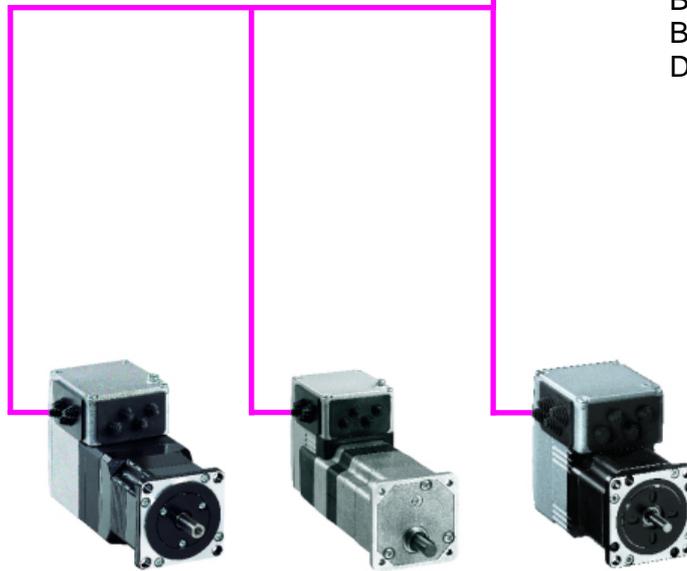
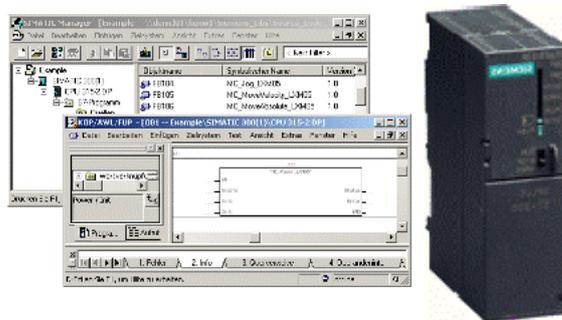


Technische Dokumentation

Handbuch

**Step 7 Bibliothek für
IFS, IFE, IFA**

Ausgabe: V1.02, 01.2008



Berger Lahr GmbH & Co.KG
Breslauer Str. 7
D-77933 Lahr

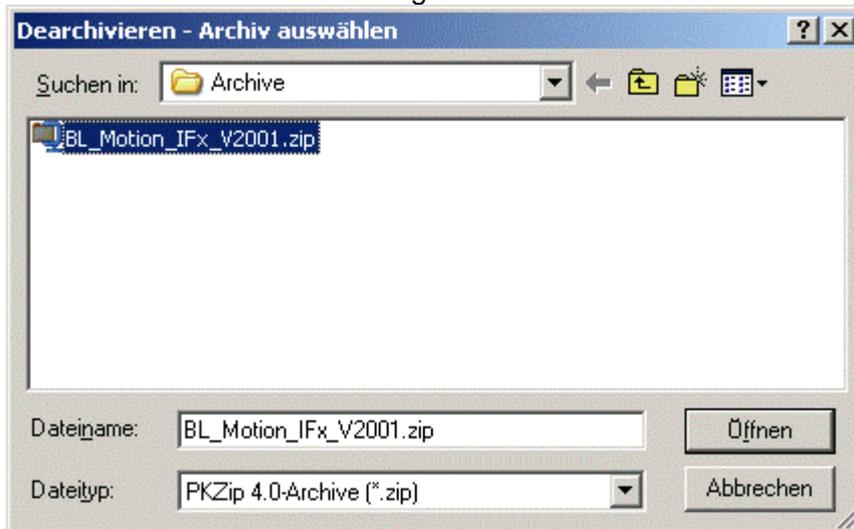
Inhalt

1	Bibliothek extrahieren	3
2	Neues Projekt anlegen	5
3	Hardware konfigurieren	6
4	GSD installieren	7
5	Antrieb ins PB-Netzwerk einbinden.....	9
6	E-/A - Adressen editieren	11
6.1	Parameterdatenkanal	11
6.2	Prozessdatenkanal	12
7	Beschreibung der Bibliotheksbausteine.....	13
7.1	Achsstruktur ins Projekt kopieren	13
7.2	Achs-Referenz anlegen.....	14
8	Bibliotheksbausteine.....	16
8.1	Prinzipielles Aufrufschema	16
8.2	Bedeutung gemeinsamer Parameter.....	17
8.3	Signaldiagramme.....	18
8.4	Initialisierung	19
8.5	Manuellfahrt	22
8.6	Referenzierung	24
8.7	Punkt-zu-Punkt Betrieb.....	27
8.8	Geschwindigkeitsbetrieb.....	30
8.9	Elektronisches Getriebe	31
8.10	Stoppen	34
8.11	Schnelle Positionserfassung	35
8.12	Parameter lesen.....	38
8.13	Parameter schreiben.....	48
8.14	Ein-/Ausgänge	52
8.15	Fehlerbehandlung	58
9	Glossar	60
10	Liste der Fehlernummern.....	62
11	Parameterliste für UP- und Downloadfunktion.....	65

1 Bibliothek extrahieren

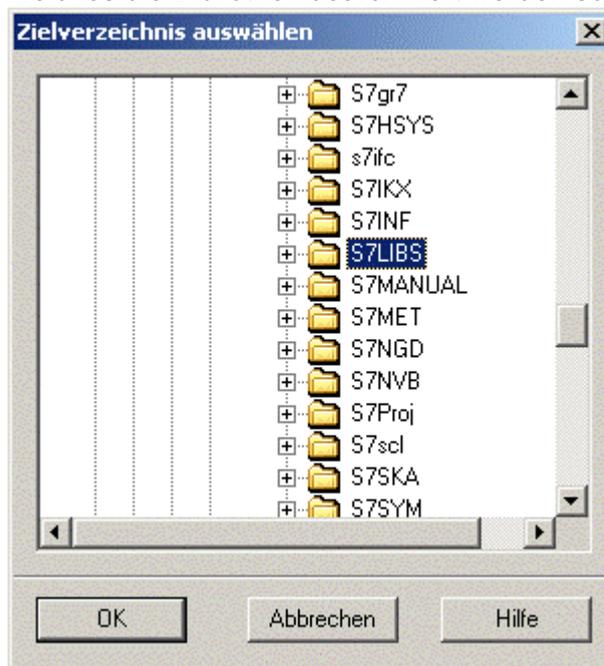
Um die Bausteine der Bibliothek zu verwenden müssen Sie zuerst das Archiv „BL_Motion_IFx_Vxxxx.zip“ mit der Step7 Software dearchivieren. Dazu wählen Sie im Menü **Datei** das Untermenü **Dearchivieren** aus.

Es öffnet sich der Dialog zur Auswahl des Archivs:



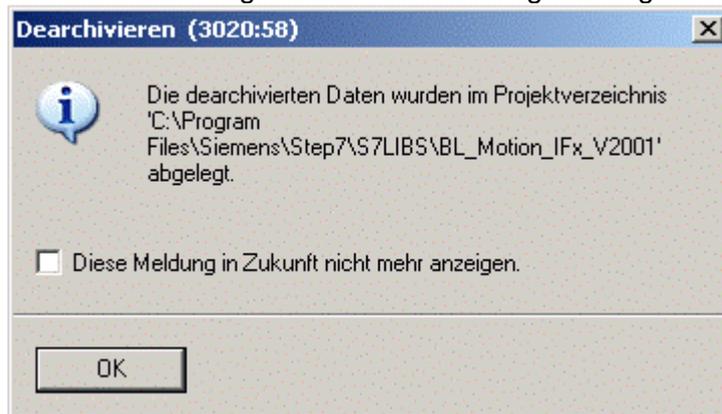
Browsen Sie in das Ablageverzeichnis des Bibliotheksarchivs und markieren die Bibliothek. Bestätigen Sie ihre Auswahl mit „Öffnen“!

Im folgenden Fenster wählen Sie das Zielverzeichnis, in welches die Bibliothek dearchiviert werden soll.



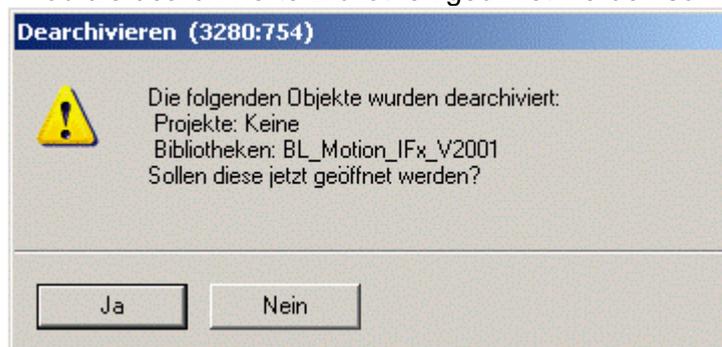
Markieren Sie das gewünschte Verzeichnis und Bestätigen Sie ihre Auswahl mit „OK“!
 Empfehlung: <Siemens Verzeichnis>\Step7\ S7LIBS
 Bsp.: C:\Programme\Siemens\Step7\ S7LIBS

Abhängig von der Konfiguration Ihrer Step7 Software wird die erfolgreiche Dearchivierung bestätigt.



Bestätigen Sie mit „OK“!

In einem weiteren Bestätigungsfenster werden Sie gefragt, ob die dearchivierte Bibliothek geöffnet werden soll.



Bestätigen Sie mit „Nein“!

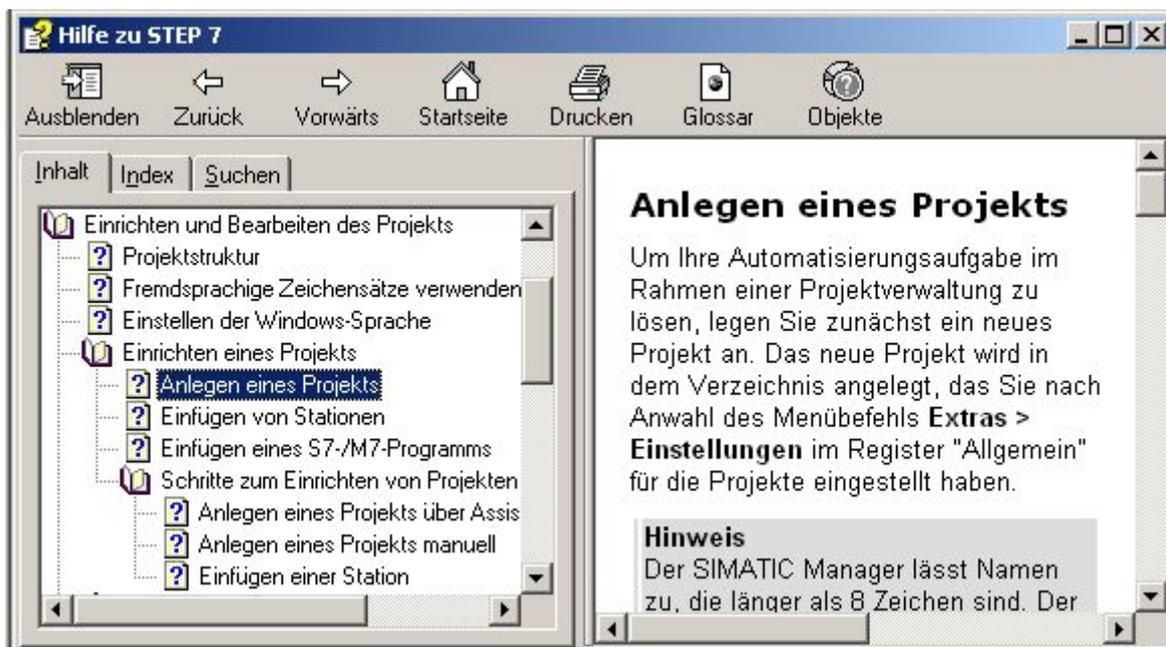
Hinweis: Sie können selbstverständlich auch die Bibliothek öffnen und die Bausteine, welche für Ihre Applikation relevant sind mit der Kopierfunktion der Step 7 Software manuell in Ihre Applikation kopieren.

Sie haben nun die Bibliothek erfolgreich dearchiviert und können somit in den Step7-Editoren auf die Bausteine zugreifen und diese wie nachfolgend beschrieben in Ihrer Applikation verwenden.

Für die Funktion der Bibliothek ist es zwingend erforderlich, die zur Bibliothek gehörende GSD (Geräte Stamm Datei) zu verwenden (**BLS70746.GSD**). Diese GSD unterstützt alle drei Antriebstypen. Dafür muss zuerst die entsprechende GSD installiert werden, damit sie im Hardware Manager zur Verfügung steht. Dazu muss ein Projekt angelegt und der Hardware Manager gestartet werden.

