beim Datenaustausch einfache Wort-Anforderungen verwendet. Während Doppelwörter verwendet werden, findet im HMI-Speicher eine Überlappung für einfache Wörter statt, nicht jedoch im Speicher der Steuerung (siehe nachstehendes Diagramm). Um eine Übereinstimmung zwischen dem Speicherbereich der HMI und demjenigen der Steuerung zu erzielen, muss das Verhältnis zwischen den Doppelwörtern im HMI-Speicher und im Steuerungsspeicher 2 betragen.



Im Folgenden sind Beispiele einer Speicherübereinstimmung für Doppelwörter aufgeführt:

- Der %MD2-Speicherbereich der HMI entspricht dem Speicherbereich %MD1 der Steuerung, da von der Modbus-Anforderung dieselben einfachen Wörter verwendet werden.
- Der %MD20-Speicherbereich der HMI entspricht dem Speicherbereich %MD10 der Steuerung, da von der Modbus-Anforderung dieselben einfachen Wörter verwendet werden.

Es folgenden Beispiele von Speicherübereinstimmungen für die Bits:

- Der %MW0:X9-Speicherbereich der HMI entspricht dem Speicherbereich %MX1.1 der Steuerung, da die einfachen Wörter im Steuerungsspeicher in 2 separate Bytes aufgeteilt wurden.

### Hinzufügen eines Modems

Informationen zum Hinzufügen eines Modems zum Modbus-Manager finden Sie unter Hinzufügen eines Modems zu einem Manager (*siehe Seite 249*).

---

## Hinzufügen eines Modems zu einem Manager

### Einführung

Den folgenden Managern kann ein Modem hinzugefügt werden:

- ASCII-Manager
- Modbus-Manager
- SoMachine-Netzwerkmanager

**HINWEIS:** Verwenden Sie Modem TDW-33 (implementiert AT- und A1-Befehle), wenn Sie eine Modemverbindung für SoMachine Network Manager benötigen.

### Hinzufügen eines Modems zu einem Manager

Wenn Sie Ihrer Steuerung ein Modem hinzufügen möchten, wählen Sie das betreffende Modem im **Hardware-Katalog** aus, ziehen Sie es in die **Gerätebaumstruktur** und legen Sie es dann auf einem der optisch hervorgehobenen Knoten ab.

Weitere Informationen zum Hinzufügen von Geräten in einem Projekt finden Sie unter:

- Verwenden der Methode Drag&Dop (*siehe SoMachine, Programmierhandbuch*) (Ziehen und Ablegen)
- Verwenden der Kontextmenüs oder Plus-Schaltflächen (*siehe SoMachine, Programmierhandbuch*)

Weitere Informationen finden Sie unter Modem-Bibliothek (*siehe Modemfunktionen.; Modembibliothek*).



---

# Kapitel 15

## Anschließen von Modicon LMC078 Motion Controller an einen PC

---

### Verbindung der Steuerung mit einem PC

#### Überblick

Zur Übertragung, Ausführung und Überwachung der Anwendungen verbinden Sie die Steuerung mit einem USB-Kabel oder über eine Ethernet-Verbindung mit einem Computer, auf dem SoMachine installiert ist.

### *HINWEIS*

#### **GERÄT NICHT BETRIEBSBEREIT**

Verbinden Sie das Kommunikationskabel immer zuerst mit dem PC, bevor Sie es an die Steuerung anschließen.

**Die Nichtbeachtung dieser Anweisungen kann Sachschäden zur Folge haben.**


#### Verbindung über den Mini-B-USB-Port

**TCSXCNAMUM3P:** Dieses USB-Kabel ist für kurzzeitige Verbindungen wie zum Beispiel für kurze Updates oder das Abrufen von Datenwerten geeignet.

**BMXXCAUSBH045:** Dieses abgeschirmte und geerdete USB-Kabel eignet sich für langfristige Verbindungen.

**HINWEIS:** Sie können jeweils nur eine Steuerung mit dem PC verbinden.

**HINWEIS:** Der LMC078 Motion Controller muss auf der Gateway-Verwaltungskonsole ausgewählt werden. Die Konsole wird durch einen Doppelklick auf das Symbol **Gateway-Verwaltungskonsole**

 im Windows-Benachrichtigungsbereich aufgerufen. Diese Option ist standardmäßig nicht ausgewählt.

Der USB-Mini-B-Port ist eine Programmierschnittstelle, die Sie zum Anschließen eines PC an den USB-Host-Port mithilfe der SoMachine-Software verwenden. Mit einem USB-Standardkabel eignet sich dieser Anschluss für schnelle Aktualisierungen des Programms oder für kurzzeitige Verbindungen zur Durchführung von Wartungsarbeiten und Prüfung von Datenwerten. Die Schnittstelle eignet sich nicht für länger andauernde Verbindungen, wie bei der Inbetriebnahme oder der Überwachung, ohne die Verwendung speziell angepasster Kabel zur Minimierung der Auswirkungen elektromagnetischer Störungen.

## ⚠️ WARNUNG

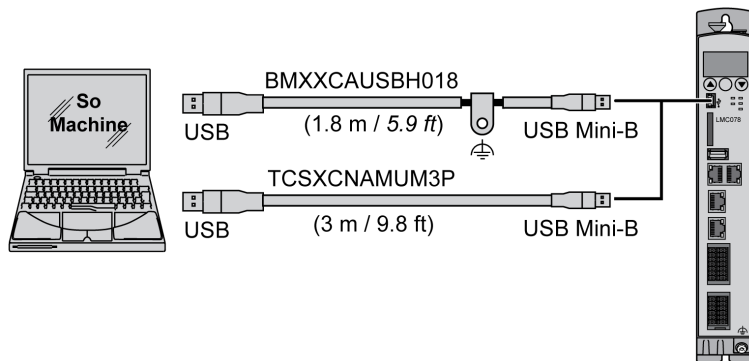
### UNBEABSICHTIGTER GERÄTEBETRIEB ODER FUNKTIONSunFÄHIGE GERÄTE

- Für lang andauernde Verbindungen muss ein geschirmtes und mit der Funktionserde (FE) des Systems verbundenes USB-Kabel verwendet werden, z. B. BMXXCAUSBH0••.
- Schließen Sie nie mehr als eine Steuerung gleichzeitig mithilfe von USB-Verbindungen an.
- Verwenden Sie USB-Ports, sofern vorhanden, nur in nicht explosionsgefährdeten Bereichen.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

Um die Auswirkungen eventueller statischer Entladungen auf die Steuerung zu minimieren, sollte das Kommunikationskabel immer zuerst an den PC angeschlossen werden.

Die folgende Abbildung veranschaulicht die USB-Verbindung mit einem PC:



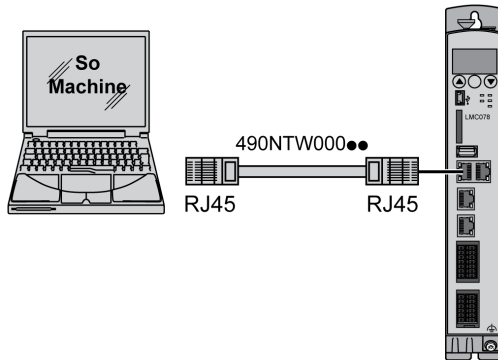
Gehen Sie vor wie folgt, um das USB-Kabel an die Steuerung anzuschließen:

Schritt	Aktion
1	<p><b>1a</b> Bei der Herstellung einer langfristigen Verbindung über ein Kabel des Typs BMXXCAUSBH045 oder über ein anderes geschirmtes Kabel mit Erdanschluss müssen Sie vor dem Anschluss des Kabels an Ihre Steuerung oder den PC die Schirmung an die Funktionserde (FE) oder Schutzterde (PE) Ihres Systems anschließen.</p> <p><b>1b</b> Bei der Herstellung einer kurzzeitigen Verbindung über ein Kabel des Typs TCSXCNAMUM3P oder ein anderes ungeerdetes USB-Kabel fahren Sie mit Schritt 2 fort.</p>
2	Schließen Sie den Stecker des USB-Kabels an den Computer an.
3	Schließen Sie den Mini-B-Stecker Ihres USB-Kabels an die USB-Buchse der Steuerung an.

### Anschluss an einen Ethernet-Port

Sie können die Steuerung auch über ein Ethernet-Kabel an den PC anschließen.

Die folgende Abbildung veranschaulicht die Ethernet-Verbindung mit einem PC:



Gehen Sie vor wie folgt, um die Steuerung mit dem PC zu verbinden:

Schritt	Aktion
1	Schließen Sie das Ethernet-Kabel an den PC ein.
2	Schließen Sie das Ethernet-Kabel an den Ethernet-Port der Steuerung ein.

**HINWEIS:** Die Standard-IP-Adresse (*siehe Seite 178*) lautet: 190.201.100.100



---

# Kapitel 16

## Firmwareaktualisierung

---

### Aktualisieren von Modicon LMC078 Motion Controller-Firmware

#### Einführung

Firmwareaktualisierungen für Modicon LMC078 Motion Controller sind unter der Webadresse <http://www.schneider-electric.com> verfügbar.

Eine Firmwareaktualisierung kann über den **Steuerungs-Assistenten** vorgenommen werden.

Der **Steuerungs-Assistent** stellt zwei verschiedene Möglichkeiten zum Aktualisieren der Firmware bereit:

- Bei der ersten Art der Firmwareaktualisierung wird automatisch die Anwendung in der Steuerung entfernt.
- Bei der zweiten Art der Firmwareaktualisierung wird die Anwendung in der Steuerung nicht entfernt.

#### Firmwareaktualisierung mit automatischer Entfernung der Anwendung

Bei Ausführung einer Firmwareänderung wird das aktuelle Anwendungsprogramm auf dem Gerät gelöscht, einschließlich der Bootapplikation auf der SD-Karte.

### ***HINWEIS***

#### **VERLUST VON ANWENDUNGSDATEN**

- Erstellen Sie eine Sicherungskopie des Anwendungsprogramms auf der Festplatte des PCs, bevor Sie ein Firmware-Upgrade beginnen.
- Stellen Sie das Anwendungsprogramm im Anschluss an ein erfolgreiches Firmware-Upgrade auf dem Gerät wieder her.

**Die Nichtbeachtung dieser Anweisungen kann Sachschäden zur Folge haben.**

Wenn Sie während der Übertragung der Anwendung die Spannungszufuhr zum Gerät trennen oder ein Stromausfall bzw. eine Unterbrechung der Kommunikation auftritt, kann das die Funktionsunfähigkeit des Geräts verursachen. Sollte die Kommunikation unterbrochen werden oder ein Stromausfall auftreten, dann führen Sie die Übertragung erneut durch. Wenn bei der Aktualisierung der Firmware ein Spannungsausfall oder eine Unterbrechung der Kommunikation auftritt, oder wenn eine ungültige Firmware verwendet wird, wird die Maschine betriebsunfähig. In diesem Fall verwenden Sie eine gültige Firmware und starten die Firmwareaktualisierung erneut.

## ***HINWEIS***

### **GERÄT NICHT BETRIEBSBEREIT**

- Unterbrechen Sie die Übertragung des Anwendungsprogramms oder einer Firmware-Änderung nicht, nachdem die Übertragung begonnen hat.
- Wenn die Übertragung aus irgendeinem Grund unterbrochen wurde, starten Sie die Übertragung erneut.
- Versuchen Sie keinesfalls, das Gerät (Logiksteuerung, Bewegungssteuerung, HMI-Controller oder Antrieb) in Betrieb zu nehmen, bevor die Dateiübertragung erfolgreich abgeschlossen wurde.

**Die Nichtbeachtung dieser Anweisungen kann Sachschäden zur Folge haben.**

Starten Sie **SoMachine Central**, und klicken Sie auf **Wartung** → **Steuerungs-Assistent**, um den **Steuerungs-Assistenten** zu öffnen.

Gehen Sie wie folgt vor, um eine vollständige Firmwareaktualisierung einer Steuerung durchzuführen:

Schritt	Aktion
1	Klicken Sie im Dialogfeld <b>Home</b> auf die Schaltfläche <b>Firmware aktualisieren</b> . <b>Ergebnis:</b> Das Dialogfeld <b>Firmware aktualisieren</b> wird geöffnet.
2	Gehen Sie so vor wie im Kapitel <i>Aktualisieren der Firmware</i> des <i>SoMachine Controller Assistant Benutzerhandbuchs</i> beschrieben.

### Firmwareaktualisierung ohne automatische Entfernung der Anwendung

Starten Sie **SoMachine Central**, und klicken Sie auf **Wartung** → **Steuerungs-Assistent**, um den **Steuerungs-Assistenten** zu öffnen.

Um für eine Steuerung eine vollständige Firmwareaktualisierung durchzuführen, ohne die Bootapplikation und Daten zu ersetzen, gehen Sie wie folgt vor:

Schritt	Aktion
1	Klicken Sie im Dialogfeld <b>Home</b> auf die Schaltfläche <b>Images verwalten</b> . <b>Ergebnis:</b> Das Dialogfeld <b>Images verwalten</b> wird angezeigt.
2	Klicken Sie auf <b>Lesen von...</b> <b>Ergebnis:</b> Das Dialogfeld <b>Steuerungsauswahl</b> wird geöffnet.
3	Wählen Sie den gewünschten Verbindungstyp und die gewünschte Steuerung aus und klicken Sie auf die Schaltfläche <b>Lesen</b> . <b>Ergebnis:</b> Das Image wird von der Steuerung an den Computer übertragen. Nach erfolgreichem Abschluss des Vorgangs werden Sie zum Dialogfeld <b>Home</b> weitergeleitet.
4	Klicken Sie auf die Schaltfläche <b>Neu / Bearbeiten...</b> und dann auf <b>Firmware aktualisieren...</b> <b>Ergebnis:</b> Das Dialogfeld zur Aktualisierung der Firmware wird geöffnet.
5	Führen Sie die einzelnen Schritte zur Aktualisierung der Firmware im aktuellen Image aus (Änderungen werden nur im Image auf Ihrem Computer wirksam). Im letzten Schritt können Sie eine Sicherungskopie des von der Steuerung gelesenen Image erstellen. <b>Ergebnis:</b> Nach der Aktualisierung der Firmware wird das Dialogfeld <b>Nächste Aktion auswählen</b> angezeigt.
6	Klicken Sie im Dialogfeld <b>Nächste Aktion auswählen</b> auf die Schaltfläche <b>Auf Steuerung schreiben...</b> <b>Ergebnis:</b> Das Dialogfeld <b>Steuerungsauswahl</b> wird geöffnet.
7	Wählen Sie den gewünschten Verbindungstyp und die gewünschte Steuerung aus und klicken Sie auf die Schaltfläche <b>Schreiben</b> . <b>Ergebnis:</b> Das Image wird vom Computer in die Steuerung übertragen. Nach der Übertragung kehren Sie zum Dialogfeld <b>Home</b> zurück.

Weitere Informationen über die Firmwareaktualisierung und die Erstellung einer neuen SD-Karte mit Firmware finden Sie im *SoMachine Controller Assistant Benutzerhandbuch*.





## Überblick

In diesem Anhang werden die Dokumente angeführt, die für das technische Verständnis des Modicon LMC078 Motion Controller-Programmierhandbuchs erforderlich sind.