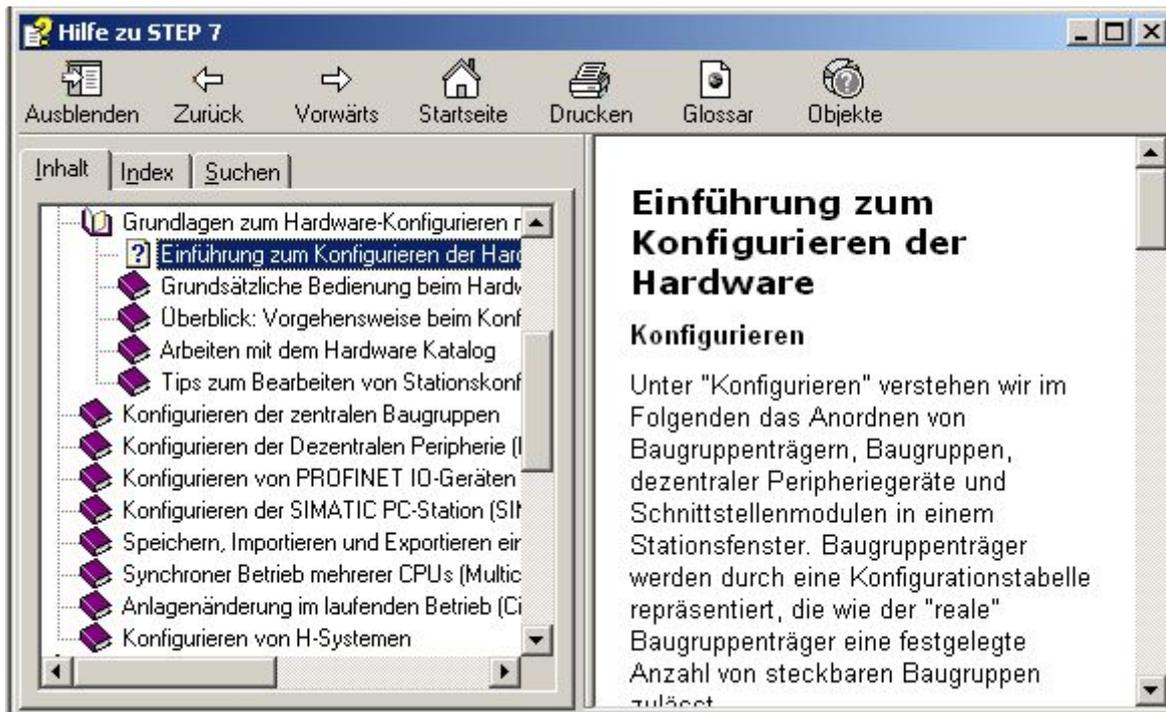
2 Neues Projekt anlegen

Legen Sie ein neues Projekt an. Dazu wählen Sie im Menü **Datei** das Untermenü **Neu** oder **Assistent 'Neues Projekt'** aus. Die Kenntnisse für das Erstellen eines Projekts werden vorausgesetzt, deshalb wird an dieser Stelle auf die Online-Hilfe sowie die Dokumentationen von Step7 bzw. Siemens verwiesen.



3 Hardware konfigurieren

Nachdem Sie ein neues Projekt erstellt haben müssen Sie die zu verwendende Hardware festlegen. Dazu fügen Sie zuerst im Menü **Einfügen** über das Untermenü **Station** eine Station ein. Anschließend markieren Sie die eingefügte Station und Starten aus dem Menü **Bearbeiten** durch Auswahl des Untermenüs **Objekt öffnen** den Hardware Konfigurator. Die Kenntnisse für das Konfigurieren der Hardware werden vorausgesetzt, deshalb wird an dieser Stelle auf die Online-Hilfe sowie die Dokumentationen von Step7 bzw. Siemens verwiesen.



Um den Antrieb in das Profibus-Netzwerk einbinden zu können müssen Sie zuerst, wie im folgenden Abschnitt beschrieben, die zum entsprechenden Antrieb gehörende GSD installieren. Sollten dies bereits erfolgt sein können sie mit dem Kapitel [Antrieb ins PB-Netzwerk einfügen](#) fortfahren.

4 GSD installieren

Hinweis: Die Bibliothek darf nur mit der zugehörigen GSD **BLS70746.GSD** verwendet werden (Diese GSD unterstützt alle drei Antriebstypen IFS, IFE, IFA).

Mit der Standard GSD funktioniert die Bibliothek nicht !

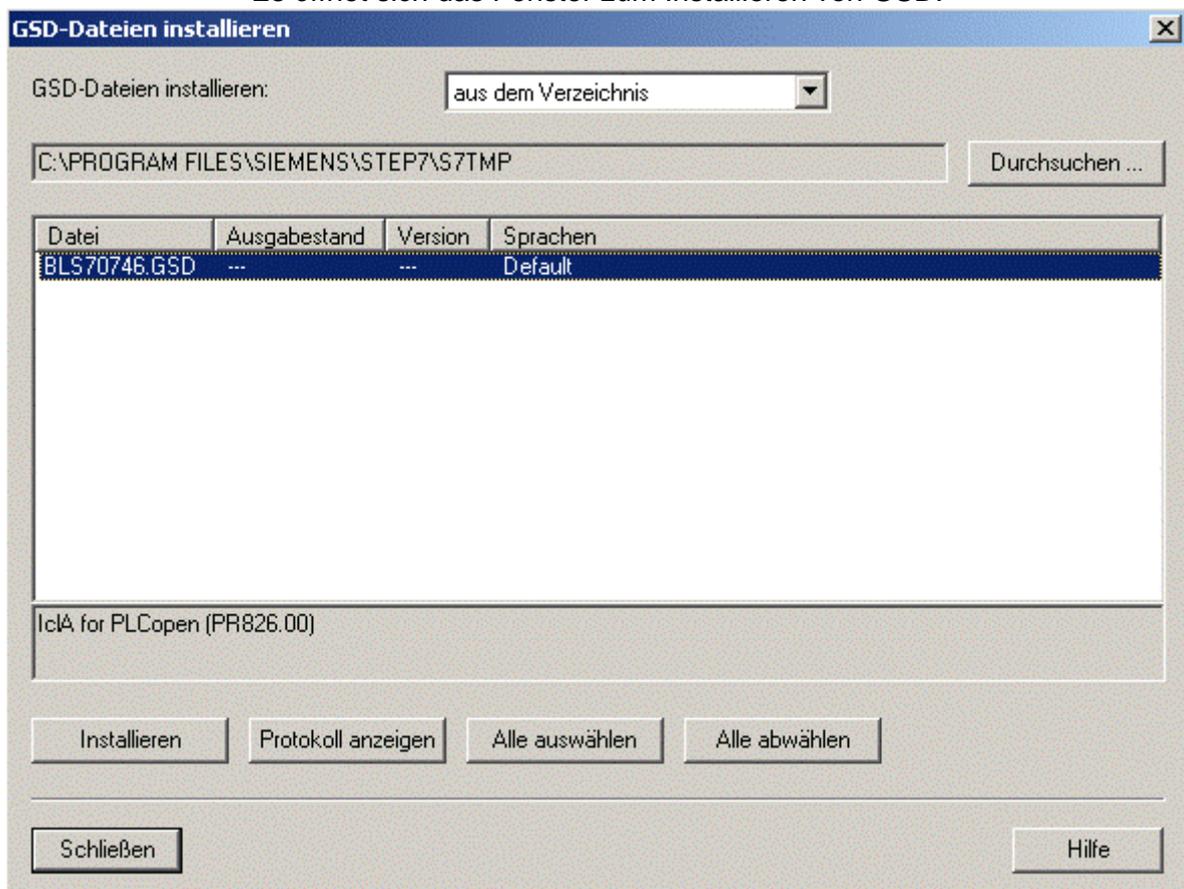
Kopieren Sie die GSD in ein beliebiges Verzeichnis auf Ihrer Festplatte.

Empfehlung: <Siemens Verzeichnis>\Step7\ S7TMP

Bsp.: C:\Programme\Siemens\Step7\ S7TMP

Öffnen Sie Anschließend im Hardware Konfigurator den Dialog zum Installieren von GSD-Dateien. Dazu wählen Sie im Menü **Extras** das Untermenüs **GSD-Dateien installieren...**

Es öffnet sich das Fenster zum Installieren von GSD.



Browsen Sie in das Ablageverzeichnis der GSD und markieren diese. Bestätigen Sie ihre Auswahl mit „Installieren“!

Bestätigung der erfolgreichen Installation.

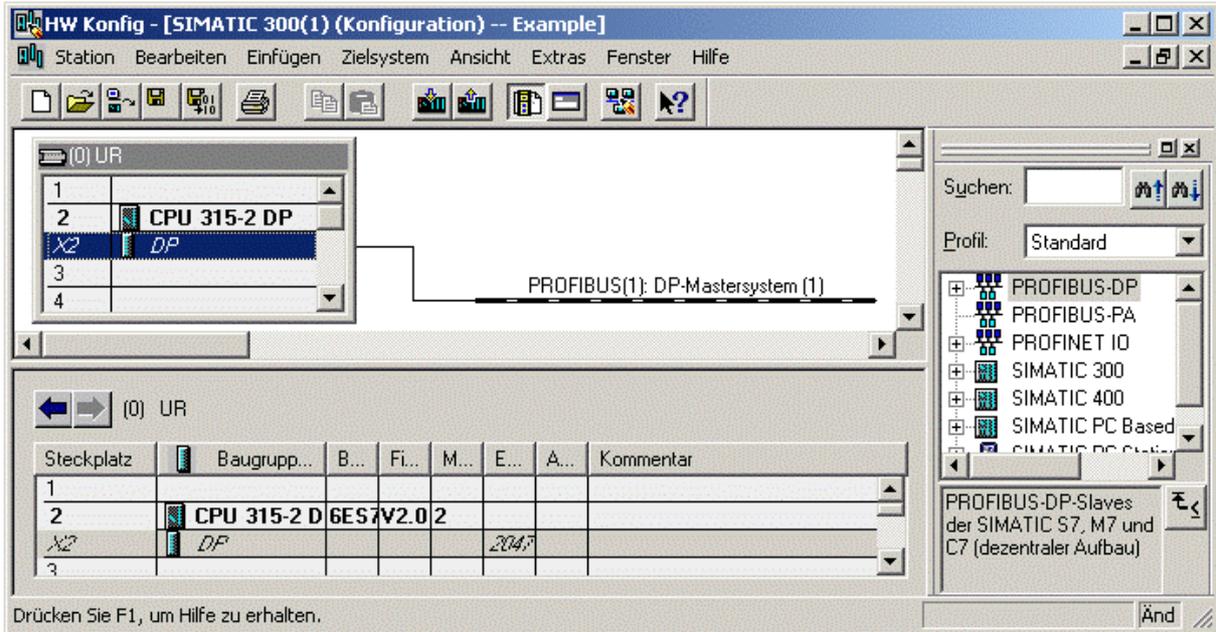


Schließen Sie das Bestätigungsfenster mit „OK“.

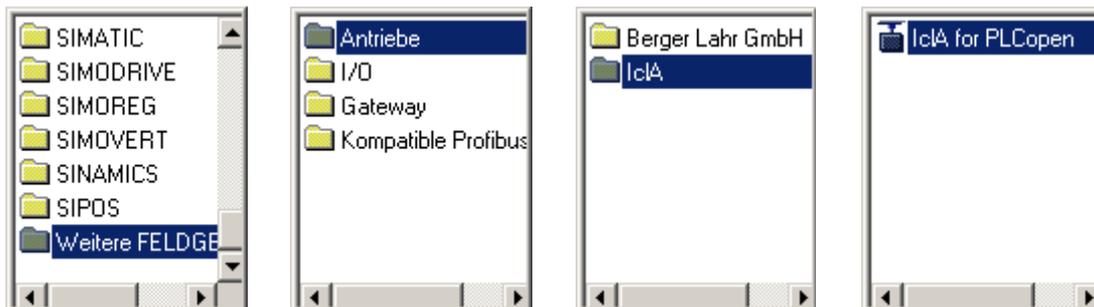
Schließen Sie nun auch das Fenster zum Installieren von GSD mit „Schließen“, um in den Hardware Konfigurator zurückzukehren und im Weiteren die Hardware einzurichten.

5 Antrieb ins PB-Netzwerk einbinden

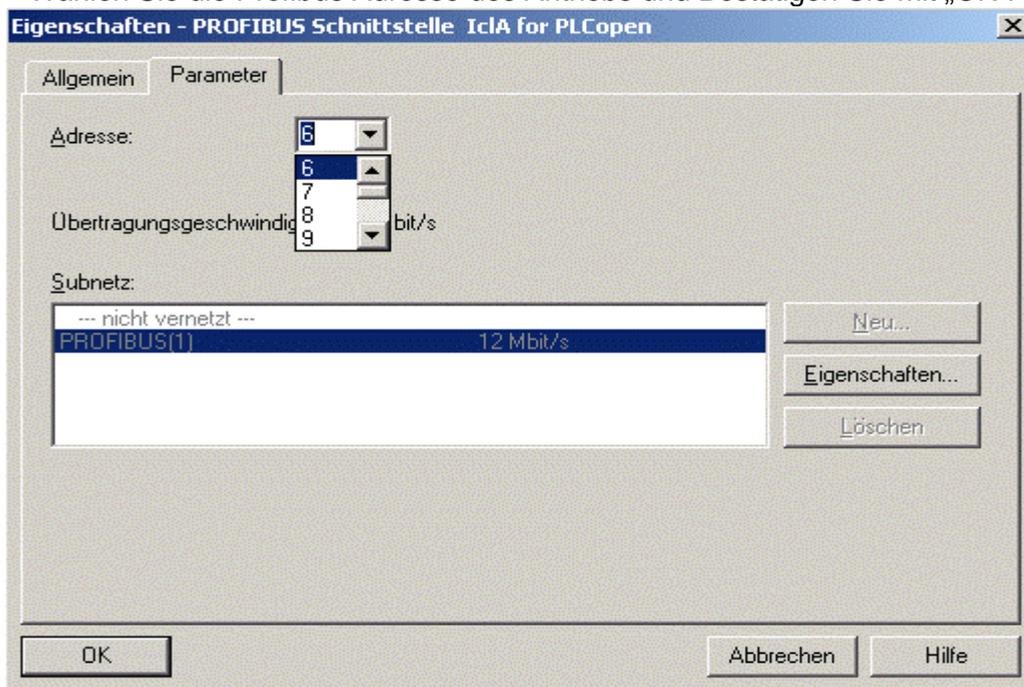
Voraussetzung zum Einbinden der Antriebe ist, dass Sie einen Baugruppenträger, eine CPU und ein DP-Mastersystem in Ihre Hardwarekonfiguration eingefügt haben.



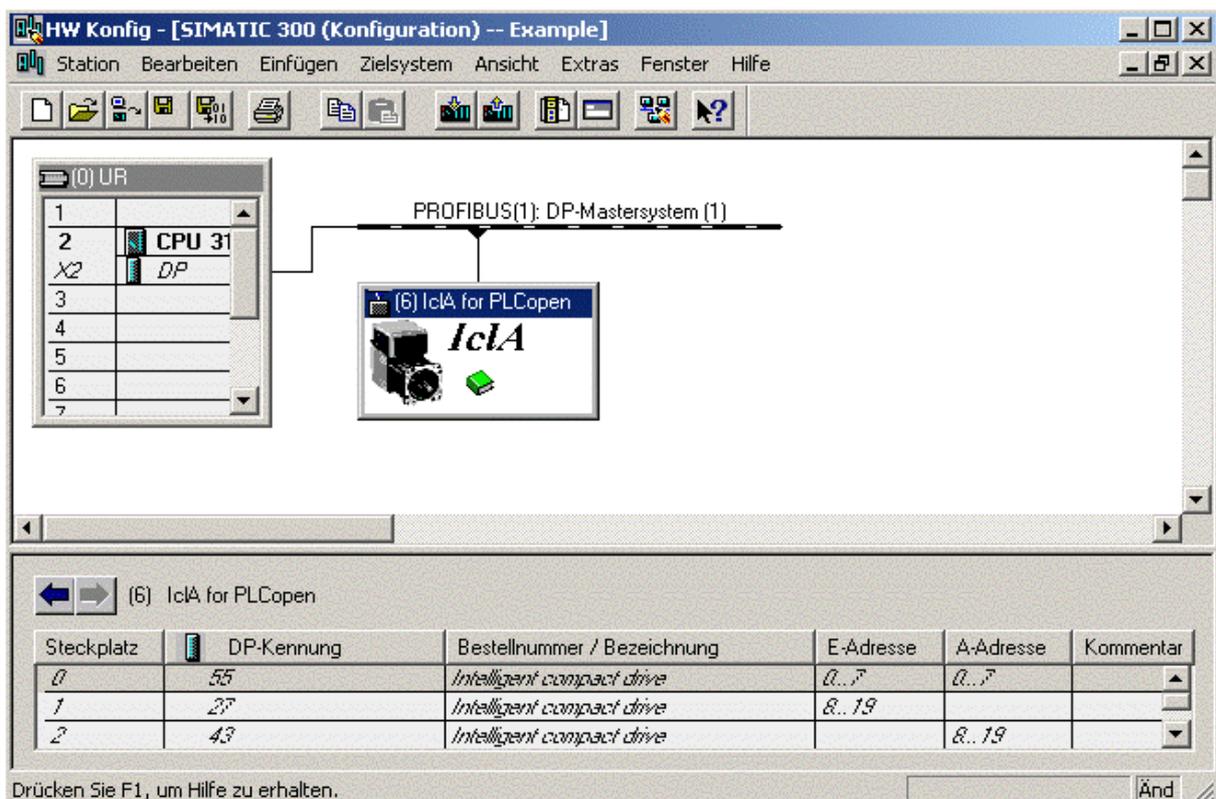
Markieren Sie im Hardware Konfigurator das Mastersystem und über das Menü **Einfügen/Objekt einfügen...** wählen Sie über die Untermenüs **Weitere Feldgeräte**, **Antriebe** und **IcIA** den Eintrag **IcIA for PLCopen** aus.



Wählen Sie die Profibus Adresse des Antriebs und Bestätigen Sie mit „OK“.



Sie haben nun den Antrieb als Profibus Slave eingefügt und an das DP-Mastersystem angebunden. Für die Kommunikation mit dem Antrieb verwendet die Bibliothek zwei Kommunikationskanäle: Den Parameterdatenkanal (8 Byte) auf dem Steckplatz eins und den Prozessdatenkanal (12 Byte) auf den Steckplätzen zwei und drei.



Für die Kommunikationskanäle müssen abschliessend, wie im folgenden Kapitel beschrieben, noch die E-/A - Adressen eingestellt werden.

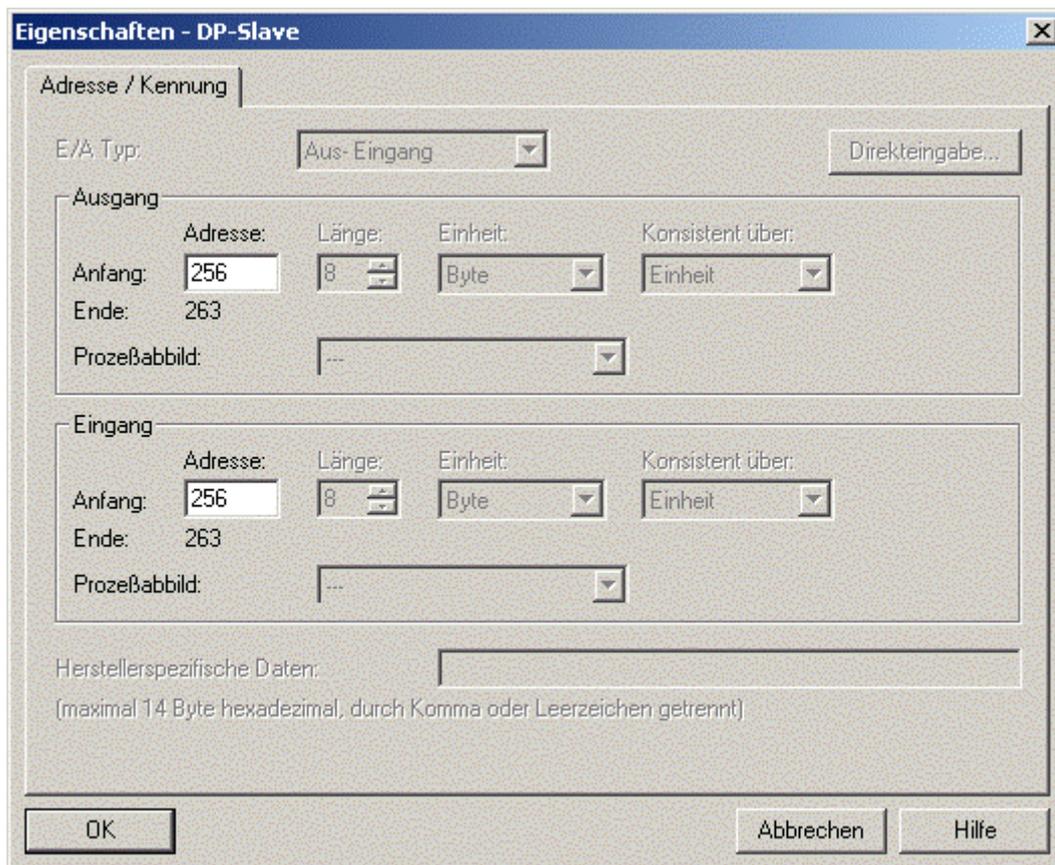
6 E-/A - Adressen editieren

Als letzte Einstellung im Hardware Konfigurator müssen Sie die Eingangs- und Ausgangsadressen der Kommunikationskanäle einstellen. Diese Adressen sind abhängig von der projektierten CPU und von der Ausbaustufe der SPS.

6.1 Parameterdatenkanal

Um die Adressen für den Parameterdatenkanal zu definieren müssen Sie den Steckplatz 1 markieren und im Menü **Bearbeiten** das Untermenü **Objekteigenschaften...** auswählen. Editieren Sie wie unten dargestellt jeweils einen freien Adressbereich für die Ausgangs- und Eingangsdaten.

Hinweis: Diese Adressen müssen Sie der Bibliothek bekannt geben. Die Anfangsadresse des Ausgangsbereichs muss der Funktion [MC_Init_IFx](#) am Eingang AdrParameterOut und die Anfangsadresse des Eingangsbereichs am Eingang AdrParameterIn übergeben werden.



Eigenschaften - DP-Slave

Adresse / Kennung

E/A Typ: Aus- Eingang Direkteingabe...

Ausgang

	Adresse:	Länge:	Einheit:	Konsistent über:
Anfang:	256	8	Byte	Einheit
Ende:	263			
Prozeßabbild:	---			

Eingang

	Adresse:	Länge:	Einheit:	Konsistent über:
Anfang:	256	8	Byte	Einheit
Ende:	263			
Prozeßabbild:	---			

Herstellerspezifische Daten:

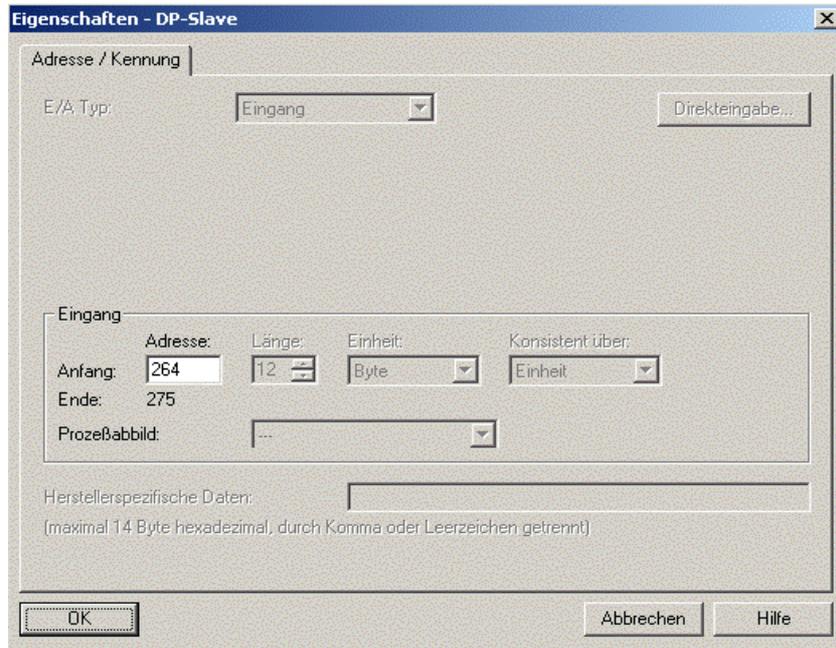
(maximal 14 Byte hexadezimal, durch Komma oder Leerzeichen getrennt)

OK Abbrechen Hilfe

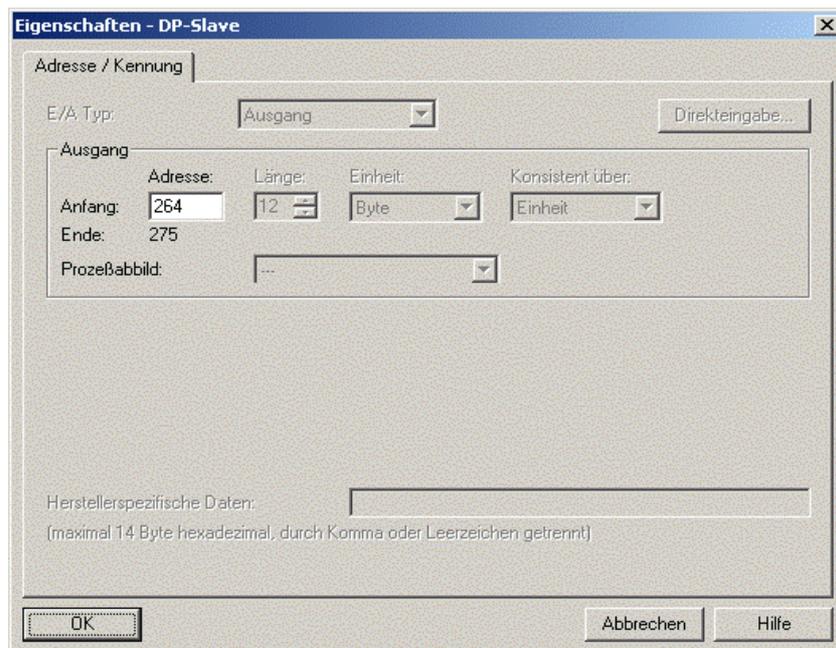
6.2 Prozessdatenkanal

Die Adressen des Prozessdatenkanals werden auf die gleiche Art und Weise editiert wie die Adressen des Parameterdatenkanals. Mit dem Unterschied, dass Eingangsbereich und Ausgangsbereich auf unterschiedlichen Steckplätzen liegen.

Hinweis: Diese Adressen müssen Sie der Bibliothek bekannt geben. Die Anfangsadresse des Ausgangsbereichs muss der Funktion [MC_Init_IFx](#) am Eingang AdrProcessdataOut und die Anfangsadresse des Eingangsbereich am Eingang AdrProcessdataIn übergeben werden.



The screenshot shows the 'Eigenschaften - DP-Slave' dialog box with the 'Adresse / Kennung' tab selected. The 'E/A Typ:' dropdown is set to 'Eingang'. The 'Eingang' section contains the following fields: 'Anfang:' with a value of 264, 'Ende:' with a value of 275, 'Länge:' with a value of 12, 'Einheit:' set to 'Byte', and 'Konsistent über:' set to 'Einheit'. There is also a 'Prozeßabbild:' dropdown menu. At the bottom, there is a text field for 'Herstellerspezifische Daten:' with a note '(maximal 14 Byte hexadezimal, durch Komma oder Leerzeichen getrennt)'. Buttons for 'OK', 'Abbrechen', and 'Hilfe' are visible at the bottom.



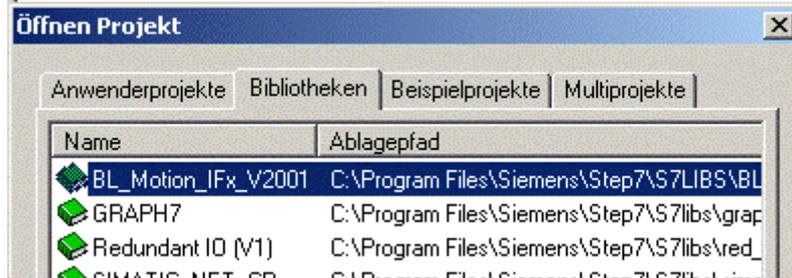
The screenshot shows the 'Eigenschaften - DP-Slave' dialog box with the 'Adresse / Kennung' tab selected. The 'E/A Typ:' dropdown is set to 'Ausgang'. The 'Ausgang' section contains the following fields: 'Anfang:' with a value of 264, 'Ende:' with a value of 275, 'Länge:' with a value of 12, 'Einheit:' set to 'Byte', and 'Konsistent über:' set to 'Einheit'. There is also a 'Prozeßabbild:' dropdown menu. At the bottom, there is a text field for 'Herstellerspezifische Daten:' with a note '(maximal 14 Byte hexadezimal, durch Komma oder Leerzeichen getrennt)'. Buttons for 'OK', 'Abbrechen', and 'Hilfe' are visible at the bottom.