## Inhalt dieses Anhangs

Dieser Anhang enthält die folgenden Kapitel:

Kapitel	Kapitelname	Seite
A	Vorgehensweise zur Änderung der IP-Adresse der Steuerung	261
B	Diagnosemeldungen	265
C	LMC078 Sercos3-Bibliothek	281
D	Funktionen zum Abrufen/Einrichten der Konfiguration der seriellen Leitung in einem Anwenderprogramm	309
E	SPS-Leistung	315



---

# Anhang A

## Vorgehensweise zur Änderung der IP-Adresse der Steuerung

---

### changeIPAddress: Ändern der IP-Adresse der Steuerung

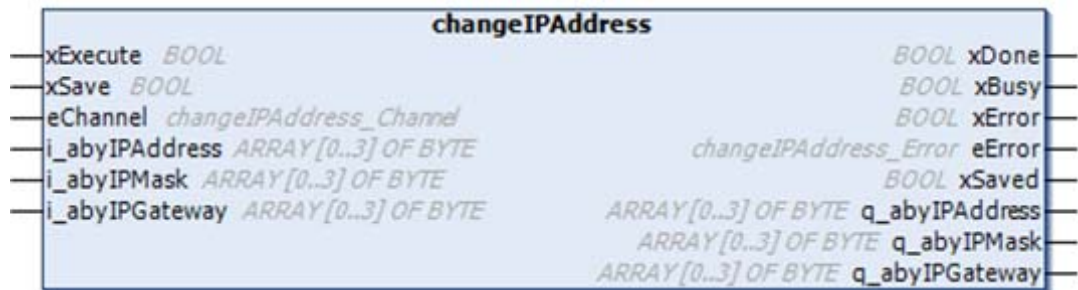
#### Beschreibung des Funktionsbausteins

Der Funktionsbaustein `changeIPAddress` bietet die Möglichkeit, die IP-Adresse einer Steuerung, die Subnetzmaske und die Gateway-Adresse dynamisch zu ändern. Er kann darüber hinaus die IP-Adresse speichern, damit diese auch bei nachfolgenden Neustarts der Steuerung verwendet wird.

**HINWEIS:** Das Ändern der IP-Adresse ist nur möglich, wenn als IP-Modus **Feste IP-Adresse** festgelegt ist. Weitere Informationen finden Sie unter Konfiguration der IP-Adresse (*siehe Seite 175*).

**HINWEIS:** Weitere Informationen zum Funktionsbaustein finden Sie auf der Registerkarte **Dokumentation** des Editors des SoMachine-Bibliotheksverwalters. Informationen zur Verwendung dieses Editors finden Sie im SoMachine - Programmierhandbuch.

#### Grafische Darstellung



## Parameterbeschreibung

Eingänge	Typ	Kommentar
xExecute	BOOL	<ul style="list-style-type: none"> <li>Steigende Flanke: Die Aktion startet.</li> <li>Fallende Flanke: Die Ausgänge werden zurückgesetzt. Wenn eine fallende Flanke eintritt, bevor der Funktionsbaustein seine Aktion abgeschlossen hat, funktionieren die Ausgänge auf normale Weise und werden nur zurückgesetzt, wenn die Aktion abgeschlossen ist oder ein Fehler auftritt. In diesem Fall sind die entsprechenden Ausgangswerte (xDone, xError, iError) an den Ausgängen für genau einen Zyklus vorhanden.</li> </ul>
xSave	BOOL	TRUE: Die Konfiguration wird für nachfolgende Neustarts der Steuerung gespeichert.
eChannel	changeIPAddress_Channel	Der Eingang eChannel ist der zu konfigurierende Ethernet-Port. Je nach Anzahl der auf der Steuerung verfügbaren Ports ist dies einer der 2 Werte ( <i>siehe Seite 263</i> ) in changeIPAddress_Channel (0 oder 1).
i_abyIPAddress	ARRAY[0..3] OF BYTE	Die neue IP-Adresse, die konfiguriert werden soll. Format: 0.0.0.0. <b>HINWEIS:</b> Wenn dieser Eingang auf 0.0.0.0 gesetzt wird, wird die Standard-IP-Adresse ( <i>siehe Seite 178</i> ) der Steuerung konfiguriert.
i_abyIPMask	ARRAY[0..3] OF BYTE	Die neue Subnetzmaske. Format: 0.0.0.0
i_abyIPGateway	ARRAY[0..3] OF BYTE	Die neue Gateway-IP-Adresse. Format: 0.0.0.0

Ausgänge	Typ	Kommentar
xDone	BOOL	TRUE: Wenn IP-Adressen oder wenn Standard-IP-Adressen erfolgreich konfiguriert wurden, weil der Eingang i_abyIPAddress auf 0.0.0.0 gesetzt ist.
xBusy	BOOL	Funktionsbaustein aktiv.
xError	BOOL	<ul style="list-style-type: none"> <li>TRUE: Fehler erkannt, der Funktionsbaustein bricht die Aktion ab.</li> <li>FALSE: Es wurde kein Fehler festgestellt.</li> </ul>
eError	changeIPAddress_Error	Fehlercode des identifizierten Fehlers ( <i>siehe Seite 263</i> ).
xSaved	BOOL	Die Konfiguration wird für die nachfolgenden Neustarts der Steuerung gespeichert.
q_abyIPAddress	ARRAY[0..3] OF BYTE	Aktuelle IP-Adresse der Steuerung. Format: 0.0.0.0.
q_abyIPMask	ARRAY[0..3] OF BYTE	Aktuelle Subnetzmaske. Format: 0.0.0.0.
q_abyIPGateway	ARRAY[0..3] OF BYTE	Aktuelle Gateway-IP-Adresse. Format: 0.0.0.0.

**changeIPAddress\_Channel: Zu konfigurierender Ethernet-Port.**

Der Enumerationsdatentyp `changeIPAddress_Channel` enthält die folgenden Werte:

Enumerator	Wert	Beschreibung
CHANNEL_ETHERNET_NETWORK	0	M241, M251MESC, M258, LMC058, LMC078: <b>Ethernet-Port</b> M251MESE: <b>Ethernet_2-Port</b>
CHANNEL_DEVICE_NETWORK	1	M241: <b>TM4ES4-Ethernet-Port</b> M251MESE: <b>Ethernet_1-Port</b>

**changeIPAddress\_Error: Fehlercodes**

Der Enumerationsdatentyp `changeIPAddress_Error` enthält die folgenden Werte:

Enumerator	Wert	Beschreibung
ERR_NO_ERROR	00 hex.	Kein Fehler erkannt.
ERR_UNKNOWN	01 hex.	Interner Fehler erkannt.
ERR_INVALID_MODE	02 hex.	Die IP-Adresse ist nicht als feste IP-Adresse konfiguriert.
ERR_INVALID_IP	03 hex.	Ungültige IP-Adresse.
ERR_DUPLICATE_IP	04 hex.	Die neue IP-Adresse wird bereits im Netzwerk verwendet.
ERR_WRONG_CHANNEL	05 hex.	Ungültiger Ethernet-Kommunikationsport.
ERR_IP_BEING_SET	06 hex.	IP-Adresse wird bereits geändert.
ERR_SAVING	07 hex.	IP-Adressen wurden aufgrund eines Fehlers nicht gespeichert oder weil kein nicht flüchtiger Speicher vorhanden.



---

# Anhang B

## Diagnosemeldungen

---

### Einführung

In diesem Kapitel werden die Diagnosemeldungen von Modicon LMC078 Motion Controller beschrieben.

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Meldungslogger	266
Diagnosemeldungen	272

## Meldungslogger

### Überblick

Der Meldungslogger protokolliert wichtige Ereignisse auf der Steuerung. Diese Informationen werden in Kategorien sortiert, analysiert und verständlich dargestellt. Wird ein Fehler erkannt, helfen diese Informationen dabei, ein Problem zu lösen oder den Fehler zu lokalisieren.

Um den **Meldungslogger** zu öffnen, wählen Sie die **Tools-Baumstruktur** aus, und doppelklicken Sie auf den Knoten **Meldungslogger**:

Meldungslogger	Nein	Zeitstempel	Typ	Objekt	Instanz	Diag. Code
+ LMCxx8 ...	1	Mi 14.Jul 2010 12:06:28				
+ LMCxx8 ...	2	Mi 14.Jul 2010 12:06:50				
+ LMCxx8 ...	3	Mi 14.Jul 2010 12:07:01				
+ LMCxx8 ...	4	Mi 14.Jul 2010 12:07:10				
+ LMCxx8 ...	5	Mi 14.Jul 2010 12:08:02				

Diese Datenansicht zeigt die Informationen des Meldungsloggers an. Sie können z. B. nacheinander Daten verschiedener Meldungslogger der Steuerung aufrufen und verarbeiten. Sie können auch Dateien des Meldungsloggers hinzufügen, die Sie zuvor gespeichert haben.

Sie können folgende Funktionen nutzen:

- Speichern der angezeigten Meldungslogger als Datei
- Entfernen der angezeigten Meldungslogger aus der Liste

Sie haben außerdem folgende Möglichkeiten:

- Hinzufügen eines Kommentars zu jeder einzelnen Zeile
- Markieren aller Zeilen mit bestimmten Eigenschaften in einer bestimmten Farbe

Diese Optionen stehen direkt über das Kontextmenü (Rechtsklick) zur Verfügung.

Die angezeigte Zeit entspricht der Zeit, zu der das Element diesem Dialogfeld hinzugefügt wurde. Die Nummerierung erfolgt kontinuierlich.

Sie können den ausgewählten Meldungslogger öffnen, indem Sie auf das Pluszeichen (+) links neben dem Logger klicken. Um ihn zu schließen, klicken Sie auf das Minuszeichen (-):

Message logger	No.	Timestamp	Type	Object	Instance	Diag. code	Ext. diagnosis	Message
LMCxx8 r...	1	Wed 02. Apr 2014 08:45:24						
...	500	We Apr/02/2014 08:32:57...	1	LMCxx8	MyCon...	8003		Controller boot finished
...	499	We Apr/02/2014 08:32:56...	1	LMCxx8	MyCon...	8051		LMC 078/AX=8/RAM=500/↑
...	498	We Apr/02/2014 08:32:56...	1	SERC32	Sercos...	8042	CP4	Sercos phase switched
...	497	We Apr/02/2014 08:32:56...	1	SERC32	Sercos...	8042	CP3/1 ms	Sercos phase switched
...	496	We Apr/02/2014 08:32:55...	1	LMCxx8	MyCon...	8970	end	Fast Device Replacement
...	495	We Apr/02/2014 08:32:55...	1	LMCxx8	MyCon...	8970	start	Fast Device Replacement
...	494	We Apr/02/2014 08:32:55...	1	SERC32	Sercos...	8042	CP2/use=0	Sercos phase switched
...	493	We Apr/02/2014 08:32:54...	1	SERC32	Sercos...	8042	CP1/scan=2	Sercos phase switched
...	492	We Apr/02/2014 08:32:53...	1	SERC32	Sercos...	8042	CP0	Sercos phase switched
...	491	We Apr/02/2014 08:32:52...	1	LMCxx8	MyCon...	8002	V01.51.03.03	Controller boot started
...	490	We Apr/02/2014 08:32:14...	1	OBJVER		8016		Controller reset
...	489	We Apr/02/2014 08:31:52...	1	LMCxx8	MyCon...	8003		Controller boot finished
...	488	We Apr/02/2014 08:31:51...	1	LMCxx8	MyCon...	8051		LMC 078/AX=8/RAM=500/↑
...	487	We Apr/02/2014 08:31:51...	1	SERC32	Sercos...	8042	CP4	Sercos phase switched
...	486	We Apr/02/2014 08:31:51...	1	SERC32	Sercos...	8042	CP3/1 ms	Sercos phase switched
...	485	We Apr/02/2014 08:31:50...	1	LMCxx8	MyCon...	8970	end	Fast Device Replacement
...	484	We Apr/02/2014 08:31:50...	1	LMCxx8	MyCon...	8970	start	Fast Device Replacement
...	483	We Apr/02/2014 08:31:50...	1	SERC32	Sercos...	8042	CP2/use=0	Sercos phase switched
...	482	We Apr/02/2014 08:31:49...	1	SERC32	Sercos...	8042	CP1/scan=2	Sercos phase switched
...	481	We Apr/02/2014 08:31:48...	1	SERC32	Sercos...	8042	CP0	Sercos phase switched
...	480	We Apr/02/2014 08:31:47...	1	LMCxx8	MyCon...	8002	V01.51.03.03	Controller boot started

Einige Einträge sind farbig markiert. Dadurch finden Sie schneller Zeilen mit vergleichbaren Eigenschaften im Meldungslogger. Sie können dies über das Kontextmenü anpassen. Wenn Sie eine Datei gespeichert oder gesendet haben und dann erneut öffnen, bleiben die zuletzt ausgewählten Farbeinstellungen erhalten.

### Kommentar bearbeiten

Wählen Sie im Kontextmenü die Option **Kommentar bearbeiten** aus. Sie können für jeden Eintrag einen beliebigen Kommentar hinzufügen. Auf diese Weise stellen Sie zusätzliche Informationen bereit, die auch noch nach längerer Zeit einen Überblick über die erkannten Fehler geben, falls einmal Service erforderlich ist.

### Kopieren

Wählen Sie im Kontextmenü die Option **Kopieren** aus, oder drücken Sie die Tastenkombination **Strg + C**. Sie können den Inhalt der ausgewählten Zeile in die Windows-Zwischenablage speichern und ihn in eine beliebige Verarbeitungsanwendung einfügen.

### Zusätzlichen Meldungslogger von der Steuerung herunterladen

Wählen Sie aus dem Kontextmenü **Zusätzlichen Meldungslogger von der Steuerung herunterladen** aus. Dadurch wird in der Datenansicht ein neuer Meldungslogger erstellt. Es wird eine neue kontinuierliche Nummerierung mit dem aktuellen Zeitstempel erstellt.

### Meldungslogger zurücksetzen

Wählen Sie aus dem Kontextmenü **Meldungslogger zurücksetzen** aus. Dadurch wird in der Datenansicht lokal ein leerer Meldungslogger (ohne Meldungen) erstellt. Der Meldungslogger auf der Steuerung wird gelöscht. Alle Meldungslogger-Einträge der Steuerung werden entfernt.

### Meldungslogger in Datei speichern

Wählen Sie aus dem Kontextmenü **Meldungslogger in Datei speichern** aus. Es wird ein Windows-Standarddialogfeld geöffnet. Geben Sie einen Dateinamen ein, und speichern Sie den ausgewählten Meldungslogger in einem beliebigen Verzeichnis.

### Meldungslogger aus Datei laden

Wählen Sie aus dem Kontextmenü **Meldungslogger aus Datei laden** aus. Es wird ein Windows-Standarddialogfeld geöffnet. Wählen Sie in diesem Dialogfeld den gewünschten Meldungslogger aus, der in einem beliebigen Verzeichnis gespeichert ist. Dadurch wird in der Datenansicht ein neuer Meldungslogger erstellt. Es wird eine neue kontinuierliche Nummerierung ohne aktuellen Zeitstempel erstellt.

### Meldungslogger von Steuerung importieren

Wählen Sie aus dem Kontextmenü **Meldungslogger von Steuerung importieren** aus. Es wird ein Windows-Standarddialogfeld geöffnet. Wählen Sie in diesem Dialogfeld den gewünschten Meldungslogger aus, der in der Steuerung gespeichert ist. Dadurch wird in der Datenansicht ein neuer Meldungslogger erstellt. Es wird eine neue kontinuierliche Nummerierung ohne aktuellen Zeitstempel erstellt.