Speichern und Übersetzen Sie zum Abschluss der Konfiguration Ihre Einstellungen mit **Speichern und Übersetzen** im Menü **Station** des Hardware Konfigurators.

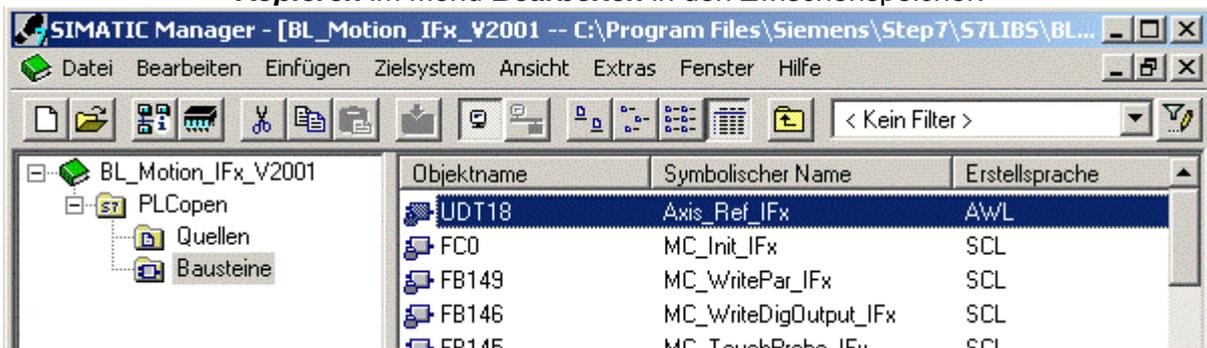
7 Beschreibung der Bibliotheksbausteine

7.1 Achsstruktur ins Projekt kopieren

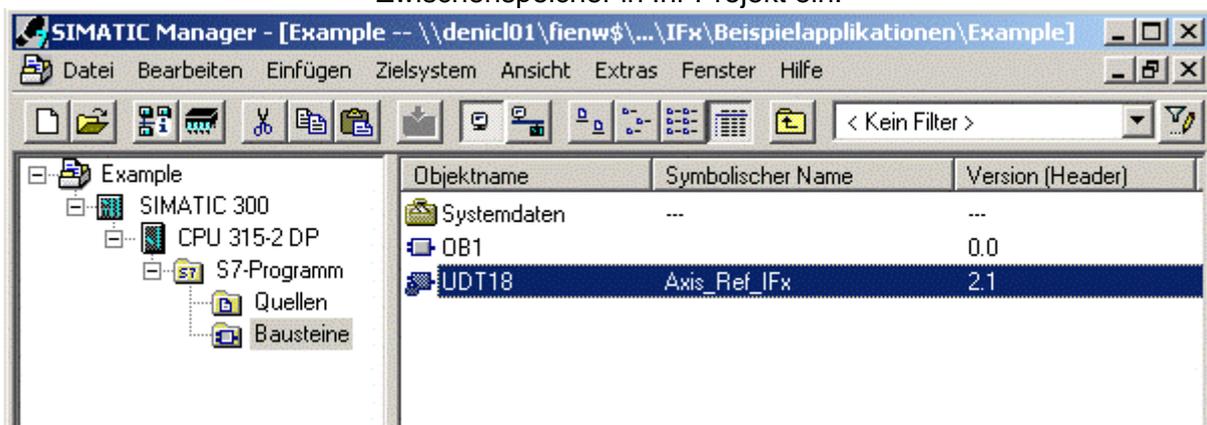
Öffnen Sie im SIMATIC Manager die zuvor extrahierte Bibliothek.



Markieren Sie Anschließend den Baustein UDT18 und kopieren ihn mit dem Kommando **Kopieren** im Menü **Bearbeiten** in den Zwischenspeicher.



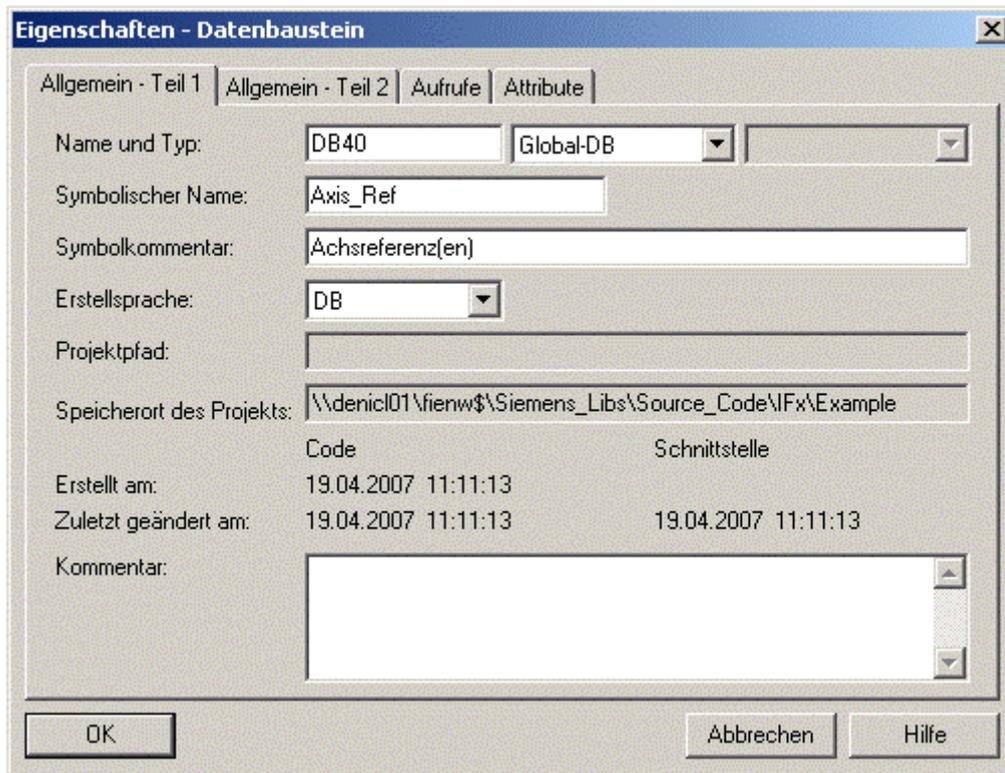
Schließen Sie die Bibliothek und markieren in Ihrem Projekt den Bausteinordner. Fügen Sie nun den Baustein mit dem Kommando **Einfügen** im Menü **Bearbeiten** aus dem Zwischenspeicher in Ihr Projekt ein.



Hinweis: Sie können selbstverständlich die Nummer des UDT ändern. Verwenden Sie dazu im Menü **Bearbeiten** das Kommando **Umbenennen**.

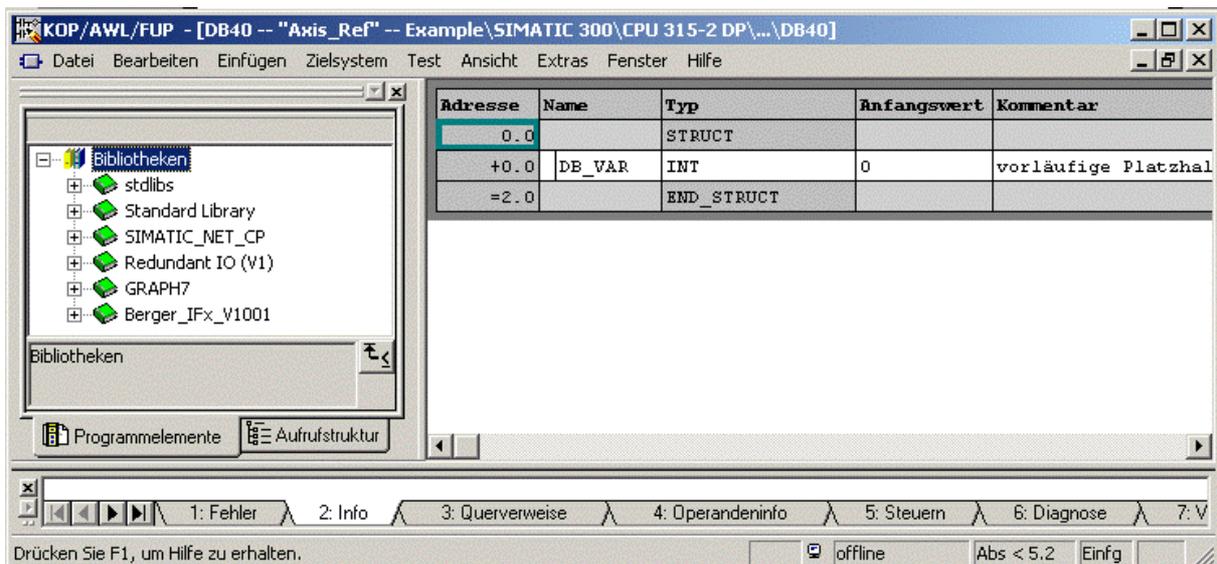
7.2 Achs-Referenz anlegen

Erzeugen Sie im Bausteinordner einen globalen Datenbaustein, den Achs-DB.
 Im Simatic Manager **Einfügen / S7-Baustein / Datenbaustein** auswählen.
 Editieren Sie die Eigenschaften des Bausteins nach Ihren Bedürfnissen. Zu Beachten ist dabei lediglich, dass der Baustein ein Global-DB ist.



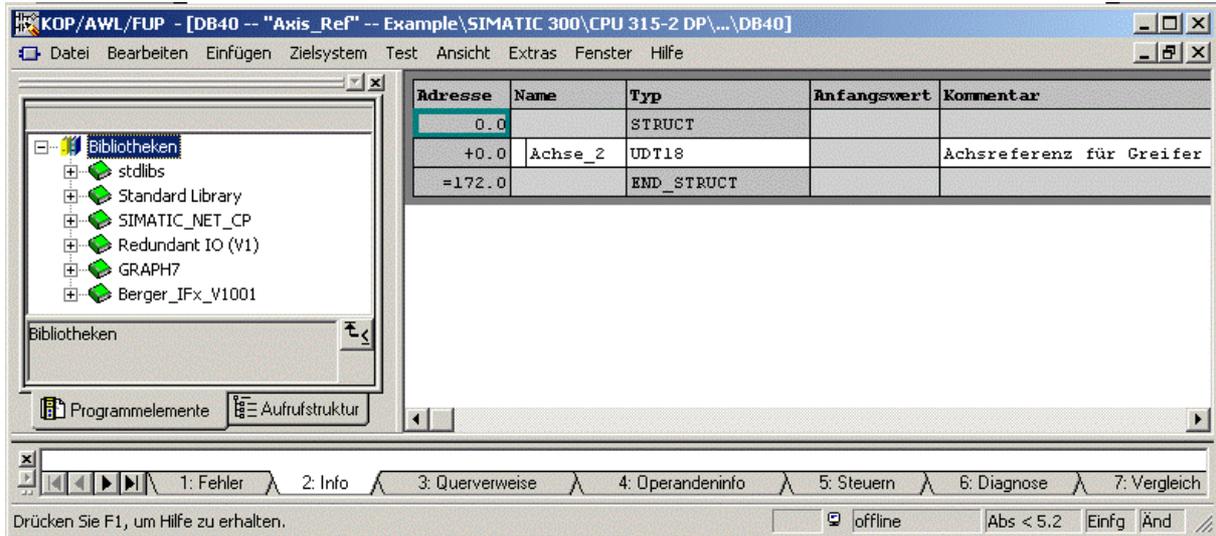
Bestätigen Sie ihre Eingaben mit „OK“.

Öffnen Sie Anschließend den Baustein, indem Sie diesen markieren und im Menü **Bearbeiten** das Untermenü **Objekt öffnen** auswählen. Es wird der KOP/AWL/FUP-Editor gestartet und Sie können den Datenbaustein bearbeiten.



Step 7 Motion Bibliothek für IFS, IFE und IFA

Legen Sie nun die Achs-Referenz an, indem Sie eine Variable vom Typ UDT18 definieren. Sollten Sie den UDT umbenannt haben müssen Sie den Typ ebenfalls umbenennen.



The screenshot shows the SIMATIC Manager interface with the following data table:

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Achse_2	UDT18		Achsreferenz für Greifer
=172.0		END_STRUCT		

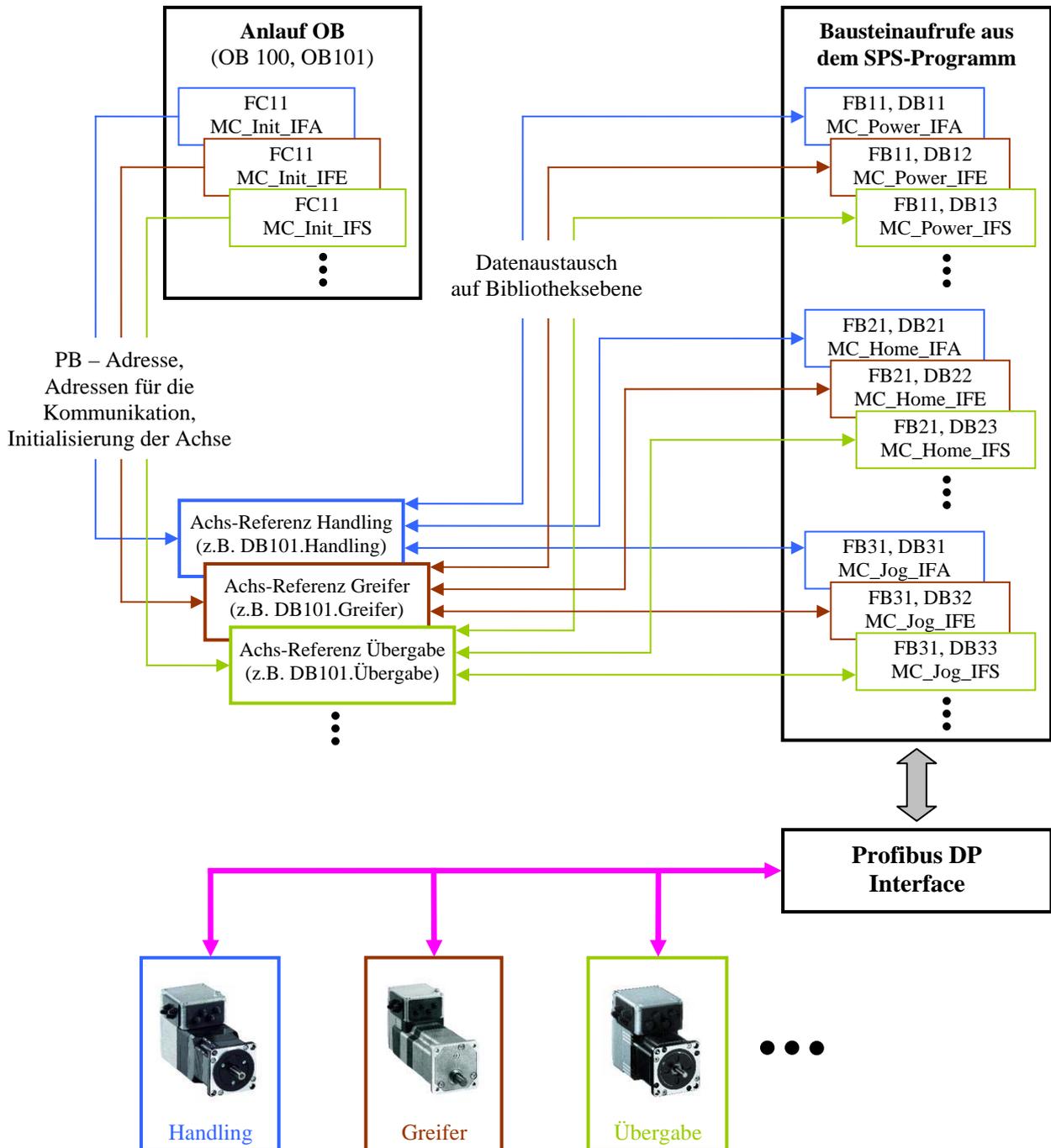
The interface also shows a 'Bibliotheken' (Libraries) tree on the left and a status bar at the bottom with navigation buttons and the text 'Drücken Sie F1, um Hilfe zu erhalten.' (Press F1 for help).