### Entferne Meldungslogger aus der Liste

Wählen Sie aus dem Kontextmenü **Entferne Meldungslogger aus der Liste**, oder drücken Sie die Taste **Entf**. Dadurch wird der ausgewählte Meldungslogger aus der Liste entfernt.

Sie können mehrere Meldungslogger auswählen, indem Sie die Taste **Strg** gedrückt halten.







## Zeilen markieren

Sie können im Meldungslogger Zeilen mit bestimmten Eigenschaften markieren. Dafür stehen verschiedene Farben zur Auswahl. Z. B. können Sie alle Zeilen vom **Typ 2** in Pink markieren. Oder Sie markieren alle Zeilen mit **Diagnosecode= 8002** in Hellgrün.

Das folgende Kontextmenü zeigt die Konfiguration, die im vorhergehenden Abschnitt beschrieben wurde. Die Auswahlkriterien werden nach der Farbmarkierung aufgeführt.

Hilfe zum Diagnose Code zeigen		F1	MCx00C	LMC_P...	8002	V01
Kommentar bearbeiten			SERC3	SERCO...	8042	CP
Kopieren		Strg+C	SERC3	SERCO...	8042	CP
Zusätzlichen Meldungslogger von der Steuerung herunterladen			SERC3	SERCO...	8042	CP
Meldungslogger zurücksetzen			SERC3	SERCO...	8042	CP
Meldungslogger in Datei speichern			SERC3	SERCO...	8042	CP
Meldungslogger aus Datei laden			SERC3	SERCO...	8042	CP
Meldungslogger von Steuerung importieren			SERC3	SERCO...	8042	CP
Entferne Meldungslogger aus der Liste		Entf	SERC3	SERCO...	8042	CP
<b>Zeilen markieren (Nr. -448)</b>						
Zeile zum Vergleichen auswählen		Einfg				
Nicht vergleichen						
Liste auf-/zuklappen						
431	Mi 14.Jul 2010 12:58:30.876	1				
430	Mi 14.Jul 2010 12:58:29.256	1				
429	Mi 14.Jul 2010 12:58:27.682	1				
428	Mi 14.Jul 2010 12:58:26.000	1				
427	Mi 14.Jul 2010 12:51:42.276	1				
426	Do 08.Jul2010 12:51:37.500	1				

	Typ = 2
	Typ = 3
	Typ = 4
	Diag. Code = 8002
	
	
Markierung aufheben	
Alle Markierungen aufheben	

Wählen Sie aus dem Kontextmenü **Zeilen markieren (xxx=yyy)** aus. Es wird ein Untermenü geöffnet, das bereits einige Farbeinträge mit der entsprechenden Auswahl anzeigt. Hier sind dies, wie im Beispiel beschrieben, **Typ=2** und **Diag.code=8002**.

Sie können Ihre eigenen Auswahlkriterien festlegen. Der ausgewählte Text ist von der zuletzt ausgewählten Spalte vor dem Öffnen des Kontextmenüs abhängig. Gehen Sie vor wie folgt:

Hinzufügen oder Ändern eines Farbeintrags:

Schritt	Aktion
1	Klicken Sie mit der rechten Maustaste auf die Spalte mit dem gewünschten Auswahlkriterium (z. B. Kopfzeile <b>Diag. Code</b> und Zellenwert <b>8014</b> ). <b>Ergebnis:</b> Das Kontextmenü wird angezeigt.
2	Klicken Sie auf <b>Zeilen markieren (Diag. code=8014)</b> . <b>Ergebnis:</b> Das Untermenü mit der Farbauswahl wird geöffnet.
3	Wählen Sie eine Farbe für die Markierung aus, z. B. Blau. <b>Ergebnis:</b> Alle Zeilen mit dem Auswahlkriterium ( <b>Diag. code=8014</b> ) werden blau markiert.

Löschen eines Farbeintrags:

Schritt	Aktion
1	Klicken Sie mit der rechten Maustaste auf die Farbe, die Sie aus dem Meldungslogger entfernen möchten. <b>Ergebnis:</b> Das Kontextmenü wird angezeigt.
2	Klicken Sie auf <b>Zeilen markieren</b> . <b>Ergebnis:</b> Das Untermenü mit der Farbauswahl wird geöffnet.
3	Klicken Sie auf <b>Markierung aufheben</b> . <b>Ergebnis:</b> Alle Zeilenmarkierungen mit dieser Farbe werden aufgehoben.

Löschen aller Farbeinträge:

Schritt	Aktion
1	Klicken Sie mit der rechten Maustaste auf die Farbe, die Sie aus dem Meldungslogger entfernen möchten. <b>Ergebnis:</b> Das Kontextmenü wird angezeigt.
2	Klicken Sie auf <b>Zeilen markieren</b> . <b>Ergebnis:</b> Das Untermenü mit der Farbauswahl wird geöffnet.
3	Klicken Sie auf <b>Alle Markierungen aufheben</b> . <b>Ergebnis:</b> Alle Farbmarkierungen werden entfernt.

**HINWEIS:** Es kann geschehen, dass eine Zeile mehreren gültigen Farbauswahlkriterien entspricht. Z. B. könnten die Farben Grün und Rot dem **Diag.Code=8014** entsprechen. In diesem Fall ist die verwendete Farbe die Farbe, die im Kontextmenü weiter unten aufgeführt wird.

**HINWEIS:** Wenn ein Farbauswahlkriterium mehreren Farben zugeordnet wurde, wird die Markierung mit der höchsten Priorität beim Aufheben der Auswahl zuerst entfernt.

### Zeile zum Vergleichen auswählen

Wählen Sie aus dem Kontextmenü **Zeile zum Vergleichen auswählen**, oder drücken Sie die Taste **Einfg**. Auf diese Weise können Sie die ausgewählte Zeile **fett** markieren, um den Vergleich mit anderen Zeilen zu vereinfachen. Wenn Sie auf eine andere Meldung klicken, wird die erforderliche Zeit zum Markieren in fett in der Statusleiste angezeigt (z. B. **Zeitunterschied: 0.00:03:36.992 von 500(1) bzgl. 481(1)**).

### Nicht vergleichen

Wählen Sie aus dem Kontextmenü **Nicht vergleichen** aus, um eine zuvor ausgewählte Zeile für den Vergleich zurückzusetzen und die Fettmarkierung zu entfernen.

### Liste auf-/zuklappen

Wenn Sie im Kontextmenü **Liste auf-/zuklappen** auswählen, können alle Meldungslogger entweder vollständig zu- oder aufgeklappt werden.

## Diagnosemeldungen

### Klassen der Diagnosemeldungen

In dieser Tabelle werden die Klassen der Diagnosemeldungen beschrieben.

Diagnoseklasse	Bezeichnung	Priorität
4	Fehler erkannt, der zu einem vollständigen Stopp führt	Hoch
3	Fehler erkannt, der zu einem einmaligen Stopp führt (wenn der Fehler durch eine Achse ausgelöst wird)	-
2	Hinweis	-
1	Message	Niedrig
0	Deaktiviert	Keine

### Diagnosemeldungen

In dieser Tabelle werden die Diagnosemeldungen und ihre Klassen beschrieben.

Diagnosecode (DiagCode)	Diagnosemeldung (DiagMsg)	Diagnoseklasse (DiagClass)
8001	Diagnostic acknowledgement	1
8002	Controller boot started	1
8003	Controller boot finished	1
8004	Program started	1
8005	Program automatic start active	1
8006	Program stopped	1
8007	Controller login	1
8008	Controller logout	1
8009	Program reset	1
8010	write file	1
8013	Controller connect to TCP/IP server	1
8014	Controller disconnect from TCP/IP server	1
8015	filesystem <ide0:> repaired	1
8016	Controller reset	1
8017	CANopen emergency message reset	1
8018	CANopen node guarding error resolved	1
8019	CANopen node error info	1
8020	Program cycle check has changed	1
8021	Program cycle check values are changed	1

Diagnosecode (DiagCode)	Diagnosemeldung (DiagMsg)	Diagnoseklasse (DiagClass)
8022	FC_SetTaskPriority() called	1
8023	Controller shutdown	1
8027	File write open	1
8028	File write close	1
8029	UPS OK	1
8030	UPS active -no power	1
8031	UPS power supply OK	1
8032	UPS begin saving retain area	1
8033	UPS retain area saved	1
8034	UPS program tasks terminated	1
8035	UPS active -system shutdown started	1
8036	UPS controller rebooting started	1
8037	Battery low	2
8038	NvRam/RTC power outage detected	2
8042	SERCOS phase switched	1
8043	SERCOS detect configuration	1
8044	SERCOS firmware download	1
8045	File write error detected	1
8046	FPGA firmware download	1
8047	PIC firmware download	1
8048	BT-4 firmware download	1
8051	Controller type	1
8052	SERCOS extended diagnostic (MASTER)	1
8053	UPS active overtemperature	1
8054	Controller temperature out of range	2
8055	Controller message HW monitor	1
8056	Controller power supply low	1
8057	Program online change	1
8059	UPS active -IEC-control task running	1
8060	UPS changing state	1
8100	Motor overload	3
8101	Power stage overtemperature	3
8102	Motor overtemperature	3
8104	Control voltage out of range	3

Diagnosecode (DiagCode)	Diagnosemeldung (DiagMsg)	Diagnoseklasse (DiagClass)
8105	Encoder signal out of range	3
8106	DC bus controller communication not possible	3
8107	Overcurrent	3
8108	DC bus overvoltage	3
8109	DC bus undervoltage	3
8110	Phase missing	3
8111	Shutdown due to tracking deviation	3
8112	SERCOS telegram invalid	3
8113	Braking resistor error detected	3
8114	Device type plate not readable	3
8116	Commutation error detected	3
8117	Motor type plate not readable	3
8119	Power stage short-circuit /ground error detection	3
8120	Power stage overload	3
8121	Braking resistor - overtemperature	3
8122	Shutdown due to velocity limit	3
8123	Safe Torque Off incorrect	3
8125	Motor load high	2
8126	Power stage temperature high	2
8127	Motor temperature high	2
8129	Power stage load high	2
8130	Temperature of braking resistor high	2
8132	Tracking deviation limit exceeded	2
8133	Speed-dependent current reduction	2
8134	External 24 Vdc low	2
8135	DC bus voltage low	2
8136	Safe Torque Off active	2
8137	Motorless	3
8138	Motor/Drive combination not supported	3
8139	DC bus precharge not possible	3
8140	Motor stop time limit exceeded	3
8142	Control board overtemperature	3
8143	Encoder temperature high	2
8144	DC bus short-circuit or ground error	3

Diagnosecode (DiagCode)	Diagnosemeldung (DiagMsg)	Diagnoseklasse (DiagClass)
8146	DC bus overload	3
8153	DC bus discharge not possible	3
8154	Phase L1 missing	2
8155	Phase L2 missing	2
8156	Phase L3 missing	2
8157	DC bus load high	2
8159	DC bus discharge delayed	2
8161	Control board temperature high	2
8163	SERCOS Slave C1D error detected	3
8164	SERCOS C1D man.-specific error detected	3
8165	SERCOS Slave C2D advisory detected	2
8166	SERCOS C2Dman.specific advisory detected	2
8169	SERCOS Slave communication disturbance detected	2
8170	Encoder position not accessible	3
8171	Encoder communication disturbance detected	2
8172	Encoder extended diagnostic error detected	1
8173	Encoder error (track monitoring) detected	1
8177	Power board overtemperature	3
8178	Device internal error detected	3
8179	Braking resistor load high	2
8180	Power board temperature high	2
8181	Fan error detected	2
8182	External 24 Vdc power supply high	1
8183	Device fallback firmware active	3
8184	HW/SW combination not supported	3
8185	Device error detected	3
8186	DC bus voltage high	2
8204	Program cannot be loaded	3
8205	Impermissible parameter value	3
8209	Last boot unsuccessful	3
8300	Program divide by zero	3
8301	coprocessor segment overflow	3
8302	stack error detected	3
8303	general protection error detected	3

Diagnosecode (DiagCode)	Diagnosemeldung (DiagMsg)	Diagnoseklasse (DiagClass)
8304	coprocessor error detected	3
8305	memory limit exceeded	3
8306	arithmetic overflow	3
8307	double execution error detected	3
8308	invalid task state segment	3
8309	no memory segment	3
8310	invalid memory segment adjustment	3
8311	coprocessor division error detected	3
8312	Parameter relocation unsuccessful	3
8313	excessive cycle time overrun	3
8316	NvRam data not valid	2
8317	Program cycle time overrun	2
8318	Program calculated profile deleted	3
8320	incorrect array access	3
8321	division by zero	3
8322	exception in IEC task	3
8323	string too long	3
8324	UPS error detected	3
8325	File corrupt	3
8326	Program function not supported	3
8327	CamTrack invalid Position Source	2
8328	CamTrack invalid Destination	2
8329	CamTrack invalid Bit number	2
8330	Program master job not executable	3
8331	Licensing	3
8332	Licensing	3
8333	EncoderNet receiving data not possible	3
8334	EncoderNet receiving data dist.detected	2
8335	EncoderNet synchronization not possible	3
8336	EncoderNet synchronization disturbance detected	2
8337	Parameter DynIECData value too high	3
8338	UPS battery not charged	3
8339	UPS active -system temperature high	3
8340	Data/parameter out of range	3

Diagnosecode (DiagCode)	Diagnosemeldung (DiagMsg)	Diagnoseklasse (DiagClass)
8341	CamTrack invalid Position type	2
8400	Program diagnostic message class 0	0
8401	Program diagnostic message class 1	1
8402	Program diagnostic message class 2	2
8403	Program diagnostic message class 3	3
8404	Program diagnostic message class 4	4
8406	IEC diagnostic message class 1	1
8501	SERCOS slave not found	3
8502	SERCOS loop not closed	1
8503	SERCOS service channel error detected	3
8504	SERCOS read cycle overflow	3
8505	SERCOS Master communication disturbance detected	2
8506	SERCOS Master communication not possible	3
8507	SERCOS write cycle overflow	3
8508	SERCOS run-up not possible	3
8509	SERCOS slave SW not supported	3
8510	SERCOS Interrupt lost	3
8511	CPU time overflow	3
8512	SERCOS incorrect device type	3
8517	SERCOS addressing not unique	3
8518	SERCOS too many real slaves	3
8600	Master Encoder communication not possible	3
8601	Master Encoder signal out of range	3
8610	Async FB error	3
8611	Copied Async FB. Use a reference	3
8612	Async FB declared as retain/persistent	3
8613	Asynchr. Job-Timeout-Fehler	3
8700	CAN layer 2 driver error detected	3
8701	CAN layer2 initialization error detected	3
8702	CAN layer2 single error detected	3
8703	CAN layer2 errors reach advisory limit	3
8704	CAN layer2 switched passive	3
8705	CAN layer 2 system error detected	3
8706	CAN layer2 errors below advisory limit	1

Diagnosecode (DiagCode)	Diagnosemeldung (DiagMsg)	Diagnoseklasse (DiagClass)
8707	CAN layer2 switched active	1
8710	communication error detected	3
8720	no module found	3
8722	no cyclic telegram	3
8723	no PROFIBUS config data	3
8725	firmware of the module was replaced	1
8726	firmware of the module is incorrect	3
8730	incorrect master parameter data	3
8131	automatic bus deactivation	3
8732	slave not responding	3
8733	unrecoverable bus error detected	3
8734	Bus short circuit detected	3
8735	reject bus telegrams	3
8736	no I/O data exchange with slave	3
8737	double IEC address assigned	3
8738	Configuration I/O data > permissible I/O area	3
8739	double PROFIBUS address assigned	3
8750	CANopen node does not exist	3
8751	CANopen node not configured	1
8752	no CANopen EDS file exists	3
8753	initialisation CANopen module unsuccessful	3
8754	CANopen Emergency Message	3
8755	CANopen node guarding error detected	3
8756	CANopen DPM access timeout	3
8757	CANopen configuration error detected	3
8758	Application object Size not supported	3
8759	Application object maximum count limit reached	3
8780	Encoder output frequency > 1MHz	3
8781	Master Encoder no connection	3
8782	Master Encoder signal out of range	3
8785	Hardware component error detected	3
8786	Asynchronous to SERCOS bus	3
8787	configuration error detected	3
8788	Wiring error detected	2

Diagnosecode (DiagCode)	Diagnosemeldung (DiagMsg)	Diagnoseklasse (DiagClass)
8789	PacNet communication disturbance detected	2
8790	Module error detected	3
8791	TM5/TM7 module error detected on TM5NS31	3
8800	Insufficient working memory	3
8826	PIC update not possible	4
8827	Controller power-off/hardware monitor	3
8828	Library error detected	1
8903	Software error detected (class 3)	1
8904	Software error detected (class 4)	4
8905	FC_UserRefGeneratorStart not possible	4
8906	ControlMode invalid	3
8907	Encoder interface invalid	3
8908	Unintended motor reaction detected	3
8909	Motor type plate parameter invalid	3
8910	Reference value invalid	3
8910	Name too long	3
8957	SERCOS bus topology changed	2
8958	Encoder communication not possible	3
8959	Mains contactor error detected	3
8960	Invalid mains voltage mode setting	2
8961	Phase missing	2
8963	NRT IPAddr not in IPAddressRangeStatic	2
8964	NRT IPAddressRangeDynamic is insufficient	2
8965	NRT IP parameter read not possible	2
8966	NRT IP parameter write not possible	2
8967	NRT IP parameter device different	2
8968	NRT network overlapping detected	2
8969	Motor cable not connected	3
8970	Fast Device Replacement	1
8971	Fast Device Replacement not successful	2
8972	NRT gateway not in network	2
8973	Program download	1
8974	Brake voltage too low	3
8975	Motor commutation invalid	2

Diagnosecode (DiagCode)	Diagnosemeldung (DiagMsg)	Diagnoseklasse (DiagClass)
8976	Mains phases wiring not correct	3
8977	Motor temp. monitoring disabled	2
8978	InverterEnableConfig invalid	3
8979	STO_A and STO_B different levels	3
8980	Braking resistor not connected	3
8981	Bootloader update	1
8982	Device state	1
8983	DC bus precharge active	1
8984	DC bus precharge complete	1
8990	Firmware update not possible	1
8991	Data transfer invalid	1
8992	Braking resistor short circuit	3
8993	Last device on SERCOS port	1
8994	Invalid ProducerCycleTime	3
8995	Update motor type plate	1
8996	Update motor type plate not successful	3
8997	Motor identification invalid	3
8998	TM5/TM7 supply voltage low	3
8999	TM5/TM7 supply voltage advisory	1

---

# Anhang C

## LMC078 Sercos3-Bibliothek

---

### Einführung

In diesem Kapitel wird die LMC078 Sercos3-Bibliothek beschrieben.

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Abschnitte:

Abschnitt	Thema	Seite
C.1	Datentypen	282
C.2	Sercos-Funktionen	288
C.3	Asynchrone Sercos-Funktionsbausteine	302

# Abschnitt C.1

## Datentypen

---

### Überblick

In diesem Abschnitt werden die in der LMC078 Sercos3-Bibliothek enthaltenen Datentypen beschrieben.

### Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
ST_SercosConfiguration-Datentyp	283
ST_SercosConfigurationDevice-Datentyp	284
ET_Sercos3CmdType-Datentyp	286
ET_Sercos3IDNType-Datentyp	287

## ST\_SercosConfiguration-Datentyp

### Einführung

Die ST\_SercosConfiguration-Struktur speichert die allgemeinen Werte des Sercos-Bus.

Dieser Datentyp wird mit den folgenden Funktionen verwendet:

- FC\_SercosGetConfiguration
- FC\_SercosScanConfiguration

### Datentypstruktur

In der folgenden Tabelle werden die Parameter der ST\_SercosConfiguration-Schnittstelle beschrieben.

Aufzählung	Typ	Beschreibung
uiNumberOfEntries	UINT	Anzahl der gescannten SercosGeräte auf dem Sercos-Bus.
uiNumberOfPhysicalDevices	UINT	Anzahl der gefundenen Sercos-Geräte auf dem Sercos-Bus, die im SoMachine-Projekt konfiguriert sind.
uiPhaseRunUpCount	UINT	Zählt die Sercos-Phasenstartvorgänge.
iCurrentPhase	INT	Aktuelle Phase, die auf dem Sercos-Bus ausgeführt wird.
astDevices	ARRAY [0...254] von ST_SercosConfigurationDevice	Array mit einer spezifischen Beschreibung der Sercos-Geräte, die auf dem Sercos-Bus gefunden wurden.

## ST\_SercosConfigurationDevice-Datentyp

### Einführung

Die ST\_SercosConfigurationDevice-Struktur speichert die spezifische Beschreibung eines Sercos-Geräts, das auf dem Sercos-Bus gefunden wurde.

Dieser Datentyp wird vom ST\_SercosConfiguration-Datentyp verwendet.

Dieser Datentyp wird mit den folgenden Funktionen verwendet:

- FC\_SercosGetConfiguration
- FC\_SercosScanConfiguration

### Datentypstruktur

In der folgenden Tabelle werden die Parameter der ST\_SercosConfigurationDevice-Schnittstelle beschrieben.

Aufzählung	Typ	Beschreibung
stLogicalAddress	ST_LogicalAddress	Logische Adresse des Sercos-Slave.
uiVendorCode	UINT	Herstellercode des Geräts.
sVendorDeviceId	STRING(40)	Sercos-Geräte-ID.
sName	STRING(80)	Der Name des Geräts in der Steuerungskonfiguration.
sPowerSupply	STRING(40)	Aktuell zugewiesene Spannungsversorgung.
udiWorkingMode	UDINT	Betriebsmodus des Geräts.
udiWorkingState	UDINT	Aktuell aktiver Modus.
udiIdentificationMode	UDINT	Identifizierungsmodus des Geräts.
uiTopologyAddress	UINT	Physische Position des Geräts im Sercos-Ring.
uiConfiguredTopologyAddress	UINT	Topologische Adresse auf dem Sercos-Bus, der das Gerät zugewiesen ist.
uiSercosAddress	UINT	Sercos-Adresse, der das Gerät zugewiesen ist. Die Sercos-Adresse wird während des Startvorgangs des Sercos-Bus automatisch vom Master zugewiesen, falls sich das Gerät nicht im IdentificationMode SercosAddress befindet.
uiConfiguredSercosAddress	UINT	Konfigurierte Sercos-Adresse.
sSerialNumberController	STRING(80)	Seriennummer einer Motorsteuerung auf dem Sercos-Bus.

Aufzählung	Typ	Beschreibung
sConfiguredSerialNumberController	STRING(80)	Seriennummer einer Motorsteuerung auf dem Sercos-Bus, der der Antrieb zugewiesen ist.
sSerialNumberMotor	STRING(40)	Seriennummer des Servomotors.
sConfiguredSerialNumberMotor	STRING(40)	Konfigurierte Motor-Seriennummer.
sApplicationType	STRING(40)	Anwendungstyp.
sConfiguredApplicationType	STRING(40)	Konfigurierter Anwendungstyp.
sControllerType	STRING(40)	Gerätetyp.
sMotorType	STRING(40)	Name des Motors.
sFW_Version	STRING(40)	Firmwareversion des Geräts. Wenn das Gerät in der Steuerungskonfiguration virtuell (nicht real) ist, lautet der Parameterwert <LMC version>.virtual.
sHW_Version	STRING(40)	Hardwareversion des Geräts. Wenn das Gerät in der Steuerungskonfiguration virtuell (nicht real) ist, lautet der Parameterwert <LMC version>.virtual.
sBootloaderVersion	STRING(40)	Versionsname des Geräte-Bootloaders.
sFPGA_Version	STRING(40)	FPGA-Version des Geräts.
sIPAddress	STRING(15)	Definierte IP-Adresse des Geräts.
sSubnetmask	STRING(15)	Definierte Subnetzmaske des Geräts.
sGateway	STRING(15)	Definiertes Gateway des Geräts.
sMACAddress	STRING(17)	MAC-Adresse des Geräts.

## ET\_Sercos3CmdType-Datentyp

### Einführung

Der Datentyp ET\_Sercos3CmdType wird mit den folgenden Funktionsbausteinen verwendet, um Sercos-Befehle festzulegen:

- FB\_SercosReadServiceDataAsync
- FB\_SercosWriteServiceDataAsync

Lesen- und Schreibbefehle verwenden die Einträge 0 bis 6, während Prozedurbefehle die Einträge 7 bis 10 verwenden.

### Datentypstruktur

In der folgenden Tabelle wird der Inhalt der ET\_Sercos3CmdType-Aufzählung beschrieben:

Aufzählung	Typ	Initialwert	Beschreibung
ReadWriteName	WORD	0	Einen Namen lesen oder schreiben.
ReadWriteAttribute	WORD	1	Ein Attribut lesen oder schreiben.
ReadWriteUnit	WORD	2	Eine Einheit lesen oder schreiben.
ReadWriteMinValue	WORD	3	Einen Mindestwert lesen oder schreiben.
ReadWriteMaxValue	WORD	4	Einen Höchstwert lesen oder schreiben.
ReadWriteUserData	WORD	5	Benutzerdaten lesen oder schreiben.
ReadWriteVarLength	WORD	6	Eine variable Länge lesen oder schreiben.
ExecuteCommand	WORD	7	Einen Befehl ausführen.
ExecuteCommandStart	WORD	8	Die Ausführung eines Befehls starten.
ExecuteCommandCheck	WORD	9	Den Zustand des ausgeführten Befehls überprüfen.
ExecuteCommandStop	WORD	10	Die Ausführung eines Befehls beenden.

## ET\_Sercos3IDNType-Datentyp

### Einführung

Der Datentyp ET\_Sercos3IDNType wird mit den folgenden Funktionsbausteinen verwendet, um den Sercos-IDN-Typ (Standard oder Proprietär) festzulegen:

- FB\_SercosProcedureCommandAsync
- FB\_SercosReadServiceDataAsync
- FB\_SercosWriteServiceDataAsync

### Datentypstruktur

In der folgenden Tabelle wird der Inhalt der ET\_Sercos3IDNType-Aufzählung beschrieben:

Aufzählung	Typ	Initialwert	Beschreibung
ParameterType_P	WORD	1	Proprietäre IDN (P-x-xxxx.x.x).
ParameterType_S	WORD	2	Standard-IDN (S-x-xxxx.x.x).

# Abschnitt C.2

## Sercos-Funktionen

---

### Überblick

Dieser Abschnitt beschreibt die Sercos-Funktionen.

### Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
FC_SercosGetConfiguration-Funktion	289
FC_SercosReadServiceData-Funktion	290
FC_SercosReadServiceDataByTopAddr-Funktion	293
FC_SercosScanConfiguration-Funktion	296
FC_SercosWriteServiceData-Funktion	298
FC_SercosWriteServiceDataByTopAddr-Funktion	300

## FC\_SercosGetConfiguration-Funktion

### Beschreibung der Funktion

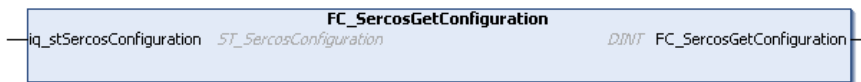
Die Funktion `FC_SercosGetConfiguration` wird verwendet, um eine Liste aller mit dem Sercos-Bus verbundenen Geräte abzurufen, auch derer, die nicht konfiguriert sind.

**HINWEIS:** Der Sercos-Bus muss mindestens einmal Phase 2 erreicht haben.

Die Funktion gibt die Sercos-Geräte aus, die am Ende des Wechsels zu Phase 2 festgelegt werden.

Die gescannten Geräte werden in einer Struktur ausgegeben. Wenn der Sercos-Bus niemals Phase 2 erreicht hat, gibt die Funktion einen Fehler aus.

### Grafische Darstellung



### Darstellung in AWL (IL) und ST

Sie finden eine allgemeine Darstellung in AWL (IL) oder ST im Kapitel *Darstellung von Funktionen und Funktionsbausteinen*.

### Beschreibung der E/A-Variablen

In der folgenden Tabelle wird die Eingangsvariable beschrieben:

Eingangs-	Typ	Kommentar
<code>iq_stSercosConfiguration</code>	<code>ST_SercosConfiguration</code>	Struktur, die auf dem Sercos-Bus konfiguriert wird.

In der folgenden Tabelle wird die Ausgangsvariable beschrieben:

Ausgang	Typ	Kommentar
<code>FC_SercosGetConfiguration</code>	DINT	Siehe Tabelle mit den Beschreibungen der Ausgangswerte unten.

In der folgenden Tabelle wird der Ausgangswert beschrieben:

Wert	Beschreibung
0	Die Funktion wurde erfolgreich ausgeführt.
-1	Fehler erkannt.
-2	Der Sercos-Bus hat niemals Phase 2 erreicht.

## FC\_SercosReadServiceData-Funktion

### Beschreibung der Funktion

Die Funktion `FC_SercosReadServiceData` liest über Sercos Dienstdaten, die für das Debugging verwendet werden.

### ⚠️ WARNUNG

#### UNBEABSICHTIGTER GERÄTEBETRIEB

Verwenden Sie diese Funktion nur nach Absprache mit der Anwendungsabteilung von Schneider Electric.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

### Grafische Darstellung



### Darstellung in AWL (IL) und ST

Sie finden eine allgemeine Darstellung in AWL (IL) oder ST im Kapitel *Darstellung von Funktionen und Funktionsbausteinen*.

## Beschreibung der E/A-Variablen

In der folgenden Tabelle werden die Eingangsvariablen beschrieben:

Eingangs-	Typ	Kommentar
i_stLogAddr	ST_LogicalAddress	Logische Adresse des Sercos-Geräts.
i_dwIDN	DWORD	Datendienst IDN.
i_wType	WORD	Datendiensttyp. <ul style="list-style-type: none"> <li>● 0: Name lesen</li> <li>● 1: Attribut lesen</li> <li>● 2: Einheit lesen</li> <li>● 3: Mindestwert lesen</li> <li>● 4: Höchstwert lesen</li> <li>● 5: Benutzerdaten lesen</li> <li>● 6: Benutzerdaten mit variabler Länge lesen</li> <li>● 7: Befehl ausführen</li> </ul>
i_dwPointerToData	DWORD	Zeiger auf den Datenspeicher, gefüllt mit der tatsächlichen Länge der Datenbytes.
i_wMaxDataLen	WORD	Tatsächliche Länge der zu lesenden Datenbytes.