Hinweis: Sie können bei der Verwendung mehrerer Antriebe alle Achs-Referenzen in einem Datenbaustein anlegen !
Diese Art der Definition der Achs-Referenz stellt nur eine von mehreren Möglichkeiten dar. Selbstverständlich können auch andere Konzepte angewendet werden, es muss nur sichergestellt werden, dass alle Bausteine der einer Achse die selbe Struktur verwenden.

Sie haben nun die Grundvoraussetzung geschaffen, um mit der eigentlichen Programmierung Ihrer Applikation zu beginnen.

8 Bibliotheksbausteine

8.1 Prinzipielles Aufrufschema

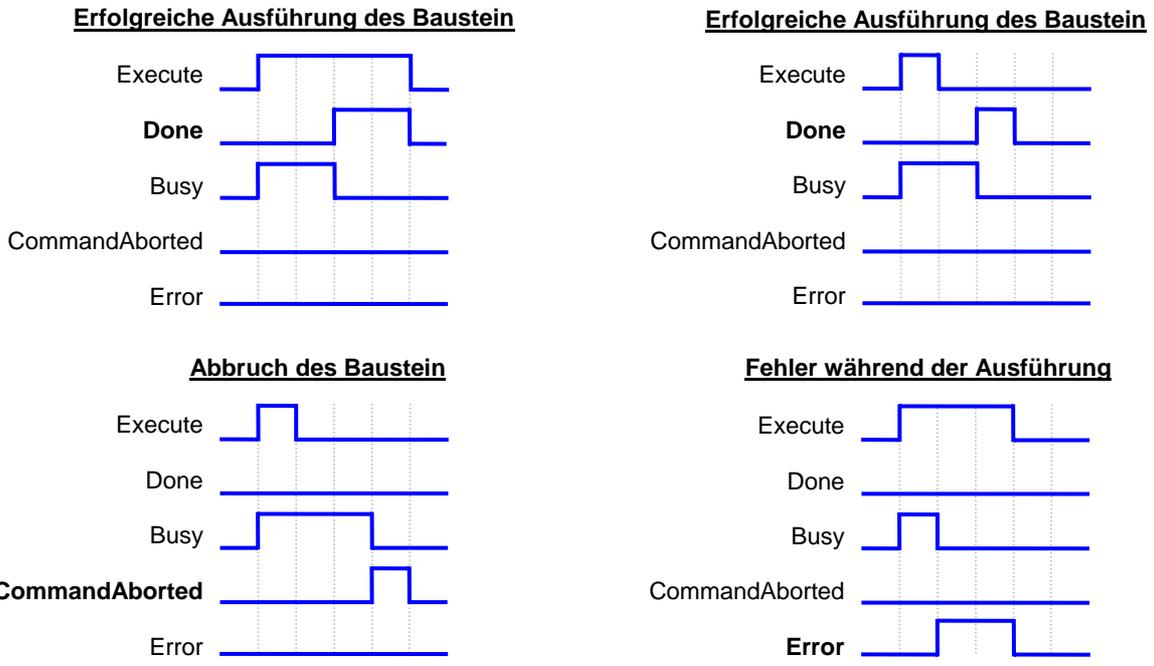


8.2 Bedeutung gemeinsamer Parameter

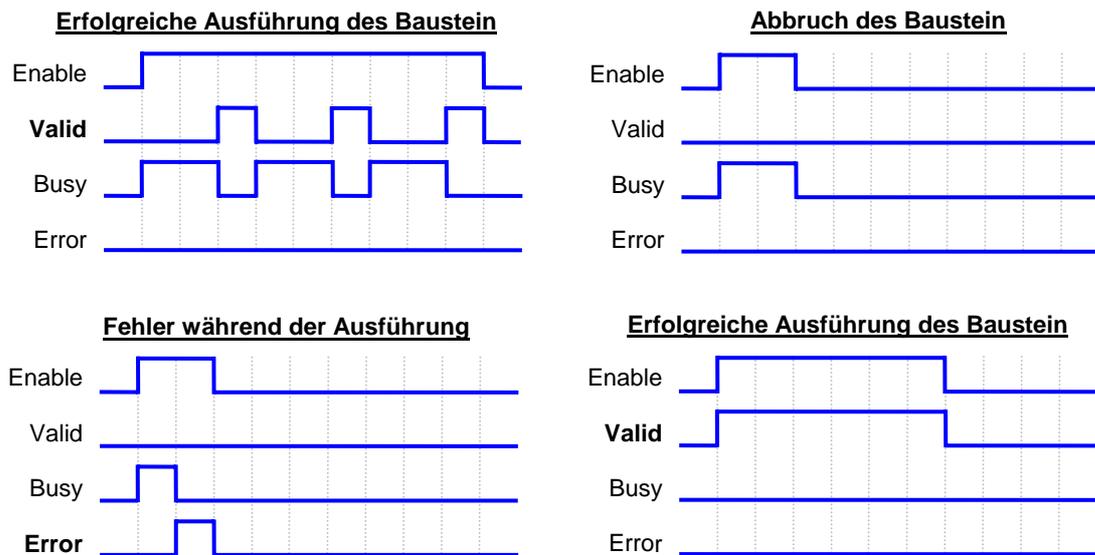
Par-typ	Parameter	Datentyp	Bedeutung
IN	Enable	BOOL	Startet (=TRUE) und beendet (=FALSE) die Ausführung des Bausteins. Der Baustein wird immer wieder ausgeführt, so lange TRUE übergeben wird (pegelsensitiv).
	Execute	BOOL	<p>Startet bei steigender Flanke die einmalige Ausführung des Bausteins. Bei allen Bewegungsbausteinen (außer MC_Home) werden bei steigender Flanke während der Ausführung die Eingangsparameter gelesen und die Bewegung mit den dann neuen Parametern fortgesetzt.</p> <p>Nach beendeter Ausführung (Busy = FALSE) des Bausteins bleiben die Ausgangsparameter so lange erhalten, bis FALSE übergeben wird. Die fallende Flanke löscht die Ausgangsparameter. Ist bei beendeter Ausführung der Eingang bereits FALSE werden die Ausgangsparameter für genau einen Bausteinaufruf ausgegeben und Anschließend gelöscht. (flankensensitiv).</p>
OUT	Valid	BOOL	TRUE: Der zu lesende Wert steht zur Verfügung.
	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.
	CommandAborted	BOOL	TRUE: Die Ausführung des Bausteins wurde abgebrochen.
INOUT	Axis	STRUCT	Diesem Parameter wird dem Achs-DB übergeben. Beispiel: Axis := DBname.Achsname
	Init	BOOL	Diesem Parameter wird ein Initialisierungsbit im Achs-DB, das von keinem anderen FB benutzt wird, übergeben ("Init.I0" .."Init.I63", siehe Bitfeld für die Initialisierungsfunktion). Der Baustein führt bei Init (=TRUE) seine Initialisierung durch und setzt anschließend das Bit zurück. Beispiel: Init := DBname.Achsname.Init.Ix mit Ix = I0 .. I63.

8.3 Signaldiagramme

Signaldiagramme mit Execute



Signaldiagramme mit Enable



8.4 Initialisierung

Nach jedem Neustart (Kalt- oder Warm) der SPS ist es notwendig, dass die Bibliotheksbausteine neu initialisiert werden, um die Lokaldaten der Bausteine in einen definierten Ursprungszustand (Initialwert) zu bringen.

Zu diesem Zweck ist im Achs-DB das Bitfeld "Init.Ix" definiert und jeder Baustein besitzt den Durchgangsparameter "Init". Die Bausteine durchlaufen einmalig ihre Initialisierungsroutine, wenn ihr Parameter "Init" gesetzt ist und setzen anschließend das übergebene Initialisierungsbit selbständig zurück. Die Funktion MC_Init_IFx übernimmt dabei unter anderem die Initialisierung des Achs-DB und setzt auch die Initialisierungsbits im Achs-DB zur Initialisierung der Bibliotheksbausteine.

Jedem, im Anwenderprogramm aufgerufenem Bibliotheksbaustein, muss deshalb am Durchgangsparameter "Init" ein Initialisierungsbit, aus dem Achs-DB übergeben werden.

Wichtig: Jedes Initialisierungsbit darf nur von einem Bibliotheksbaustein verwendet werden. Hiefür stehen insgesamt pro Achse 64 Initialisierungsbits (Achs-DB.Init.I0.. Achs-DB.Init.I63) zur Verfügung.

Durch die Initialisierung ist gewährleistet, dass keine Fehlfunktionen und gefährliche unerwartete Motorbewegungen durch alte bzw. nicht mehr gültige Daten verursacht werden.

Hinweis:

Durch die Verwendung der bibliotheksspezifischen GSD, werden automatisch bei der Profibus-Initialisierung, im Prozessdatenkanal das PZD5 und das PZD6 für die Senderichtung gemappt.

Dieses Mapping darf nicht verändert werden, da andernfalls die Funktion der Bibliothek nicht mehr gewährleistet ist !

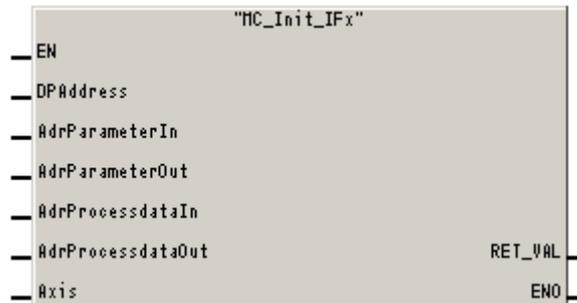
In Empfangsrichtung können das PZD5 und PZD6, wie in der Gratedokumentation beschrieben, gemappt werden.

8.4.1 MC_Init_IFx

Aufgabe:

Initialisierung einer Achse.

Aufruf:



Rufen Sie den Baustein MC_Init_IFx für jede Achse einmal, nach jedem Anlauf der CPU (OB100 und OB101), auf. Abhängig von Ihrer Programmstruktur können Sie den Baustein auch direkt im entsprechenden Anlauf-OB aufrufen.

Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	DPAddress	INT	Profibus Adresse der Achse
	AdrParameterIn	INT	Eingangsadresse des Parameterdatenkanal .
	AdrParameterOut	INT	Ausgangsadresse des Parameterdatenkanal .
	AdrProcessdataIn	INT	Eingangsadresse des Prozessdatenkanal .
	AdrProcessdataOut	INT	Ausgangsadresse des Prozessdatenkanal .
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
OUT	Ret_Val	INT	Fehlernummer (Wert <>0 = Fehler).

Funktionsweise:

Die übergebenen Adressen werden soweit möglich auf Plausibilität geprüft und in die Achsstruktur des Parameters Axis eingetragen. Außerdem werden die Initialisierungsbits gesetzt, um die Initialisierung der Bibliotheksbausteine vorzubereiten.

8.4.2 MC_Power_IFx

Aufgabe:

Ein-/Ausschalten des Motorstroms.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Enable	BOOL	FALSE: Ausschalten des Motorstroms. TRUE: Einschalten des Motorstroms.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Status	BOOL	Zeigt den Zustand des Motorstroms an. FALSE: Motorstrom ist ausgeschaltet. TRUE: Motorstrom ist eingeschaltet.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

Funktionsweise:

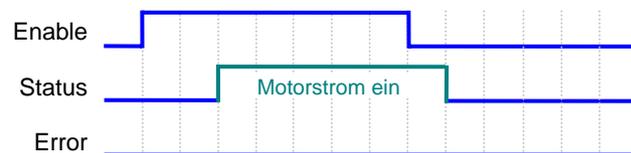
Mit TRUE am Eingang *Enable* wird der Motorstrom eingeschaltet. Sobald der Motorstrom eingeschaltet ist wird der Ausgang *Status* gesetzt.

Mit FALSE am Eingang *Enable* wird der Motorstrom ausgeschaltet. Sobald der Motorstromlos ist wird der Ausgang *Status* zurückgesetzt.

Treten bei der Ausführung Fehler auf, wird der Ausgang *Error* gesetzt.

Das Ausschalten des Motorstroms ist aus jedem Zustand heraus möglich. Ein zu diesem Zeitpunkt aktiver Bewegungsbaustein wird abgebrochen.