In der folgenden Tabelle werden die Ein-/Ausgangsvariablen beschrieben:

Eingangs-	Typ	Kommentar
iq_uiReadDataLen	UINT	Variable zum Auslesen der Datenlänge des Sercos-Parameters (uwType=6).
iq_uiMaxReadDataLen	UINT	Variable zum Auslesen der Datenhöchstlänge des Sercos-Parameters (uwType=6).

In der folgenden Tabelle wird die Ausgangsvariable beschrieben:

Ausgang	Typ	Kommentar
FC_SercosReadServiceData	DINT	Siehe Tabelle mit den Beschreibungen der Ausgangswerte unten.

In der folgenden Tabelle wird der Ausgangswert beschrieben:

Wert	Beschreibung
0	Die Funktion wurde erfolgreich ausgeführt.
-1	Die logische Adresse ist ungültig.
-431	Während der Dienstanfrage wurde ein Fehler entdeckt (z. B. Timeout).
-445	Dienst-Timeout.
-461	In der aktuellen Phase wird das Lesen von Parametern über den Servicekanal nicht unterstützt.
-462	Das adressierte Gerät unterstützt die Funktion <code>ServiceDataRead</code> nicht.
-464	Ungültiger Service-Transfer.

## FC\_SercosReadServiceDataByTopAddr-Funktion

### Beschreibung der Funktion

Die Funktion FC\_SercosReadServiceDataByTopAddr liest Dienstdaten über den Sercos-Servicekanal vom adressierten Gerät.

### **⚠️ WARNUNG**

#### **UNBEABSICHTIGTER GERÄTEBETRIEB**

Verwenden Sie diese Funktion nur nach Absprache mit der Anwendungsabteilung von Schneider Electric.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

### Grafische Darstellung



### Darstellung in AWL (IL) und ST

Sie finden eine allgemeine Darstellung in AWL (IL) oder ST im Kapitel *Darstellung von Funktionen und Funktionsbausteinen*.

## Beschreibung der E/A-Variablen

In der folgenden Tabelle werden die Eingangsvariablen beschrieben:

Eingangs-	Typ	Kommentar
i_wTopoAddr	WORD	Topologische Adresse des Sercos-Geräts.
i_dwIDN	DWORD	Datendienst IDN.
i_wType	WORD	Datendiensttyp. <ul style="list-style-type: none"> <li>● 0: Name lesen</li> <li>● 1: Attribut lesen</li> <li>● 2: Einheit lesen</li> <li>● 3: Mindestwert lesen</li> <li>● 4: Höchstwert lesen</li> <li>● 5: Benutzerdaten lesen</li> <li>● 6: Benutzerdaten mit variabler Länge lesen</li> <li>● 7: Befehl ausführen</li> </ul>
i_dwPointerToData	DWORD	Zeiger auf den Datenspeicher, gefüllt mit der tatsächlichen Länge der Datenbytes.
i_wMaxDataLen	WORD	Tatsächliche Länge der zu lesenden Datenbytes.

In der folgenden Tabelle werden die Ein-/Ausgangsvariablen beschrieben:

Eingangs-	Typ	Kommentar
iq_uiReadDataLen	UINT	Variable zum Auslesen der Datenlänge des Sercos-Parameters (uwType=6).
iq_uiMaxReadDataLen	UINT	Variable zum Auslesen der Datenhöchstlänge des Sercos-Parameters (uwType=6).

In der folgenden Tabelle wird die Ausgangsvariable beschrieben:

Ausgang	Typ	Kommentar
FC_SercosReadServiceDataByTopAddr	DINT	Siehe Tabelle mit den Beschreibungen der Ausgangswerte unten.

In der folgenden Tabelle wird der Ausgangswert beschrieben:

Wert	Beschreibung
0	Die Funktion wurde erfolgreich ausgeführt.
-1	Die topologische Adresse ist ungültig.
-431	Während der Dienstanfrage wurde ein Fehler entdeckt (z. B. Timeout).
-445	Dienst-Timeout.
-461	In der aktuellen Phase wird das Lesen von Parametern über den Servicekanal nicht unterstützt.
-462	Das adressierte Gerät unterstützt die Funktion <code>ServiceDataRead</code> nicht.
-464	Ungültiger Service-Transfer.

## FC\_SercosScanConfiguration-Funktion

### Beschreibung der Funktion

Die Funktion `FC_SercosScanConfiguration` wird verwendet, um eine Liste aller mit dem Sercos-Bus verbundenen Geräte abzurufen, auch derer, die nicht konfiguriert sind.

Der Sercos-Bus wird von dieser Funktion automatisch gescannt.

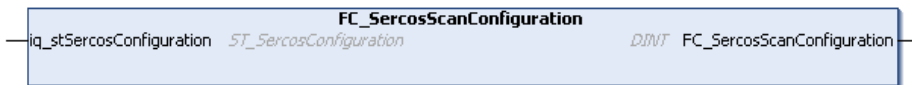
Wenn der Sercos-Bus sich in einer Phase unter 2 befindet, wechselt die Funktion zu Phase 2.

In Phase 2 wird der Sercos-Bus gescannt. Wenn der Sercos-Bus gescannt wurde, wird er in die anfängliche Phase zurückversetzt.

Wenn sich der Sercos-Bus in Phase 2 oder höher befindet, werden die gescannten Geräte ausgegeben.

Die gescannten Geräte werden in einer Struktur ausgegeben.

### Grafische Darstellung



### Darstellung in AWL (IL) und ST

Sie finden eine allgemeine Darstellung in AWL (IL) oder ST im Kapitel *Darstellung von Funktionen und Funktionsbausteinen*.

### Beschreibung der E/A-Variablen

In der folgenden Tabelle wird die Ein-/Ausgangsvariable beschrieben:

Eingangs-	Typ	Kommentar
<code>iq_stSercosConfiguration</code>	<code>ST_SercosConfiguration</code>	Struktur, die auf dem Sercos-Bus konfiguriert wird.

In der folgenden Tabelle wird die Ausgangsvariable beschrieben:

Ausgang	Typ	Kommentar
<code>FC_SercosScanConfiguration</code>	DINT	Siehe Tabelle mit den Beschreibungen der Ausgangswerte unten.

In der folgenden Tabelle wird der Ausgangswert beschrieben:

<b>Wert</b>	<b>Beschreibung</b>
0	Die Funktion wurde erfolgreich ausgeführt.
-1	Fehler erkannt.
-2	Der Sercos-Bus kann Phase 2 nicht erreichen.

## FC\_SercosWriteServiceData-Funktion

### Beschreibung der Funktion

Die Funktion `FC_SercosWriteServiceData` schreibt über Sercos Dienstdaten, die für das Debugging verwendet werden.

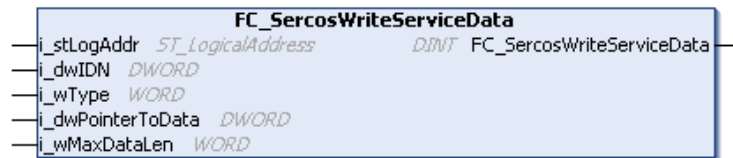
### WARNUNG

#### UNBEABSICHTIGTER GERÄTEBETRIEB

Verwenden Sie diese Funktion nur nach Absprache mit der Anwendungsabteilung von Schneider Electric.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

### Grafische Darstellung



### Darstellung in AWL (IL) und ST

Sie finden eine allgemeine Darstellung in AWL (IL) oder ST im Kapitel *Darstellung von Funktionen und Funktionsbausteinen*.

## Beschreibung der E/A-Variablen

In der folgenden Tabelle werden die Eingangsvariablen beschrieben:

Eingangs-	Typ	Kommentar
i_stLogAddr	ST_LogicalAddress	Logische Adresse des Sercos-Geräts.
i_dwIDN	DWORD	Datendienst IDN.
i_wType	WORD	Datendiensttyp. <ul style="list-style-type: none"> <li>● 0: Name schreiben</li> <li>● 1: Attribut schreiben</li> <li>● 2: Einheit schreiben</li> <li>● 3: Mindestwert schreiben</li> <li>● 4: Höchstwert schreiben</li> <li>● 5: Benutzerdaten schreiben</li> <li>● 6: Benutzerdaten mit variabler Länge schreiben</li> <li>● 7: Befehl ausführen</li> </ul>
i_dwPointerToData	DWORD	Zeiger auf den Datenspeicher, gefüllt mit der tatsächlichen Länge der Datenbytes.
i_wMaxDataLen	WORD	Tatsächliche Länge der zu schreibenden Datenbytes.

In der folgenden Tabelle wird die Ausgangsvariable beschrieben:

Ausgang	Typ	Kommentar
FC_SercosWriteServiceData	DINT	Siehe Tabelle mit den Beschreibungen der Ausgangswerte unten.

In der folgenden Tabelle wird der Ausgangswert beschrieben:

Wert	Beschreibung
0	Die Funktion wurde erfolgreich ausgeführt.
-1	Die logische Adresse ist ungültig.
-431	Während der Dienstanfrage wurde ein Fehler entdeckt (z. B. Timeout).
-445	Dienst-Timeout.
-461	In der aktuellen Phase wird das Schreiben von Parametern über den Servicekanal nicht unterstützt.
-462	Das adressierte Gerät unterstützt die Funktion <code>ServiceDataWrite</code> nicht.
-464	Ungültiger Service-Transfer.

## FC\_SercosWriteServiceDataByTopAddr-Funktion

### Beschreibung der Funktion

Die Funktion FC\_SercosWriteServiceDataByTopAddr schreibt Dienstdaten über den Sercos-Servicekanal in das adressierte Gerät.

### WARNUNG

#### UNBEABSICHTIGTER GERÄTEBETRIEB

Verwenden Sie diese Funktion nur nach Absprache mit der Anwendungsabteilung von Schneider Electric.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

### Grafische Darstellung



### Darstellung in AWL (IL) und ST

Sie finden eine allgemeine Darstellung in AWL (IL) oder ST im Kapitel *Darstellung von Funktionen und Funktionsbausteinen*.

## Beschreibung der E/A-Variablen

In der folgenden Tabelle werden die Eingangsvariablen beschrieben:

Eingangs-	Typ	Kommentar
i_wTopoAddr	WORD	Topologische Adresse des Sercos-Geräts.
i_dwIDN	DWORD	Datendienst IDN.
i_wType	WORD	Datendiensttyp. <ul style="list-style-type: none"> <li>● 0: Name schreiben</li> <li>● 1: Attribut schreiben</li> <li>● 2: Einheit schreiben</li> <li>● 3: Mindestwert schreiben</li> <li>● 4: Höchstwert schreiben</li> <li>● 5: Benutzerdaten schreiben</li> <li>● 6: Benutzerdaten mit variabler Länge schreiben</li> <li>● 7: Befehl ausführen</li> </ul>
i_dwPointerToData	DWORD	Zeiger auf den Datenspeicher, gefüllt mit der tatsächlichen Länge der Datenbytes.
i_wMaxDataLen	WORD	Tatsächliche Länge der zu schreibenden Datenbytes.

In der folgenden Tabelle wird die Ausgangsvariable beschrieben:

Ausgang	Typ	Kommentar
FC_SercosWriteServiceDataByTopAddr	DINT	Siehe Tabelle mit den Beschreibungen der Ausgangswerte unten.

In der folgenden Tabelle wird der Ausgangswert beschrieben:

Wert	Beschreibung
0	Die Funktion wurde erfolgreich ausgeführt.
-1	Die topologische Adresse ist ungültig.
-431	Während der Dienstanfrage wurde ein Fehler entdeckt (z. B. Timeout).
-445	Dienst-Timeout.
-461	In der aktuellen Phase wird das Schreiben von Parametern über den Servicekanal nicht unterstützt.
-462	Das adressierte Gerät unterstützt die Funktion <code>ServiceDataWrite</code> nicht.
-464	Ungültiger Service-Transfer.

## Abschnitt C.3

### Asynchrone Sercos-Funktionsbausteine

#### Überblick

In diesem Abschnitt werden die Sercos-Funktionsbausteine beschrieben.

Diese Funktionsbausteine werden für die azyklische Kommunikation auf dem Sercos-Servicekanal verwendet.

Sie werden für das asynchrone Lesen/Schreiben von Geräteparametern und zum Senden von Befehlen über die Sercos-Schnittstelle verwendet.

Alle IDN des LXM32S-Antriebs sind über diese Funktionsbausteine zugänglich.

#### Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
FB_SercosReadServiceDataAsync : Asynchrones Lesen von Daten über die Sercos-Schnittstelle	303
FB_SercosWriteServiceDataAsync : Asynchrones Schreiben von Daten über die Sercos-Schnittstelle	305
FB_SercosProcedureCommandAsync: Asynchrones Senden von Befehlen über die Sercos-Schnittstelle	307

## FB\_SercosReadServiceDataAsync : Asynchrones Lesen von Daten über die Sercos-Schnittstelle

### Beschreibung des Funktionsbausteins

Der Funktionsbaustein `FB_SercosReadServiceDataAsync` liest Daten asynchron über die Sercos-Schnittstelle.

### Grafische Darstellung



### Darstellung in AWL (IL) und ST

Sie finden eine allgemeine Darstellung in AWL (IL) oder ST im Kapitel *Darstellung von Funktionen und Funktionsbausteinen*.

### Beschreibung der E/A-Variablen

In der folgenden Tabelle werden die Eingangsvariablen beschrieben:

Eingang	Typ	Kommentar
<code>i_xExecute</code>	BOOL	TRUE gibt an, dass der Funktionsbaustein ausgeführt wird.
<code>i_ifSlave</code>	SystemConfiguration.IF_IdentificationMandatory	Schnittstelle, die das Gerät beschreibt. Sie enthält den Namen des Geräts und seine logische Nummer.
<code>i_uwIDN</code>	WORD	IDN des Geräts. Beispiel für IDN S-0-1027.0.1: <code>i_uwIDN = 1027</code>
<code>i_etIDNType</code>	ET_Sercos3IDNType (siehe Seite 287)	IDN-Typ (Standard oder Proprietär). Beispiel für IDN S-0-1027.0.1: <code>i_etIDNType = ET_Sercos3IDNType.ParameterType_S;</code>
<code>i_uwIDN_SI</code>	WORD	IDN-Strukturinstanz. Beispiel für IDN S-0-1027.0.1: <code>i_uwIDN_SI = 0</code>

Eingang	Typ	Kommentar
i_uwIDN_SE	WORD	IDN-Strukturelement. Beispiel für IDN S-0-1027.0.1: i_uwIDN_SE = 1
i_etCmdType	ET_Sercos3CmdType (siehe Seite 286)	Typ des auszuführenden Befehls. Der Befehl unterstützt die Befehlstypen 0 bis 6.
i_pBuffer	POINTER TO BYTE	Zeiger auf einen Puffer der Länge i_szSize.
i_wSize	WORD	Länge des Puffers.

In der folgenden Tabelle werden die Ausgangsvariablen beschrieben:

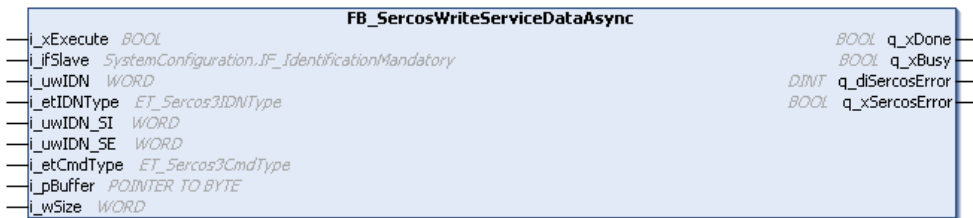
Ausgang	Typ	Kommentar
q_xDone	BOOL	TRUE gibt an, dass die Ausführung des Funktionsbausteins abgeschlossen ist und keine Fehler erkannt wurden.
q_xBusy	BOOL	TRUE gibt an, dass der Funktionsbaustein ausgeführt wird.
q_diSercosError (siehe Seite 224)	DINT	Sercos (siehe Seite 224)-Fehlerwert, der vom synchronen Lesebefehl ausgegeben wird.
q_xSercosError	BOOL	TRUE gibt an, dass ein Sercos-Fehler erkannt wurde (negativer Wert).
q_uiReadDataLen	WORD	Anzahl der gelesenen Daten.

## FB\_SercosWriteServiceDataAsync : Asynchrones Schreiben von Daten über die Sercos-Schnittstelle

### Beschreibung des Funktionsbausteins

Der Funktionsbaustein `FB_SercosWriteServiceDataAsync` schreibt Daten asynchron über die Sercos-Schnittstelle.

### Grafische Darstellung



### Darstellung in AWL (IL) und ST

Sie finden eine allgemeine Darstellung in AWL (IL) oder ST im Kapitel *Darstellung von Funktionen und Funktionsbausteinen*.

### Beschreibung der E/A-Variablen

In der folgenden Tabelle werden die Eingangsvariablen beschrieben:

Eingang	Typ	Kommentar
<code>i_xExecute</code>	BOOL	TRUE gibt an, dass der Funktionsbaustein ausgeführt wird.
<code>i_ifSlave</code>	SystemConfiguration.IF_IdentificationMandatory	Schnittstelle, die das Gerät beschreibt. Sie enthält den Namen des Geräts und seine logische Nummer.
<code>i_uwIDN</code>	WORD	IDN des Geräts. Beispiel für IDN S-0-1027.0.1: <code>i_uwIDN = 1027</code>
<code>i_etIDNType</code>	ET_Sercos3IDNType (siehe Seite 287)	IDN-Typ (Standard oder Proprietär). Beispiel für IDN S-0-1027.0.1: <code>i_etIDNType = ET_Sercos3IDNType.ParameterType_S;</code>
<code>i_uwIDN_SI</code>	WORD	IDN-Strukturinstanz. Beispiel für IDN S-0-1027.0.1: <code>i_uwIDN_SI = 0</code>

Eingang	Typ	Kommentar
i_uwIDN_SE	WORD	IDN-Strukturelement. Beispiel für IDN S-0-1027.0.1: i_uwIDN_SE = 1
i_etCmdType	ET_Sercos3CmdType (siehe Seite 286)	Typ des auszuführenden Befehls. Der Befehl unterstützt die Befehlstypen 0 bis 6.
i_pBuffer	POINTER TO BYTE	Zeiger auf einen Puffer der Länge i_szSize.
i_wSize	WORD	Länge des Puffers.

In der folgenden Tabelle werden die Ausgangsvariablen beschrieben:

Ausgang	Typ	Kommentar
q_xDone	BOOL	TRUE gibt an, dass die Ausführung des Funktionsbausteins abgeschlossen ist und keine Fehler erkannt wurden.
q_xBusy	BOOL	TRUE gibt an, dass der Funktionsbaustein ausgeführt wird.
q_diSercosError (siehe Seite 224)	DINT	Sercos (siehe Seite 224)-Fehlerwert, der vom synchronen Lesebefehl ausgegeben wird.
q_xSercosError	BOOL	TRUE gibt an, dass ein Sercos-Fehler erkannt wurde (negativer Wert).

## FB\_SercosProcedureCommandAsync: Asynchrones Senden von Befehlen über die Sercos-Schnittstelle

### Beschreibung des Funktionsbausteins

Der Funktionsbaustein `FB_SercosProcedureCommandAsync` sendet Befehle asynchron über die Sercos-Schnittstelle.

### Grafische Darstellung



### Darstellung in AWL (IL) und ST

Sie finden eine allgemeine Darstellung in AWL (IL) oder ST im Kapitel *Darstellung von Funktionen und Funktionsbausteinen*.

### Beschreibung der E/A-Variablen

In der folgenden Tabelle werden die Eingangsvariablen beschrieben:

Eingang	Typ	Kommentar
<code>i_xExecute</code>	BOOL	TRUE gibt an, dass der Funktionsbaustein ausgeführt wird.
<code>i_xAbort</code>	BOOL	TRUE gibt an, dass die Ausführung des Befehls abgebrochen wird. FALSE gibt an, dass der Befehl wie geplant ausgeführt wird.
<code>i_ifSlave</code>	SystemConfiguration.IF_ IdentificationMandatory	Schnittstelle, die das Gerät beschreibt. Sie enthält den Namen des Geräts und seine logische Nummer.
<code>i_uwIDN</code>	WORD	IDN des Geräts. Beispiel für IDN S-0-1027.0.1: <code>i_uwIDN = 1027</code>
<code>i_etIDNType</code>	ET_Sercos3IDNType (siehe Seite 287)	IDN-Typ (Standard oder Proprietär). Beispiel für IDN S-0-1027.0.1: <code>i_etIDNType = ET_Sercos3IDNType.ParameterType_S;</code>

Eingang	Typ	Kommentar
i_uwIDN_SI	WORD	IDN-Strukturinstanz. Beispiel für IDN S-0-1027.0.1: i_uwIDN_SI = 0
i_uwIDN_SE	WORD	IDN-Strukturelement. Beispiel für IDN S-0-1027.0.1: i_uwIDN_SE = 1

In der folgenden Tabelle werden die Ausgangsvariablen beschrieben:

Ausgang	Typ	Kommentar
q_xDone	BOOL	TRUE gibt an, dass die Ausführung des Funktionsbausteins abgeschlossen ist und keine Fehler erkannt wurden.
q_xBusy	BOOL	TRUE gibt an, dass der Funktionsbaustein ausgeführt wird.
q_diSercosError (siehe Seite 224)	DINT	Sercos (siehe Seite 224)-Fehlerwert, der vom synchronen Lesebefehl ausgegeben wird.
q_xSercosError	BOOL	TRUE gibt an, dass ein Sercos-Fehler erkannt wurde (negativer Wert).
q_uiProcCmdError	UINT	Prozedurbefehl-Fehlerwert, der vom synchronen Lesebefehl ausgegeben wird.
q_xProcCmdError	BOOL	TRUE gibt an, dass ein Prozedurbefehlsfehler erkannt wurde.

---

# Anhang D

## Funktionen zum Abrufen/Einrichten der Konfiguration der seriellen Leitung in einem Anwenderprogramm

---

### Überblick

In diesem Abschnitt werden die Funktionen zum Abrufen/Einrichten der Konfiguration der seriellen Leitung in einem Anwenderprogramm beschrieben.

Um diese Funktionen nutzen zu können, müssen Sie die **M2xx Communication**-Bibliothek hinzufügen.

Weitere Informationen über das Hinzufügen einer Bibliothek finden Sie im SoMachine Programmierhandbuch.

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

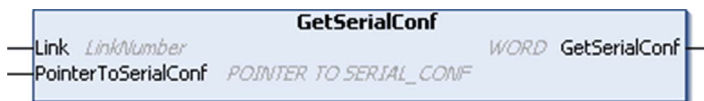
Thema	Seite
GetSerialConf: Abrufen der Konfiguration der seriellen Leitung	310
SetSerialConf: Ändern der Konfiguration der seriellen Leitung	311
SERIAL_CONF: Struktur des Datentyps für die serielle Leitungskonfiguration	313

## GetSerialConf: Abrufen der Konfiguration der seriellen Leitung

### Beschreibung der Funktion

GetSerialConf gibt die Konfigurationsparameter für den Kommunikationsport einer bestimmten seriellen Leitung zurück.

### Grafische Darstellung



### Parameterbeschreibung

Eingänge	Typ	Kommentar
Link	LinkNumber <i>(siehe SoMachine, Modbus- und ASCII-Lese-/Schreibfunktionen, PLCCommunication-Bibliothekshandbuch)</i>	Link ist die Nummer des Kommunikationsports.
PointerToSerialConf	POINTER TO SERIAL_CONF <i>(siehe Seite 313)</i>	PointerToSerialConf ist die Adresse der Konfigurationsstruktur (Variable vom Typ SERIAL_CONF), in der die Konfigurationsparameter gespeichert werden. Die Standardfunktion ADR muss zum Definieren des zugehörigen Zeigers verwendet werden (siehe nachstehendes Beispiel).

Ausgänge	Typ	Kommentar
GetSerialConf	WORD	Diese Funktion gibt Folgendes zurück: <ul style="list-style-type: none"> <li>● 0: Die Konfigurationsparameter werden zurückgegeben.</li> <li>● 255: Die Konfigurationsparameter werden nicht zurückgegeben, da: <ul style="list-style-type: none"> <li>○ die Funktion nicht erfolgreich war.</li> <li>○ die Funktion gerade ausgeführt wird.</li> </ul> </li> </ul>

### Beispiel

Siehe Beispiel für SetSerialConf *(siehe Seite 312)*.

## SetSerialConf: Ändern der Konfiguration der seriellen Leitung

### Funktionsbeschreibung

SetSerialConf ermöglicht die Änderung der Konfiguration der seriellen Leitung.

### Grafische Darstellung



**HINWEIS:** Das Ändern der Konfiguration der Ports für serielle Leitungen während der Programmausführung kann zu einer Unterbrechung der Kommunikation zwischen zwei miteinander verbundenen Geräten führen.

## ⚠️ WARNUNG

### STEUERUNGSAusFALL AUFGruND EINER UNERWARTETEN KONFIGURATIONSÄNDERUNG

Validieren und testen Sie alle Parameter der Funktion SetSerialConf vor der Ausführung des Programms.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

### Parameterbeschreibung

Eingang	Typ	Kommentar
Link	LinkNumber (siehe SoMachine, Modbus- und ASCII-Lese-/Schreibfunktionen, PLCCommunication-Bibliothekshandbuch)	LinkNumber ist die Nummer des Kommunikationsports.
PointerToSerialConf	POINTER TO SERIAL_CONF (siehe Seite 313)	PointerToSerialConf ist die Adresse der Konfigurationsstruktur (Variable vom Typ SERIAL_CONF), in der die neuen Konfigurationsparameter gespeichert werden. Die Standardfunktion <code>ADR</code> muss zum Definieren des zugehörigen Zeigers verwendet werden. (siehe nachstehendes Beispiel). Wenn 0, stellen Sie die Standardkonfiguration der Anwendung auf die serielle Leitung ein.

Ausgang	Typ	Kommentar
SetSerialConf	WORD	<p>Diese Funktion gibt Folgendes zurück:</p> <ul style="list-style-type: none"> <li>● 0: Die neue Konfiguration ist eingerichtet.</li> <li>● 255: Die neue Konfiguration wird abgelehnt, da: <ul style="list-style-type: none"> <li>○ die Funktion gerade ausgeführt wird.</li> <li>○ die Eingangsparameter ungültig sind.</li> </ul> </li> </ul>

### Beispiel

```

VAR
  MySerialConf: SERIAL_CONF
  result: WORD;
END_VAR

(*Aktuelle Konfiguration der seriellen Leitung 1 abrufen*)
GetSerialConf(1, ADR(MySerialConf));

(*Zu Modbus RTU-Slave-Adresse 9 ändern*)
MySerialConf.Protocol := 0; (*Modbus RTU/SoMachine-Protokoll (in diesem
Fall wählt CodesysCompliant das Protokoll aus)*)
MySerialConf.CodesysCompliant := 0; (*Modbus RTU*)
MySerialConf.address := 9; (*Modbus-Adresse auf 9 setzen*)

(*Die serielle Leitung 1 neu konfigurieren*)
result := SetSerialConf(1, ADR(MySerialConf));

```

## SERIAL\_CONF: Struktur des Datentyps für die serielle Leitungskonfiguration

### Strukturbeschreibung

Die Struktur von SERIAL\_CONF enthält Konfigurationsinformationen über den seriellen Leitungsanschluss. Sie enthält die folgenden Variablen:

Variable	Typ	Beschreibung
Bauds	DWORD	Baudrate
InterframeDelay	WORD	Mindestzeit (in ms) zwischen 2 Frames in Modbus (RTU, ASCII)
FrameReceivedTimeout	WORD	Im ASCII-Protokoll kann das System anhand von <code>FrameReceivedTimeout</code> das Ende eines Frame beim Empfang erkennen, wenn eine Stille einer festgelegten Anzahl von ms aufgetreten ist. Bei einem Wert von 0 wird dieser Parameter nicht verwendet.
FrameLengthReceived	WORD	Im ASCII-Protokoll kann das System anhand von <code>FrameLengthReceived</code> das Ende eines Frame beim Empfang ermitteln, wenn die Steuerung die festgelegte Anzahl von Zeichen empfangen hat. Bei einem Wert von 0 wird dieser Parameter nicht verwendet.
Protocol	BYTE	0: Modbus RTU oder SoMachine (siehe <code>CodesysCompliant</code> ) 1: Modbus ASCII 2: ASCII
Address	BYTE	Modbus-Adresse 0 bis 255 (0 für Master)
Parity	BYTE	0: Keine 1: Ungerade 2: Gerade
Rs485	BYTE	0: RS232 1: RS485
ModPol (Polarisierungswiderstand)	BYTE	0: Nein 1: Ja
DataFormat	BYTE	7 Bits oder 8 Bits
StopBit	BYTE	1: 1 Stoppbit 2: 2 Stoppbits
CharFrameStart	BYTE	Im ASCII-Protokoll bedeutet 0, dass im Frame kein Startzeichen vorhanden ist. Andernfalls dient das entsprechende ASCII-Zeichen dazu, den Beginn eines Frames im Empfangsmodus zu erkennen. Im Sendemodus wird dieses Zeichen zu Beginn des Benutzer-Frames hinzugefügt.

Variable	Typ	Beschreibung
CharFrameEnd1	BYTE	Im ASCII-Protokoll bedeutet 0, dass im Frame kein zweites Endzeichen vorhanden ist. Andernfalls dient das entsprechende ASCII-Zeichen dazu, das Ende eines Frames im Empfangsmodus zu erkennen. Im Sendemodus wird dieses Zeichen am Ende des Benutzer-Frames hinzugefügt.
CharFrameEnd2	BYTE	Im ASCII-Protokoll bedeutet 0, dass im Frame kein zweites Endzeichen vorhanden ist. Andernfalls dient das entsprechende ASCII-Zeichen (zusammen mit CharFrameEnd1) dazu, das Ende eines Frames im Empfangsmodus zu erkennen. Im Sendemodus wird das Zeichen am Ende des Benutzer-Frames hinzugefügt.
CodesysCompliant	BYTE	0: Modbus RTU
		1: SoMachine (wenn Protocol = 0)
CodesysNetType	BYTE	Nicht verwendet

---

# Anhang E

## SPS-Leistung

---

### Verarbeitungsleistung

#### Einführung

Dieses Kapitel enthält Informationen zur Verarbeitungsleistung des LMC078.