Signaldiagramm:



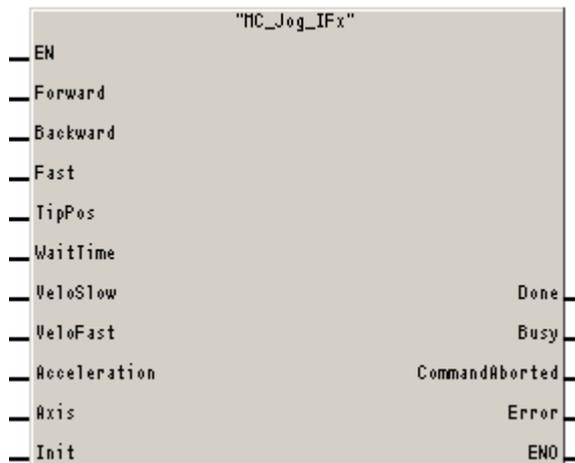
8.5 Manuellfahrt

8.5.1 MC_Jog_IFx

Aufgabe:

Die Manuellfahrt wird als "Klassische Manuellfahrt" ausgeführt, d.h. der Motor geht bei länger aktiven Eingängen *Forward* bzw. *Backward* in eine kontinuierliche Bewegung über.

Aufruf:



Parameterbeschreibung:

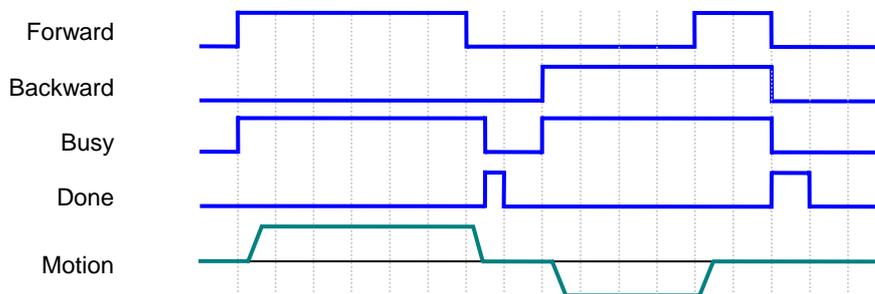
Par-typ	Parameter	Datentyp	Bedeutung
IN	Forward	BOOL	FALSE: Beenden der Bewegung. TRUE: Bewegung der Achse im Uhrzeigersinn.
	Backward	BOOL	FALSE: Beenden der Bewegung. TRUE: Bewegung der Achse gegen den Uhrzeigersinn.
	Fast	BOOL	Die Geschwindigkeit kann auch während der laufenden Fahrt zwischen zwei Werten umgeschaltet werden: FALSE: Geschwindigkeit <i>VeloSlow</i> ist ausgewählt. TRUE: Geschwindigkeit <i>VeloFast</i> ist ausgewählt.
	TipPos	INT	0: unendlich, d.h. der Motor geht sofort in eine kontinuierliche Bewegung über. >0: Wegstrecke [Inc] , die der Motor beim Start zuerst zurücklegt, bevor er nach der Verzögerungszeit (<i>WaitTime</i>) in eine kontinuierliche Bewegung übergeht. Wertebereich: IFS, IFA: 0..65535, Initialwert: 20. IFE: 0..65535, Initialwert: 2.
	WaitTime	INT	Verzögerungszeit [ms], die beginnt, nachdem der Motor eine definierte Wegstrecke (<i>TipPos</i>) zurückgelegt hat und nach deren Ablauf der Motor in eine kontinuierliche Bewegung übergeht. Wertebereich: 1..10000, Initialwert: 500.

IN	VeloSlow	INT	Geschwindigkeit [U/min] für die Fahrt wenn Fast = FALSE ist. Wertebereich: IFS: 1..3000, Initialwert: 300. IFE: 300..5000, Initialwert: 300. IFA: 1..6000, Initialwert: 300.
	VeloFast	INT	Geschwindigkeit [U/min] für die Fahrt wenn Fast = TRUE ist. Wertebereich: IFS: 1..3000, Initialwert: 1000. IFE: 300..5000, Initialwert: 1000. IFA: 1..6000, Initialwert: 1000.
	Acceleration	DINT	Wert für die Steilheit der Beschleunigungsrampe [(U/min/s)] Wertebereich: IFS: 1..765000, Initialwert: 2500. IFE: 1000..10000, Initialwert: 2500. IFA: 1..250000, Initialwert: 600..
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	CommandAborted	BOOL	TRUE: Die Ausführung des Bausteins wurde abgebrochen.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

Funktionsweise:

Mit TRUE an *Forward* oder *Backward* wird die Manuellfahrt gestartet. In Abhängigkeit des Parameters *Fast* wird entweder mit der langsamen (*VeloSlow*) oder mit der schnellen (*VeloFast*) Geschwindigkeit gefahren. Die Geschwindigkeit kann auch bei aktiver Manuellfahrt umgeschaltet werden. Durch die Parameter *TipPos* und *WaitTime* werden die Bedingungen für den Übergang vom Tipbetrieb in den kontinuierlichen Betrieb bestimmt. Sind *Forward* und *Backward* = FALSE, wird die Betriebsart beendet und *Done* gesetzt. Sind *Forward* und *Backward* = TRUE, bleibt die Betriebsart aktiv, die Manuellfahrt wird gestoppt und *Busy* bleibt gesetzt.

Signaldiagramm:



8.6 Referenzierung

Mit der Betriebsart Referenzierung wird ein absoluter Maßbezug der Motorposition zu einer definierten Achsposition hergestellt. Die Referenzierung kann durch die zwei nachfolgend beschriebenen Bausteine durchgeführt werden.

8.6.1 MC_SetPosition_IFx

Aufgabe:

Absolutes und Relatives Maßsetzen.

Aufruf:

Das Maßsetzen kann nur im Stillstand des Antriebs durchgeführt werden.



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
	Position	DINT	Maßsetzposition [Inc] Wertebereich: - 2147483648..2147483647, Initialwert: 0.
	Mode	BOOL	FALSE: aktuelle Motorposition auf <i>Position</i> setzen. TRUE: <i>Position</i> zur aktuellen Motorposition addieren.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

Funktionsweise:

Festlegen des Maßbezugs relativ zur aktuellen Motorposition. Das Maßsetzen bietet die Möglichkeit, einen Punkt auf der Achse als Bezugspunkt festzulegen, auf den sich nachfolgende Positionsangaben beziehen. Der Bezugspunkt für Sollpositionen wird durch das Maßsetzen auf die neue Maßsetzposition verschoben.

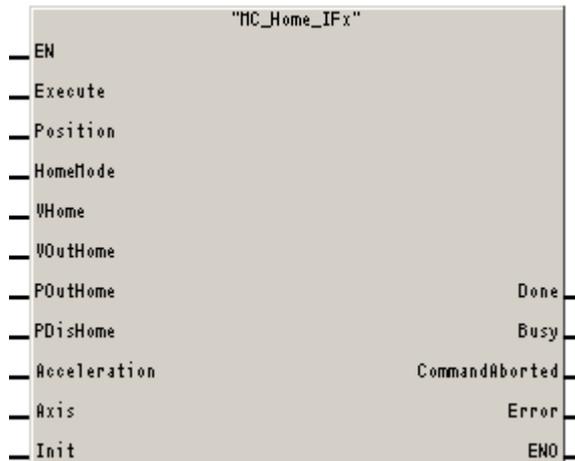
Das Maßsetzen kann eingesetzt werden, um kontinuierliche Absolut-Positionierungen ohne Überschreiten der Positioniergrenzen durchzuführen.

8.6.2 MC_Home_IFx

Aufgabe:

Durchführen der Referenzfahrt.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
	Position	DINT	Position wird zur aktuellen Motorposition nach erfolgreicher Referenzfahrt [Inc]. Wertebereich: - 2147483648..2147483647, Initialwert: 0.
	HomeMode	INT	1 = LIMP Referenzfahrt auf positiven Endschalter 2 = LIMN Referenzfahrt auf negativen Endschalter 3 = REFZ Referenzfahrt auf Referenzschalter in neg. Drehrichtung 4 = REFZ Referenzfahrt auf Referenzschalter in pos. Drehrichtung 5 = Referenzfahrt auf Indexpuls in neg. Drehrichtung 6 = Referenzfahrt auf Indexpuls in pos. Drehrichtung 7 = Blockfahrt in neg. Drehrichtung 8 = Blockfahrt in pos. Drehrichtung 9 = Maßsetzen
	VHome	INT	Geschwindigkeit für die Suche des End- bzw. Referenzschalters [U/min]. Nach dem Erkennen der Schaltkante stoppt der Antrieb. IFS: 1..3000, Initialwert: 60. IFE: 300..5000, Initialwert: 1000. IFA: 1..6000, Initialwert: 60.
	VOutHome	INT	Geschwindigkeit für das Freifahren zurück zur Schaltkante [U/min]. Die maximale Fahrstrecke zum Suchen der Schaltkante kann mit dem Parameter <i>POutHome</i> begrenzt werden. IFS: 1..3000, Initialwert: 6. IFE: 300..5000, Initialwert: 500. IFA: 1..6000, Initialwert: 6.
	POutHome	DINT	Ausfahrweg [Inc], d.h. maximale Fahrstrecke zum Suchen der Schaltkante. Wenn innerhalb dieser Strecke die Schaltkante nicht gefunden wird, erfolgt ein Abbruch der Referenzfahrt mit Fehler. Wertebereich: 1..2147483647, Initialwert: 200000.

IN	PDisHome	DINT	Abstand von der Schaltkante zum Referenzpunkt [Inc]. Der Antrieb fährt nach der Fahrt zurück zur Schaltkante weiter, bis der Abstand erreicht ist. Wertebereich: 1..2147483647, Initialwert: 200.
	Acceleration	DINT	Wert für die Steilheit der Beschleunigungsrampe [(U/min/s)] Wertebereich: IFS: 1..765000, Initialwert: 2500. IFE: 1000..10000, Initialwert: 2500. IFA: 1..250000, Initialwert: 600.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	CommandAborted	BOOL	TRUE: Die Ausführung des Bausteins wurde abgebrochen.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

Funktionsweise:

Mit der Referenzfahrt wird eine definierte Position auf der Achse angefahren. Die definierte Position ist durch einen mechanischen Schalter festgelegt: Endschalter, Referenzschalter.

Es gibt 6 Standard-Referenzfahrten:

1. Fahrt auf positiven Endschalter LIMP.
2. Fahrt auf negativen Endschalter LIMN.
3. Suche des Referenzschalter REF in positiver Drehrichtung.
4. Suche des Referenzschalter REF in negativer Drehrichtung.
5. Block (IFE) bzw. Indexpuls (IFS mit Indexpuls und IFA) in positiver Drehrichtung.
6. Block (IFE) bzw. Indexpuls (IFS mit Indexpuls und IFA) in negativer Drehrichtung.

Eine Referenzfahrt kann zusätzlich mit oder ohne Indexpuls durchgeführt werden:

- Referenzfahrt ohne Indexpuls
Fahrt von Schalterkante auf einen parametrierbaren Abstand zur Schalterkante.
- Referenzfahrt mit Indexpuls
Fahrt von der Schalterkante auf den nächsten Indexpuls des Gebers.

Für die Referenzfahrt sind Such- (*VHome*) und Freifahrtgeschwindigkeiten (*VOutHome*) sowie Sicherheitsabstand (*PDisHome*) und Freifahrtweg (*POutHome*) einstellbar. Eine Referenzfahrt muss vollständig durchgeführt werden, damit der neue Referenzpunkt gültig ist. Wenn eine Referenzfahrt abgebrochen wurde, muss sie neu gestartet werden. Der Motor bewegt sich in Abhängigkeit dieser Parameter, bis er sein Ziel erreicht oder die Betriebsart durch Ausführung eines anderen Bausteins (z.B. MC_Stop) abgebrochen wird. Eine Übernahme geänderter Eingangsparameter durch eine steigende Flanke an *Execute*, während einer laufenden Referenzierung, ist nicht erlaubt und führt zu einem Fehler. Nach erfolgreichem Abschluss der Referenzfahrt wird automatisch ein Maßbezug hergestellt. Dadurch wird die erreichte Position zur absoluten Referenzposition und auf den Wert von *Position* gesetzt.

8.7 Punkt-zu-Punkt Betrieb

In der Punkt-zu-Punkt Betriebsart wird der Motor mit einem Bewegungsbaustein von einem Punkt A auf einen Punkt B positioniert.

8.7.1 MC_MoveAbsolute_IFx

Aufgabe:

Starten und Überwachen der Betriebsart Punkt-zu-Punkt Positionierung mit absoluter Zielposition.



Aufruf:

Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
	Position	DINT	Wert für die absolute Zielposition [Inc]. Wertebereich: - 2147483648..2147483647, Initialwert: 0.
	Velocity	INT	Wert für die Sollgeschwindigkeit der Bewegung [U/min]. IFS: 0..3000, Initialwert: 60. IFE: 0..5000, Initialwert: 1000. IFA: 0..n_maxDrv ¹ , Initialwert: 60.
	Acceleration	DINT	Wert für die Steilheit der Beschleunigungsrampe [(U/min/s)] Wertebereich: IFS: 1..765000, Initialwert: 2500. IFE: 1000..10000, Initialwert: 2500. IFA: 1..250000, Initialwert: 600.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	CommandAborted	BOOL	TRUE: Die Ausführung des Bausteins wurde abgebrochen.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

¹ Der Maximalwert der Sollgeschwindigkeit ist beim IFA abhängig vom Typ des Motors und kann über die Gräte-dokumentation oder mit dem Baustein MC_ReadParameter_IFx über das Objekt (300Fh:12h) ermittelt werden.