#### Logik-Verarbeitung

Diese Tabelle zeigt die Logik-Verarbeitungsleistung für verschiedene logische Anweisungen:

Anweisungen vom Typ AWL (IL)	Dauer für 1.000 Anweisungen
Addition/Subtraktion/Multiplikation von INT	1 $\mu$ s
Addition/Subtraktion/Multiplikation von DINT	1 $\mu$ s
Addition/Subtraktion/Multiplikation von REAL	3 $\mu$ s
Division von REAL	48 $\mu$ s
Operation mit BOOLEAN, z. B. Status:= Status und Wert	2 $\mu$ s
LD INT + ST INT	1 $\mu$ s
LD DINT + ST DINT	1 $\mu$ s
LD REAL + ST REAL	9 $\mu$ s

#### Kommunikations- und Systemverarbeitungszeit

Die Kommunikationsverarbeitungszeit fällt je nach Anzahl der gesendeten/empfangenen Anforderungen unterschiedlich aus.

#### Antwortzeit bei Ereignissen

Die in der nachstehenden Tabelle gezeigte Antwortzeit entspricht der Zeit zwischen der steigenden Flanke eines Signals an einem Eingang, durch die eine externe Ereignistask ausgelöst wird, und der Flanke des durch diese Task gesetzten Ausganges. Die Ereignistask verarbeitet zudem 100 AWL-Anweisungen, bevor der Ausgang gesetzt wird:

Minimum	Typisch	Maximum
120 $\mu$ s	126 $\mu$ s	140 $\mu$ s





## !

### **%IW**

Gemäß dem IEC-Standard entspricht %IW einem Eingangswortregister (z. B. einem Sprachobjekt des Typs analoger IN).

### **%QW**

Gemäß dem IEC-Standard entspricht %QW einem Ausgangswortregister (z. B. einem Sprachobjekt des Typs analoger OUT).

## A

### **Anwendung**

Programm mit Konfigurationsdaten, Symbolen und Dokumentation.

### **ARP**

(*Address Resolution Protocol: Adressauflösungsprotokoll*) IP-Protokoll der Netzwerkschicht für Ethernet, das eine IP-Adresse einer MAC-Adresse (Hardwareadresse) zuordnet.

### **AT**

(*Acknowledge Telegram, Telegramm bestätigen*) Auf dem Sercos-Bus werden Daten über AT-Telegramme (Feedback-Werte) von den Slaves an den Master gesendet.

## B

### **BOOL**

(*Boolesch*) Basis-Datentyp in der Datenverarbeitung. Eine Variable des Typs `BOOL` besitzt einen der folgenden Werte: 0 (`FALSE`) oder 1 (`TRUE`). Ein aus einem Wort extrahiertes Bit ist vom Typ `BOOL`. Beispiel: `%MW10.4` ist das fünfte Bit des Speicherworts 10.

### **Boot-Anwendung**

(*Boot-Anwendung*) Binärdatei mit der Anwendung. In der Regel wird die Datei in der SPS gespeichert, sodass die SPS mit der vom Benutzer generierten Anwendung starten kann.

## BOOTP

(*Bootstrap-Protokoll*) UDP-Netzwerkprotokoll, das von einem Netzwerk-Client verwendet werden kann, um automatisch eine IP-Adresse (und möglicherweise weitere Daten) von einem Server zu erhalten. Der Client identifiziert sich beim Server anhand der MAC-Adresse des Clients. Der Server, der eine vorkonfigurierte Tabelle der MAC-Adressen der Client-Geräte und der zugeordneten IP-Adressen speichert, sendet dem Client seine vorkonfigurierte IP-Adresse. BOOTP wurde ursprünglich zum Remote-Booten von Hosts über ein Netzwerk verwendet, die über keinen eigenen Plattenspeicher verfügen. Der BOOTP-Prozess weist eine IP-Adresse mit unbegrenzter Laufzeit zu. Der BOOTP-Dienst nutzt die UDP-Ports 67 und 68.

## C

### CANopen

Offenes Kommunikationsprotokoll nach Industriestandard und Geräteprofil-Spezifikation (EN 50325-4).

### CFC

(*Continuous Function Chart*) Grafische Programmiersprache (Erweiterung des Standards IEC 61131-3) auf der Grundlage der FBD-Sprache (Funktionsbausteindiagramm), die wie ein Flussdiagramm aufgebaut ist. Grafische Elemente werden allerdings, sofern möglich, ohne die Verwendung von Netzwerken frei positioniert, sodass Rückkopplungsschleifen möglich sind. Bei jedem Baustein befinden sich die Eingänge links und die Ausgänge rechts. Sie können die Bausteinausgänge mit den Eingängen anderer Bausteine verbinden, um komplexe Ausdrücke zu erstellen.

### CIP

(*Common Industrial Protocol*) Wenn ein CIP-Protokoll in einer Netzwerkanwendungsschicht implementiert wird, kann es übergangslos mit anderen CIP-basierten Netzwerken ungeachtet des jeweiligen Protokolls kommunizieren. Die Implementierung von CIP in der Anwendungsschicht eines Ethernet TCP/IP-Netzwerks schafft beispielsweise eine EtherNet/IP-Umgebung. In ähnlicher Weise erzeugt CIP in der Anwendungsschicht eines CAN-Netzwerks eine DeviceNet-Umgebung. In diesem Fall können die Geräte im EtherNet/IP-Netzwerk mit den Geräten im DeviceNet-Netzwerk über CIP-Bridges oder -Router kommunizieren.

### Continuous Function Chart (Programmiersprache)

Grafische Programmiersprache (Erweiterung des Standards IEC61131-3) auf der Grundlage der FBD-Sprache (Funktionsbausteindiagramm), die wie ein Flussdiagramm aufgebaut ist. Grafische Elemente werden allerdings, sofern möglich, ohne die Verwendung von Netzwerken frei positioniert, sodass Rückkopplungsschleifen möglich sind. Bei jedem Baustein befinden sich die Eingänge links und die Ausgänge rechts. Sie können die Bausteinausgänge mit den Eingängen anderer Bausteine verbinden, um komplexe Ausdrücke zu erstellen.

**CRC**

(*Cyclical Redundancy Check: Zyklische Redundanzprüfung*) Methode zur Bestimmung der Gültigkeit einer Kommunikationsübertragung. Die Übertragung enthält ein Bitfeld, das einer Prüfsumme entspricht. Mithilfe der Nachricht wird die Prüfsumme vom Sender in Übereinstimmung mit dem Inhalt der Nachricht berechnet. Die Empfängerknoten berechnen das Feld dann auf dieselbe Weise neu. Jede Abweichung zwischen den Werten der zwei CRC-Felder verweist darauf, dass die übertragene und die empfangene Nachricht unterschiedlich sind.

**D****DHCP**

(*Dynamic Host Configuration Protocol*) Hochentwickelte Erweiterung von BOOTP. Das DHCP-Protokoll ist ausgereifter, doch sowohl DHCP als auch BOOTP sind gängig. (DHCP kann BOOTP-Client-Requests verarbeiten.)

**DINT**

(*Double Integer Type: Doppelte Ganzzahl*) Im 32-Bit-Format codierter Typ.

**DWORD**

(*Double Word: Doppelwort*) Im 32-Bit-Format codierter Typ.

**E****E/A**

*Eingang/Ausgang*

**EDS**

(*Electronic Data Sheet: Elektronisches Datenblatt*) Datei für die Beschreibung eines Feldbusgeräts, das beispielsweise die Eigenschaften des Geräts wie Parameter und Einstellungen enthält.

**Ethernet**

Technologie der physikalischen und der Daten Verbindungsschicht für LANs, auch als IEEE 802.3 bekannt.

**EtherNet/IP**

(*Ethernet Industrial Protocol*) Offenes Kommunikationsprotokoll für Fertigungsautomatisierungslösungen in industriellen Systemen. EtherNet/IP gehört zu einer Familie von Netzwerken, die CIP (Common Industrial Protocol) in den oberen Schichten implementieren. Die unterstützende Organisation (ODVA) gibt EtherNet/IP für globale Anpassungsfähigkeit und Medienunabhängigkeit vor.

## F

### FB

(*Function Block: Funktionsbaustein*) Nützlicher Programmiermechanismus, der eine Gruppe von Programmieranweisungen zur Durchführung eines spezifischen und normierten Vorgangs konsolidiert, z. B. Drehzahlregelung, Intervallkontrolle oder Zählen. Ein Funktionsbaustein kann Konfigurationsdaten, eine Gruppe interner oder externer Betriebsparameter und in der Regel 1 oder mehrere Dateneingänge und -ausgänge umfassen.

### FE

(*Functional Earth: Funktionserde*) Gemeinsame Erdungsverbindung zur Verbesserung oder Ermöglichung eines normalen Betriebs elektrisch sensibler Geräte (in Nordamerika auch als Funktionsmasse bezeichnet).

Im Gegensatz zur Schutzerde (Schutzmasse) dient eine FE-Verbindung einem anderen Zweck als dem Schutz vor elektrischen Schlägen und kann im Normalfall stromführend sein. Beispiele für Geräte, die FE-Verbindungen verwenden: Stoßspannungsbegrenzer und elektromagnetische Störungsfilter, bestimmte Antennen und Messgeräte.

### Firmware

Umfasst das BIOS, Datenparameter und Programmieranweisungen, aus denen das Betriebssystem einer Steuerung besteht. Die Firmware wird in einem nicht flüchtigen Speicher in der Steuerung abgelegt.

### FTP

(*File Transfer Protocol: Dateiübertragungsprotokoll*) Standard-Netzwerkprotokoll auf der Grundlage einer Client/Server-Architektur für den Austausch und die Bearbeitung von Dateien über TCP/IP-basierte Netzwerke ungeachtet deren Größe.

### Funktionsbaustein (FB)

Programmiereinheit, die über 1 oder mehrere Eingänge verfügt und 1 oder mehrere Ausgänge zurückgibt. FBs werden über eine Instanz (Kopie des Funktionsbausteins mit dediziertem Namen und Variablen) aufgerufen, wobei jede Instanz zwischen zwei Aufrufen einen persistenten Status aufweist (Ausgänge und interne Variablen).

Beispiele: Timer (Zeitgeber), Zähler.

### Funktionsfähigkeitsbit

Variable, die den Kommunikationszustand der Kanäle angibt.

## I

### ICMP

(*Internet Control Message Protocol*) Signalisiert Fehler und stellt Informationen zur Datagramm-Verarbeitung bereit.

**IEC**

(*International Electrotechnical Commission*) Gemeinnütziges, internationales Normungsgremium, das sich die Ausarbeitung und Veröffentlichung internationaler Normen für die Elektro- und Elektronikindustrie sowie zugehörige Technologien zur Aufgabe gemacht hat.

**IEC 61131-3**

Teil 3 eines 3-teiligen IEC-Standards für industrielle Automatisierungsanlagen. IEC 61131-3 befasst sich mit den Programmiersprachen für Steuerungen und definiert 2 grafische und 2 textbasierte Programmiersprachenstandards. Grafische Programmiersprachen: Kontaktplan (KOP oder LD: Ladder) und Funktionsbausteindiagramm (FBD oder Function Block Diagram). Textbasierte Programmiersprachen: Strukturierter Text (ST) und Anweisungsliste (AWL oder IL: Instruction List).

**IL**

(*Instruction List: Anweisungsliste (AWL)*) Ein in Anweisungsliste geschriebenes Programm besteht aus einer Abfolge textbasierter Anweisungen, die von der Steuerung der Reihe nach ausgeführt werden. Jede Anweisung besteht aus einer Zeilennummer, einem Anweisungscode und einem Operanden (siehe IEC 61131-3).

**Input Assembly**

Baugruppen (Assemblies) sind Datenblöcke, die zwischen Netzwerkgeräten und der Steuerung ausgetauscht werden. Ein Input Assembly enthält in der Regel vom Master oder Urheber gelesene Statusinformationen eines Slave- oder Zielgeräts.

**INT**

(*Integer: Ganzzahl*) Über 16 Bits codierte Ganzzahl.

**IP**

(*Internet Protocol: Internetprotokoll*) Teil der TCP/IP-Protokollfamilie, der die Internetadresse von Geräten verfolgt, das Routing für abgehende Nachrichten übernimmt und eingehende Nachrichten erkennt.

**K****Knoten**

Adressierbares Gerät in einem Kommunikationsnetzwerk (Netzwerkteilnehmer).

**Konfiguration**

Die Anordnung und Vernetzung von Hardwarekomponenten innerhalb eines Systems und die Hardware- und Softwareparameter, die die Betriebsmerkmale des Systems bestimmen.

## L

### LD

(*Ladder Diagram: Kontaktplan (KOP)*) Grafische Darstellung der Anweisungen eines Steuerungsprogramms mit Symbolen für Kontakte, Spulen und Bausteine in einer Abfolge von Programmbausteinen, die von der Steuerung der Reihe nach ausgeführt werden (siehe IEC 61131-3).

### LED

(*Light Emitting Diode*) Anzeige, die bei niedriger Stromlast aufleuchtet.

### LINT

(*Long Integer: Lange Ganzzahl*) In einem 64-Bit-Format codierte Ganzzahl (4 x INT oder 2 x DINT).

### LRC

(*Longitudinal Redundancy Checking*) Methode zur Fehlererkennung für die Bestimmung der Richtigkeit übertragener und gespeicherter Daten.

### LWORD

(*Long Word: Langes Wort*) In einem 64-Bit-Format codierter Datentyp.

## M

### MAC Adresse

(*Media Access Control*) Eindeutige 48-Bit-Zahl, die einer bestimmten Hardwarekomponente zugeordnet ist. Die MAC-Adresse wird bei der Fertigung in jede Netzwerkkarte bzw. jedes Gerät programmiert.

### MAST

Prozessortask, die über die zugehörige Programmiersoftware ausgeführt wird. Die MAST-Task besteht aus zwei Sections:

- **IN:** Vor der Ausführung der MAST-Task werden die Eingänge in die IN-Section kopiert.
- **OUT:** Nach der Ausführung der MAST-Task werden die Ausgänge in die OUT-Section kopiert.

### MDT

(*Master Data Telegram, Masterdatentelegramm*) Auf dem Sercos-Bus wird ein MDT-Telegramm vom Master gesendet, und zwar je einmal pro Übertragungszyklus, um Daten (Befehlswerte) an die Servoantriebe (Slaves) zu senden.

### ms

*Millisekunden*

### MST

(*Master Synchronization Telegram, Mastersynchronisierungstelegramm*) Auf dem Sercos-Bus wird zu Beginn jedes Übertragungszyklus vom Master ein MST-Telegramm gesendet, um die Zeit des Zyklus abzugleichen.

## N

### Netzwerk

Ein Netzwerk umfasst miteinander verbundene Geräte, die einen gemeinsamen Datenpfad und dasselbe Protokoll zur Kommunikation verwenden.

### Nullspur

Die Spur eines Inkrementalgeber-Codierers, der als Referenzpunkt dient und eine Reinitialisierung bei jeder Drehung ermöglicht. Auch als Top 0 oder Top Z bekannt.