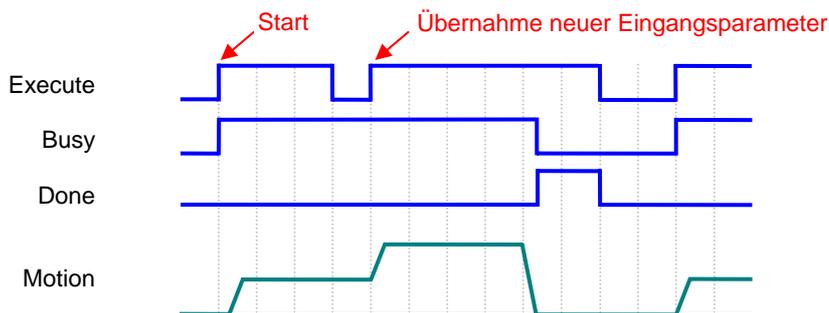
Funktionsweise:

Positionierung, mit absolutem Bezug auf den Nullpunkt der Achse, auf die Zielposition *Position* mit der Geschwindigkeit *Velocity*. Der Motor bewegt sich in Abhängigkeit dieser Parameter, bis er sein Ziel erreicht, ein neuer Sollwert übergeben oder die Betriebsart durch Ausführung eines anderen Bausteins (z.B. MC_Stop) abgebrochen wird.

Hinweis:

Vor einer Absolut-Positionierung muss der Referenzpunkt durch eine Referenzierung definiert werden.

Signaldiagramm:



8.7.2 MC_MoveAdditive_IFx

Aufgabe:

Starten und Überwachen der Betriebsart Punkt-zu-Punkt Positionierung mit Zielposition relativ zur aktuellen Zielposition.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
	Distance	DINT	Wert für die Wegstrecke die zur aktuellen Zielposition addiert wird und somit die neue Zielposition bestimmt [Inc]. Wertebereich: - 2147483648..2147483647, Initialwert: 0.
	Velocity	INT	Wert für die Sollgeschwindigkeit der Bewegung [U/min]. IFS: 0..3000, Initialwert: 60. IFE: 0..5000, Initialwert: 1000. IFA: 0..n_maxDrv ¹ , Initialwert: 60.
	Acceleration	DINT	Wert für die Steilheit der Beschleunigungsrampe [(U/min/s)] Wertebereich: IFS: 1..765000, Initialwert: 2500. IFE: 1000..10000, Initialwert: 2500. IFA: 1..250000, Initialwert: 600.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	CommandAborted	BOOL	TRUE: Die Ausführung des Bausteins wurde abgebrochen.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

¹ Der Maximalwert der Sollgeschwindigkeit ist beim IFA abhängig vom Typ des Motors und kann über die Gräterdokumentation oder mit dem Baustein MC_ReadParameter_IFx über das Objekt (300Fh:12h) ermittelt werden.

Funktionsweise:

Positionierung um die Wegstrecke *Distance* bezogen auf die aktuelle Zielposition mit der Geschwindigkeit *Velocity*. Der Motor bewegt sich in Abhängigkeit dieser Parameter, bis er sein Ziel erreicht, ein neuer Sollwert übergeben oder die Betriebsart durch Ausführung eines anderen Bausteins (z.B. MC_Stop) abgebrochen wird.

8.8 Geschwindigkeitsbetrieb

8.8.1 MC_MoveVelocity_IFx

Aufgabe:

Starten und Überwachen des Geschwindigkeitsbetrieb.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
	Velocity	INT	Wert für die Sollgeschwindigkeit der Bewegung [U/min]. Wertebereich: IFS: -3000..3000, Initialwert: 0. IFE: -5000..-300, 300..5000 ¹ , Initialwert: 0. IFA: -n_maxDrv..n_maxDrv ² , Initialwert: 0.
	Acceleration	DINT	Wert für die Steilheit der Beschleunigungsrampe [(U/min/s)] Wertebereich: IFS: 1..765000, Initialwert: 2500. IFE: 1000..10000, Initialwert: 2500. IFA: 1..250000, Initialwert: 600.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	InVelocity	BOOL	TRUE: Sollgeschwindigkeit erreicht.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	CommandAborted	BOOL	TRUE: Die Ausführung des Bausteins wurde abgebrochen.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

¹ Sollgeschwindigkeiten von -300 bis 300 U/min sind beim IFE nicht erlaubt (ausgenommen 0).

² Der Maximalwert der Sollgeschwindigkeit ist beim IFA abhängig vom Typ des Motors und kann über die Gerätedokumentation oder mit dem Baustein MC_ReadParameter_IFx über das Objekt (300Fh:12h) ermittelt werden.

Funktionsweise:

In der Betriebsart Geschwindigkeitsbetrieb wird dem Motor eine Sollgeschwindigkeit über den Parameter *Velocity* vorgegeben und eine Bewegung ohne Zielposition gestartet. Der Motor bewegt sich in Abhängigkeit dieser Sollwertvorgabe, bis ein neuer Sollwert übergeben oder die Betriebsart durch Ausführung eines anderen Bausteins (z.B. MC_Stop) abgebrochen wird.

8.9 Elektronisches Getriebe

In der Betriebsart Elektronisches Getriebe errechnet die Positioniersteuerung aus einer Positionsvorgabe und einem einstellbaren Getriebefaktor einen neuen Positionssollwert für die Motorbewegung. Die Betriebsart wird eingesetzt, wenn einer oder mehrere Motoren dem Führungssignal einer NC-Steuerung oder eines Encoders positionsgerecht folgen sollen.

Für die Betriebsart Elektronisches Getriebe müssen die Führungssignale auf dem Optionssteckplatz (CN2) eingespeist werden. Werden Führungspulse eingespeist, verrechnet die Positioniersteuerung diese mit dem Getriebefaktor und positioniert den Motor auf die neue Sollposition.

Positionswerte werden in internen Inkrementen angegeben. Einer Änderung der Werte folgt die Positioniersteuerung sofort. Die Betriebsart Elektronisches Getriebe wird nicht durch die Bereichsgrenzen der Positionierung begrenzt.

Der Getriebefaktor ist das Verhältnis zwischen extern eingespeisten Führungspulsen zu den Ausgangspulsen für die Motorbewegung. Der Getriebefaktor wird mit Zähler und Nenner festgelegt. Ein negativer Zähler kehrt die Drehrichtung um. Der resultierende Positionierweg ist abhängig von der aktuellen Motoraufösung.

Hinweise

Diese Funktion steht nur im Antrieb IFA zur Verfügung!

Synchronisation: vor der Aufnahme der Betriebsart Elektronisches Getriebe besteht keine Synchronisation zwischen Führungspulsen und Motor.

Bei einer Ausgleichsbewegung (*SyncMode* = TRUE) wird die Motorbewegung nur durch den maximalen Strom (Geräteparameter Settings.I_max, siehe Handbuch) und die maximale Drehzahl des Antriebs begrenzt.

Sobald die Getriebebearbeitung aktiviert wird, darf die Regelabweichung, die durch die aufgelaufenen Pulse entstanden ist, nicht größer als der Schleppfehlergrenzwert (Geräteparameter "Settings.p_maxDif2", siehe Handbuch) sein. Der Antrieb meldet sonst einen Schleppfehler.

Bei der Sofortsynchronisation (*SyncMode* = FALSE) folgt der Motor den Führungspulsen, ab dem Zeitpunkt an dem die Getriebebearbeitung im Antrieb aktiv wird.

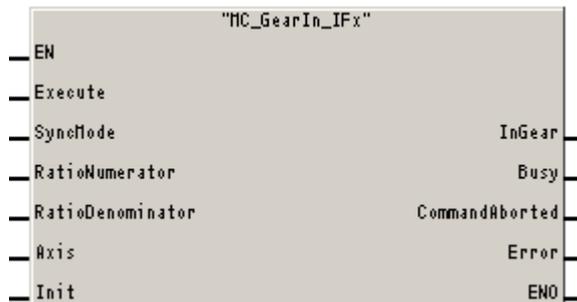
Richtungsfreigabe: vor Aktivieren der Getriebebearbeitung kann die Richtung einer Ausgleichsbewegung vorgegeben werden (Geräteparameter "Gear.dirEnGear", siehe Handbuch). Zur korrekten Freigabe einer Richtung muss die Richtungsinvertierung berücksichtigt werden, die über den Geräteparameter "Motion.invertDir" (siehe Handbuch) ermittelt werden kann.

8.9.1 MC_GearIn_IFx

Aufgabe:

Starten und Überwachen der Betriebsart Elektronisches Getriebe mit Getriebefaktor.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
	SyncMode	BOOL	FALSE: Sofort-Synchronisation. Die Positioniersteuerung folgt Führungspulsen ab dem Zeitpunkt, an dem die Getriebearbeitung aktiviert wird. Führungspulse, die vor Start der Betriebsart aufgetreten sind, werden nicht berücksichtigt. TRUE: Synchronisation mit Ausgleichsbewegung. Mit Aktivierung der Getriebearbeitung versucht der Motor die bis dahin aufgelaufenen Führungspulse nachzuholen.
	RatioNumerator	INT	Zähler des Getriebefaktors. Wertebereich: -32768 .. 32767, Initialwert: 1.
	RatioDenominator	INT	Nenner des Getriebefaktors. Wertebereich: 1 .. 32767, Initialwert: 1.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	InGear	BOOL	FALSE: Das elektronische Getriebe ist deaktiviert. TRUE: Das elektronische Getriebe ist aktiviert.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	CommandAborted	BOOL	TRUE: Die Ausführung des Bausteins wurde abgebrochen.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

Funktionsweise:

In der Betriebsart Elektronisches Getriebe werden dem Motor durch Führungsimpulse am Encodereingang in Verbindung mit dem Getriebefaktor (Parameter *Numerator* und *Denominator*) fortlaufend berechnete Positionssollwerte vorgeben. Der Motor bewegt sich in Abhängigkeit dieser Sollwertvorgabe, bis ein neuer Getriebefaktor übergeben wird. Die Betriebsart wird durch Ausführung des Bausteins MC_GearOut_IFx beendet oder durch Ausführen eines anderen Bausteins (z.B. MC_Stop) abgebrochen.

8.9.2 MC_GearOut_IFx

Aufgabe:

Ausschalten der Betriebsart Elektronisches Getriebe.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	CommandAborted	BOOL	TRUE: Die Ausführung des Bausteins wurde abgebrochen.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

Funktionsweise:

Bei aktiver Betriebsart Elektronisches Getriebe wird der Motor direkt vom Getriebemaster abgekuppelt und mit einer Momentenrampe bis zum Stillstand abgebremst. Der aktive Baustein MC_GearIn_IFx wird dadurch abgebrochen und liefert CommandAborted = TRUE.

8.10 Stoppen

8.10.1 MC_Stop_IFx

Aufgabe:

Stoppen des Antriebs mit Momentenrampe.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

Funktionsweise:

Jede Betriebsart kann durch Stoppen des Antriebs abgebrochen werden. Dies erzeugt keinen Fehler. Der abgebrochene Bewegungsbaustein beendet die Ausführung mit *CommandAborted* = TRUE und der Antrieb wechselt in den Status "Stopping". Dieser Status wird erst verlassen wenn der Antrieb steht und der Eingang *Execute* am Baustein zurückgesetzt wird. Dann wechselt der Status in "Standstill" und es können wieder Bewegungsbausteine gestartet werden.

Hinweis:

Diese Funktion kann durch andere Bewegungsbausteine nicht abgebrochen werden. Solange *Execute* = TRUE ist, kann kein anderer Bewegungsbaustein gestartet werden. Der Antrieb bleibt auch nach dem Stillstand blockiert.

Der Baustein bremst den Motor mit einer Momentenrampe. Der Parameter *LIM_I_maxHalt* (siehe Handbuch) spezifiziert den Strom für die Momentenrampe. Nach Antriebsstillstand erfolgt ein interner Positionsabgleich, die Lageregelung wird aktiviert und der Motor wird bei aktiver Endstufe gehalten.

8.11 Schnelle Positionserfassung

Über 2 parametrierbare Kanäle lässt sich die Motorposition mit einer Genauigkeit von 10µs Zeitverzögerung erfassen. Zur Erfassung der auslösenden Signale stehen die Signaleingänge CAP1 und CPA2 zur Verfügung.

Hinweise

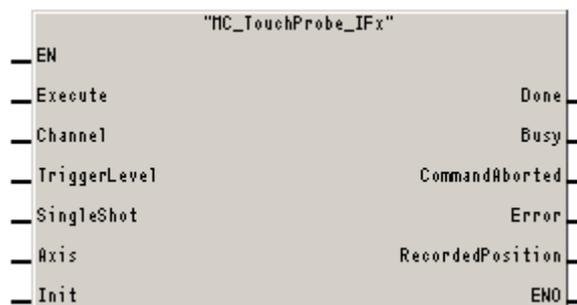
Diese Funktion steht nur in den Antrieben IFA und IFS zur Verfügung!

8.11.1 MC_TouchProbe_IFx

Aufgabe:

Einstellen, Starten und Überwachen der schnellen Positionserfassung.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
	Channel	UINT	Kanalnummer: Auswahl des Kanals, auf den sich die übrigen Parameter beziehen (1 = CAP1, 2 = CAP2). Wertebereich: 1 .. 2, Initialwert: 1.
	TriggerLevel	BOOL	Auslösende Signalfanke. FALSE: fallende Flanke TRUE: steigende Flanke
	SingleShot	BOOL	FALSE: bei wiederholtem Eintreten des auslösenden Ereignisses wird die erfasste Position durch die neue Position überschrieben. TRUE: die Positionserfassung wird nach Eintreten des auslösenden Ereignisses ausgeschaltet, sodass die erfasste Position nicht überschrieben werden kann. Initialwert: TRUE.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].

OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	CommandAborted	BOOL	TRUE: Die Ausführung des Bausteins wurde abgebrochen.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.
	RecordedPosition	BOOL	Bei Eintreten des auslösenden Ereignisses erfasste Motorposition. Wertebereich: -2147483648 .. 2147483647, Initialwert: 0 [Inc].

Funktionsweise:

Die schnelle Positionserfassung dient zur Erfassung der aktuellen Motorposition zum Zeitpunkt des Eintreffens eines digitalen 24V-Signals an einem der beiden Capture-Eingänge (IO2 = CAP1, IO3 = CAP2).

Dabei kann die Motorposition bei steigender oder fallender Flanke am entsprechenden Eingang (*Channel*) erfasst werden.