## O

### OS

(*Operating System: Betriebssystem*) Gruppe von Softwareprogrammen, die die Hardwareressourcen eines Computers verwalten und für die Computerprogramme gemeinsam nutzbare Dienste bereitstellen.

### Output Assembly

Baugruppen (Assemblies) sind Datenblöcke, die zwischen Netzwerkgeräten und der Steuerung ausgetauscht werden. Ein Output Assembly enthält Befehle, die vom Master oder Urheber an die Slave- oder Zielgeräte gesendet werden.

## P

### PDO

(*Process Data Object: Prozessdatenobjekt*) Wird in CAN-basierenden Netzwerken als nicht bestätigte Broadcast-Meldung übertragen oder von einem Erzeugergerät (Producer) an ein Verbrauchergerät (Consumer) gesendet. Das Sende-PDO vom Producer-Gerät hat eine spezifische Kennung, die dem Empfangs-PDO der Consumer-Geräte entspricht.

### PE

(*Protective Earth: Schutzerde*) Gemeinsame Erdungsverbindung zur Vermeidung elektrischer Schläge durch den Anschluss aller frei liegenden leitenden Flächen an das Massepotential. Um einen Spannungsabfall zu vermeiden, ist in diesem Leiter kein Stromfluss zugelassen (in Nordamerika auch als *Schutzmasse* oder als Gerätemasseleiter im US-amerikanischen Stromcode bezeichnet).

### Persistente Daten

Daten mit beständigen Werten, die bei der nächsten Änderung der Anwendung oder einem Kaltstart verwendet werden. Sie werden nur bei einem Neustart der Steuerung oder einem Reset auf den Ursprung neu initialisiert. Insbesondere behalten sie ihre Werte nach einem Download bei.

## POU

(*Program Organization Unit: Programmierorganisationseinheit*) Variablendeklaration im Quellcode und der entsprechende Anweisungssatz. POU ermöglichen die modulare Wiederverwendung von Softwareprogrammen, Funktionen und Funktionsbausteinen. Sobald POU deklariert sind, stehen sie sich gegenseitig zur Verfügung.

## Profibus DP

(*Profibus Decentralized Peripheral*) Offenes Bussystem, das ein auf zweidrahtigen geschirmten Kabeln basiertes elektrisches Netzwerk oder ein auf Glasfaserkabeln basiertes optisches Netzwerk verwendet. Die DP-Übertragung ermöglicht den zyklischen Hochgeschwindigkeitsaustausch von Daten zwischen der CPU der Steuerung und den verteilten E/A-Geräten.

## Programm

Komponente einer Anwendung, die aus kompiliertem Quellcode besteht und im Speicher einer programmierbaren Steuerung installiert werden kann.

## Protokoll

Konvention oder Standarddefinition, die die Verbindung, Kommunikation und Datenübertragung zwischen 2 Rechensystemen und Geräten steuert und ermöglicht.

# R

## REAL

Datentyp, der als in einem 32-Bit-Format codierte Gleitkommazahl definiert wird.

## Retentive Daten

Der Wert von Retain-Daten (beibehaltenen Daten) wird beim nächsten Einschalten oder Warmstart verwendet. Der Wert wird beim Herunterfahren der Steuerung aufgrund einer Unterbrechung der Spannungsversorgung oder im Anschluss an einen regulären Ausschaltbefehl beibehalten.

## RPDO

(*Receive Process Data Object: Empfangs-Prozessdatenobjekt*) Wird als nicht bestätigte Broadcast-Meldung übertragen oder von einem Erzeugergerät (Producer) an ein Verbrauchergerät (Consumer) in einem CAN-basierten Netzwerk gesendet. Das Sende-PDO vom Producer-Gerät hat eine spezifische Kennung, die dem Empfangs-PDO der Consumer-Geräte entspricht.

## RPI

(*Requested Packet Interval*) Der Zeitraum zwischen den vom Scanner angeforderten zyklischen Datenaustauschvorgängen. EtherNet/IP-Geräte veröffentlichen Daten mit der Rate, die durch das RPI vorgegeben wird, das ihnen vom Scanner zugewiesen wurde, und sie empfangen Nachrichtenrequests vom Scanner bei jedem RPI.

## RTC

(*Real-Time Clock: Echtzeituhr*) Batteriebetriebene Uhr zur Uhrzeit- und Datumsanzeige, die während der gesamten Lebensdauer der Batterie permanent in Betrieb ist, selbst bei ausgeschalteter Steuerung.

**RTP**

(*Real-Time Process*) Der Echtzeitprozess ist die wichtigste Systemtask. Er ist für die Ausführung sämtlicher Echtzeittasks zum jeweils richtigen Zeitpunkt verantwortlich. Die Echtzeitverarbeitung wird über den Sercos-Echtzeit-Buszyklus ausgelöst.

**S****SDO**

(*Service Data Object: Dienstdatenobjekt*) Meldung, die vom Feldbus-Master verwendet wird, um (lesend/schreibend) auf die Objektverzeichnisse von Netzwerkknoten in CAN-basierten Netzwerken zuzugreifen. Zu SDO-Typen gehören Service SDOs (SSDOs) und Client SDOs (CSDOs).

**Sercos**

(*Serial Real-Time Communications System: Serielles Echtzeit-Kommunikationssystem*) Digitaler Steuerungsbus, der Bewegungssteuerungen, Antriebe, E/A-Module, Sensoren und Aktoren für numerisch gesteuerte Maschinen und Systeme miteinander verbindet. Es handelt sich hierbei um eine genormte und offene Schnittstelle zwischen einer Steuerung und intelligenten Digitalgeräten, die für die serielle Hochgeschwindigkeitskommunikation genormter Echtzeit-Regeldaten konzipiert ist.

**SFC**

(*Sequential Function Chart*) Programmiersprache, die aus Schritten mit zugeordneten Aktionen, Übergängen mit zugeordneten Logikbedingungen und Zielverbindungen zwischen Schritten und Übergängen aufgebaut ist. (Der SFC-Standard ist in IEC 848 definiert. Er ist IEC 61131-3-konform.)

**SINT**

(*Signed Integer: Ganzzahl mit Vorzeichen*) 15-Bit-Wert plus Vorzeichen.

**ST**

(*Structured Text: Strukturierter Text*) Programmiersprache, die komplexe und verschachtelte Anweisungen umfasst (z. B. Iterationsschleifen, bedingte Ausführungen oder Funktionen). ST ist IEC 61131-3-kompatibel.

**Steuerung**

Ermöglicht die Automatisierung industrieller Prozesse (auch als speicherprogrammierbare Steuerung oder SPS bezeichnet).

**STRING**

Variable, die einer aus ASCII-Zeichen aufgebauten Zeichenkette entspricht.

**Systemvariable**

Variable, die Steuerungsdaten und Diagnoseinformationen bereitstellt und das Senden von Befehlen an die Steuerung ermöglicht.

## T

### Task

Gruppe von Sections und Unterprogrammen, die zyklisch oder periodisch (MAST-Task) bzw. periodisch (FAST-Task) ausgeführt werden.

Eine Task besitzt eine bestimmte Prioritätsstufe und ist den Eingängen und Ausgängen der Steuerung zugeordnet. Diese E/A werden in Abhängigkeit von der Task aktualisiert.

Eine Steuerung kann über mehrere Tasks verfügen.

### TCP

(*Transmission Control Protocol*) Verbindungsbasiertes Protokoll der Transportschicht, das die zuverlässige, simultane und bidirektionale Übertragung von Daten unterstützt. TCP ist Teil der TCP/IP-Protokollreihe.

### Touchprobe-Eingang

Bei Touchprobe-Eingängen handelt es sich um erweiterte Digitaleingänge. Sie werden für Messfunktionen zur präzisen Positionsbestimmung in Bezug auf einen Messeingang verwendet. Eine aktivierte Touchprobe-Funktion wird ungeachtet des IEC-Programms unabhängig im System ausgeführt. Das IEC-Programm kann Parameter heranziehen, um den Status der Messfunktion zu erfassen. Diese Funktion wird von der Hardware und der Software unterstützt.

### TPDO

(*Transmit Process Data Object: Sende-Prozessdatenobjekt*) Wird in CAN-basierenden Netzwerken als nicht bestätigte Broadcast-Meldung übertragen oder von einem Erzeugergerät (Producer) an ein Verbrauchergerät (Consumer) gesendet. Das Sende-PDO vom Producer-Gerät hat eine spezifische Kennung, die dem Empfangs-PDO der Consumer-Geräte entspricht.

## U

### UDINT

(*Unsigned Double Integer: Doppelte Ganzzahl ohne Vorzeichen*) Codiert über 32 Bit.

### UDP

(*User Datagram Protocol*) Protokoll für den verbindungslosen Modus (nach IETF RFC 768), bei dem Nachrichten in einem Datagramm (Datentelegramm) an einen Zielcomputer in einem IP-Netzwerk gesendet werden. Das UDP-Protokoll ist normalerweise mit dem Internet Protocol (IP) gebündelt. UDP/IP-Nachrichten erwarten keine Antwort und sind deshalb ideal für Anwendungen, in denen verlorene Pakete keine Neuübertragung erfordern (z.B. Streaming-Video und Netzwerke, die Echtzeitverhalten verlangen).

### UINT

(*Unsigned Integer: Ganzzahl ohne Vorzeichen*) Codiert über 16 Bit.

**Urheber**

Das Gerät, in der Regel die Steuerung, das beim expliziten EtherNet/IP-Nachrichtenaustausch den Austausch von Daten mit Zielnetzwerkgeräten initiiert.

Siehe auch *Ziel*.

**V****Variable**

Speichereinheit, die von einem Programm adressiert und geändert werden kann.

**W****WORD**

In einem 16-Bit-Format codierter Typ.

**Z****Ziel**

Das Gerät, das beim expliziten EtherNet/IP-Nachrichtenaustausch auf die von den Urhebergeräten gesendeten Requests zum Austausch von Daten antwortet.

Siehe auch *Urheber*.





## A

ASCII-Manager, *232*  
Ausgangsforcierung, *65*  
Ausgangverhalten, *64, 64, 65*

## B

Bewegung  
  Leistung, *46*  
  Programmieranforderungen, *46*  
Bewegungstask, *44*  
Bibliotheken, *25*  
  FTPRemoteFileHandling, *185*

## C

CANopen-Schnittstelle, *202*  
changeIPAddress, *261*  
  Ändern der IP-Adresse der Steuerung,  
  *261*

## D

Datentyp  
  ET\_Sercos3CmdType, *286*  
  ET\_Sercos3IDNTyp, *287*  
  ST\_SercosConfiguration, *283*  
  ST\_SercosConfigurationDevice, *284*  
Download Application-Befehl, *70*

## E

ET\_Sercos3CmdType  
  Datentyp, *286*  
ET\_Sercos3IDNTyp  
  Datentyp, *287*

## Ethernet

  Dienste, *173*  
  EtherNet/IP-Adapter, *124*  
  FTP-Server, *182*  
  Funktionsbaustein changeIPAddress,  
  *261*  
  Modbus TCP-Client/Server, *180*  
  Modbus TCP-Slave-Gerät, *186*  
EtherNet/IP  
  Azyklischer Datenaustausch, *129*  
Externes Ereignis, *42*

## F

FB\_SercosProcedureCommandAsync  
  Asynchrones Senden von Sercos-Befeh-  
  len, *307*  
FB\_SercosReadServiceDataAsync  
  Asynchrones Lesen von Sercos-Daten,  
  *303*  
FB\_SercosWriteServiceDataAsync  
  Asynchrones Schreiben von Sercos-Da-  
  ten, *305*  
FC\_SercosGetConfiguration  
  Funktion, *289*  
FC\_SercosReadServiceData  
  Funktion, *290*  
FC\_SercosReadServiceDataByTopAddr  
  Funktion, *293*  
FC\_SercosScanConfiguration  
  Funktion, *296*  
FC\_SercosWriteServiceData  
  Funktion, *298*  
FC\_SercosWriteServiceDataByTopAddr  
  Funktion, *300*  
Firewall  
  Konfiguration, *194*  
  Skriptbefehle, *195*  
  Standard-Skriptdatei, *194*  
Firmwareaktualisierung, *255*  
FTP-Client, *185*

## FTP-Server

Ethernet, *182*

FTPRemoteFileHandling-Bibliothek, *185*

## Funktion

FC\_SercosGetConfiguration, *289*

FC\_SercosReadServiceData, *290*

FC\_SercosReadServiceDataByTopAddr,  
*293*

FC\_SercosScanConfiguration, *296*

FC\_SercosWriteServiceData, *298*

FC\_SercosWriteServiceDataByTopAddr,  
*300*

**G**

Geberkonfiguration, *102*

## GetSerialConf

Abrufen der Konfiguration der seriellen  
Leitung, *310*

**H**

Hardware-Initialisierungswerte, *64*

**I**

Integrierte E/A-Konfiguration, *94*

## IP-Adresse

changeIPAddress, *261*

**K**

Konfigurationsparameter, *78*

**L**

LMC078 Sercos3

FB\_SercosProcedureCommandAsync,  
*307*

FB\_SercosReadServiceDataAsync, *303*

FB\_SercosWriteServiceDataAsync, *305*

LXM32S-Konfiguration, *220*

**M**

M2•-Kommunikation

GetSerialConf, *310*

SetSerialConf, *311*

Modbus

Protocols, *180*

Modbus TCP-Client/Server

Ethernet, *180*

Modbus-E/A-Scanner, *235*

Modbus-Manager, *244*

**N**

Neustart, *68*

**P**

Parametertypen, *29*

PROFIBUS DP

Azyklischer Datenaustausch, *120*

Protokoll

IP, *175*

Protokolle, *173*

Modbus, *180*

**R**

Remanente Variablen, *72*

Reset (kalt), *67*

Reset (Ursprung), *68*

Reset (warm), *67*

Run-Befehl, *66*

**S**

Sercos

FB\_SercosProcedureCommandAsync,  
*307*

FB\_SercosReadServiceDataAsync, *303*

FB\_SercosWriteServiceDataAsync, *305*

Hinzufügen von Geräten, *215*

Hinzufügen von Geräten von Drittanbie-  
tern, *215*

Schnittstellenkonfiguration, *211*

---

- Sercos-Geräte von Drittanbietern, Hinzufügen, *215*
- Sercos-Scan, *216*
- SERIAL\_CONF, *313*
- Serielle Leitung
  - ASCII-Manager, *232*
  - GetSerialConf, *310*
  - Modbus-Manager, *244*
  - SetSerialConf, *311*
- SetSerialConf, *311*
  - Einstellen der Konfiguration der seriellen Leitung, *311*
- Skriptbefehle
  - Firewall, *195*
- Skriptdatei
  - Syntaxregeln, *199*
- Software-Initialisierungswerte, *64*
- Speicherorganisation, *31*
- ST\_SercosConfiguration
  - Datentyp, *283*
- ST\_SercosConfigurationDevice
  - Datentyp, *284*
- Steuerungskonfiguration
  - SPS-Einstellungen, *91*
  - Steuerungsauswahl, *88*
- Stop-Befehl, *66*

## T

- Task
  - Ereignistask, *42*
  - Externe Ereignistask, *42*
  - Statustask, *43*
  - Typen, *41*
  - Watchdogs, *47*
  - Zyklische Task, *41*

## U

- Überblick über den Sercos-Standard , *208*

## Z

- Zustandsdiagramm, *54*