Des Weiteren kann mit dem Parameter *SingleShot* eingestellt werden, ob die Positionserfassung einmalig oder kontinuierlich durchgeführt wird.

Bei einmaliger Positionserfassung beendet sich der Baustein nach Auftreten der eingestellten Flanke (*TriggerLevel*) mit *Done* = TRUE und liefert die erfasste Position (*RecordedPosition*).

Bei kontinuierlicher Positionserfassung liefert der Baustein für jede aufgetretene Flanke die dann neu erfasste Position (*RecordedPosition*), wobei die alte Position überschrieben wird. Der Baustein beendet sich nicht von selbst, sondern kann nur mit MC_AbortTrigger_IFx abgebrochen werden.

8.11.2 MC_AbortTrigger_IFx

Aufgabe:

Abbrechen einer aktiven Positionserfassung.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
	Channel	INT	1: Positionserfassung auf Kanal 1 (CAP1) abbrechen. 2: Positionserfassung auf Kanal 2 (CAP2) abbrechen. Wertebereich: 1 .. 2, Initialwert: 1.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

Funktionsweise:

Die aktive Positionserfassung wird für den entsprechenden Kanal *Channel* deaktiviert. Der Baustein MC_TouchProbe_IFx liefert für diesen Kanal *CommandAborted* = TRUE.

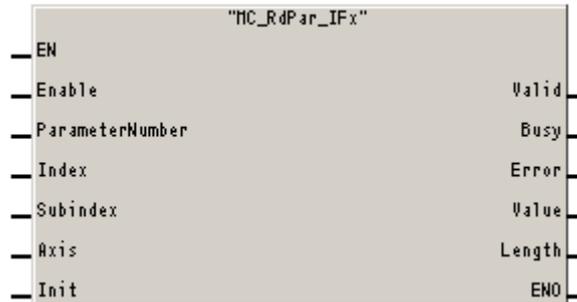
8.12 Parameter lesen

8.12.1 MC_ReadParameter_IFx

Aufgabe:

Lesen eines Objekts aus der [Geräteparameterliste](#).

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Enable	BOOL	FALSE: Ausführung des Baustein beenden. TRUE: Ausführung des Baustein starten.
	ParameterNumber	INT	0: Parameter wird durch <i>Index</i> ausgewählt. 1: aktuelle Sollposition des Profilgenerators [Inc] 10: aktuelle Istgeschwindigkeit [U/min] 11: aktuelle Sollgeschwindigkeit [U/min] Andere Nummern werden nicht unterstützt. Wertebereich: 0..32767, Initialwert: 0.
	Index	INT	Index des zu lesenden Objektes; die Objekte sind mit ihrem Index und Subindex im Handbuch gelistet. Nur gültig bei <i>ParameterNumber</i> = 0. Wertebereich: 0..32767, Initialwert: 0.
	Subindex	INT	Subindex des zu lesenden Objektes; die Objekte sind mit ihrem Index und Subindex im Handbuch gelistet. Nur gültig bei <i>ParameterNumber</i> = 0. Wertebereich: 0..32767, Initialwert: 0.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: Der zu lesende Wert steht zur Verfügung.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.
	Value	DINT	Wert des zu lesenden Parameters. Wertebereich: -2147483648..2147483647, Initialwert: 0.
	Length	INT	Länge des zu lesenden Parameters [Byte]. Wertebereich: 1..4, Initialwert: 0.

8.12.2 MC_ReadStatus_IFx

Aufgabe:

Lesen des aktuellen Zustands des Antriebs.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Enable	BOOL	FALSE: Ausführung des Baustein beenden. TRUE: Ausführung des Baustein starten.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: der gelesene Status ist gültig. FALSE: der Status ist (noch) nicht gültig.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.
	Errorstop	BOOL	TRUE: Die Achse befindet sich im Fehlerzustand.
	Disabled	BOOL	TRUE: Der Motorstrom ist ausgeschaltet.
	Stopping	BOOL	TRUE: Die Achse wurde gestoppt und der Baustein MC_Stop_IFx blockiert noch die Achse.
	Referenced	BOOL	TRUE: Der Antrieb ist referenziert.
	StandStill	BOOL	TRUE: Der Antrieb steht.
	DiscreteMotion	BOOL	TRUE: Der Antrieb befindet sich in einer PTP-Betriebsart.

OUT	ContinuousMotion	BOOL	TRUE: Der Antrieb befindet sich in einer Betriebsart ohne definierte Zielposition (MC_Jog_IFx, MC_MoveVelocity_IFx).
	SynchronizedMotion	BOOL	Der Antrieb befindet sich im Elektronischen Getriebe (nur IFA).
	Homing	BOOL	TRUE: Der Antrieb befindet sich in der Betriebsart Referenzierung.
	ConstantVelocity	BOOL	TRUE: Der Antrieb fährt mit konstanter Geschwindigkeit.
	Accelerating	BOOL	TRUE: Der Antrieb beschleunigt.
	Decelerating	BOOL	TRUE: Der Antrieb verzögert.

Funktionsweise:

Es werden die aktuellen Statusinformationen des Antriebs gelesen und ausgegeben. Diese sind nur in Verbindung mit dem Parameter *Valid* gültig.

Hinweis:

Der Antrieb befindet sich zu jedem Zeitpunkt in genau einem der Zustände *StandStill*, *Homing*, *DiscreteMotion*, *ContinuousMotion*, *SynchronizedMotion*, *Stopping*, *Disabled* oder *Errorstop*. Der gleichnamige Ausgang des Bausteins ist dann TRUE.

Gleiches gilt für die Bewegungszustände *ConstantVelocity*, *Accelerating* und *Decelerating*.

8.12.3 MC_ReadActualPosition_IFx

Aufgabe:

Lesen der aktuellen Istposition des Motors in Inkrementen.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Enable	BOOL	FALSE: Ausführung des Baustein beenden. TRUE: Ausführung des Baustein starten.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: Der zu lesende Wert steht zur Verfügung.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.
	Position	DINT	Aktuelle Istposition des Motors [Inc]. Wertebereich: -2147483648..2147483647 [Inc], Initialwert: 0 [Inc].

8.12.4 MC_ReadActualVelocity_IFx

Aufgabe:

Lesen der aktuellen Drehzahl des Motors in U/min.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Enable	BOOL	FALSE: Ausführung des Baustein beenden. TRUE: Ausführung des Baustein starten.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: Der zu lesende Wert steht zur Verfügung.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.
	Velocity	DINT	Aktuelle Drehzahl des Motors [U/min]. Wertebereich: -2147483648..2147483647, Initialwert: 0.

8.12.5 MC_ReadRefPosition_IFx

Aufgabe:

Lesen der aktuellen Position des Bewegungsprofilgenerators in Inkrementen.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Enable	BOOL	FALSE: Ausführung des Baustein beenden. TRUE: Ausführung des Baustein starten.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: Der zu lesende Wert steht zur Verfügung.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.
	Position	DINT	Aktuelle Istposition des Bewegungsprofilgenerators [Inc]. Wertebereich: -2147483648..2147483647, Initialwert: 0.

8.12.6 MC_ReadRefVelocity_IFx

Aufgabe:

Lesen der aktuellen Drehzahl des Bewegungsprofilgenerators in U/min.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Enable	BOOL	FALSE: Ausführung des Baustein beenden. TRUE: Ausführung des Baustein starten.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: Der zu lesende Wert steht zur Verfügung.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.
	Velocity	INT	Aktuelle Drehzahl des Bewegungsprofilgenerators [U/min]. Wertebereich: -32768..32767, Initialwert: 0.

8.12.7 MC_ReadActualMasterPosition_IFx

Aufgabe:

Lesen der aktuellen Position des externen Encoders in Inkrementen.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Enable	BOOL	FALSE: Ausführung des Baustein beenden. TRUE: Ausführung des Baustein starten.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: Der zu lesende Wert steht zur Verfügung.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.
	Position	DINT	Aktuelle Position des externen Encoders [Inc]. Wertebereich: -2147483648..2147483647, Initialwert: 0.

8.12.8 MC_ReadActualMasterVelocity_IFx

Aufgabe:

Lesen der aktuellen Geschwindigkeit des externen Encoders in Inkrementen pro Sekunde.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Enable	BOOL	FALSE: Ausführung des Baustein beenden. TRUE: Ausführung des Baustein starten.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: Der zu lesende Wert steht zur Verfügung.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.
	Velocity	DINT	Aktuelle Geschwindigkeit des externen Encoders [Inc/s]. Wertebereich: -2147483648..2147483647, Initialwert: 0.

8.12.9 MC_UploadParameter_IFx

Aufgabe:

Lesen der vom Anwender änderbaren Parameter und ablegen in der [Parameterliste](#).

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
	Data	ANY	Datenstruktur zur Ablage der gelesenen Daten.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.
	Errorinfo	DWORD	Zusätzliche Fehlerinformation, Adresse des Fehlers. High Word: DB-Nummer, Low Word: Anfangsadresse im DB Wertebereich: 16#0..16#FFFFFFF, Initialwert: 16#0.

Funktionsweise:

Die Liste der veränderbaren Parameter ist in der Bibliothek im UDT13 durch die Struktur Data festgelegt. Für den Upload der Parameter muss eine Struktur vom Typ UDT13 in einem globalen DB angelegt und dem Eingang *Data* am Baustein übergeben werden. Die gelesenen Parameter werden in dieser Struktur abgelegt.

Hinweis:

Durch die Bausteine MC_UploadParameter_IFx und [MC_DownloadParameter_IFx](#) kann ein defektes Gerät ohne spezielles Tool zum Parametrieren des Gerätes ausgetauscht werden.

Durch die Ausführung des Baustein werden alle Anwenderparameter vom Antrieb gelesen und in der, am Eingang *Data* übergebenen, Datenstruktur abgelegt.

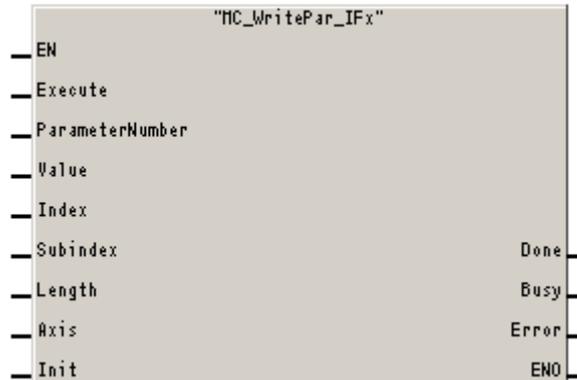
8.13 Parameter schreiben

8.13.1 MC_WriteParameter_IFx

Aufgabe:

Schreiben eines Objekts aus der [Geräteparameterliste](#).

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
	ParameterNumber	INT	0: Parameter wird durch <i>Index</i> ausgewählt. Andere Nummern werden nicht unterstützt. Wertebereich: 0..32767, Initialwert: 0.
	Value	DINT	Zu schreibender Wert des Parameters Wertebereich: -2147483648..2147483647, Initialwert: 0.
	Index	INT	Index des zu lesenden Objektes; die Objekte sind mit ihrem Index und Subindex im Handbuch gelistet. Nur gültig bei <i>ParameterNumber</i> = 0. Wertebereich: 0..32767, Initialwert: 0.
	Subindex	INT	Subindex des zu lesenden Objektes; die Objekte sind mit ihrem Index und Subindex im Handbuch gelistet. Nur gültig bei <i>ParameterNumber</i> = 0. Wertebereich: 0..32767, Initialwert: 0.
	Length	INT	Länge des zu schreibenden Parameters [Byte]. Wertebereich: 1..4, Initialwert: 0.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

8.13.2 MC_ResetParameters_IFx

Aufgabe:

Zurücksetzen der [Anwenderparameter](#) auf Werkseinstellung.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

Funktionsweise:

Alle [Anwenderparameter](#) werden auf ihre Default-Werte zurückgesetzt und im EEPROM gespeichert. Die Werte werden erst nach Aus- und anschliessendem Wiedereinschalten übernommen.

Hinweis:

Alle vom Anwender eingestellten Parameterwerte gehen verloren, wenn nicht vorher mit der Inbetriebnahmesoftware eine Sicherung durchgeführt wurde.
Die Herstellung der Default-Einstellungen ist nur im Stillstand des Antriebs möglich!

8.13.3 MC_StoreParameters_IFx

Aufgabe:

Sichern aller [Anwenderparameter](#) in einem remanenten Speicher (EEPROM).

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

Funktionsweise:

Alle [Anwenderparameter](#) werden und im nicht flüchtigen Speicher (EEPROM) gespeichert.

Hinweis:

Die Sicherung der [Anwenderparameter](#) ist nur im Stillstand des Antriebs möglich!

8.13.4 MC_DownloadParameter_IFx

Aufgabe:

Schreiben der vom Anwender änderbaren Parameter zum Antrieb aus der [Parameterliste](#).

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
	Data	ANY	Datenstruktur der zu schreibenden Daten.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.
	Errorinfo	DWORD	Zusätzliche Fehlerinformation, Adresse des Fehlers. High Word: DB-Nummer, Low Word: Anfangsadresse im DB Wertebereich: 16#0..16#FFFFFFF, Initialwert: 16#0.

Funktionsweise:

Die Liste der veränderbaren Parameter ist in der Bibliothek im UDT13 durch die Struktur Data festgelegt. Für den Download der Parameter muss eine Struktur vom Typ UDT13 in einem globalen DB angelegt und dem Eingang *Data* am Baustein übergeben werden. Die zu schreibenden Parameter werden aus dieser Struktur gelesen.

Hinweis:

Es wird empfohlen, vor dem Ausführen dieses Bausteins, alle Parameter mit dem Baustein [MC UploadParameters_IFx](#) zu lesen, da die Initialwerte der Parameter nicht zwingend mit den Initialwerten des Antriebs übereinstimmen müssen. Nach dem Lesen können die Parameter im DB geändert und der Download ausgeführt werden.

8.14 Ein-/Ausgänge

Neben dem Prozessabbild, in dem die digitalen Ein- und Ausgänge des Zielsystems abgebildet sind, stehen Bausteine zur Verfügung, mit denen auf digitale Ein-/Ausgänge jedes Antriebs im System zugegriffen werden kann. Die 24V-Schnittstelle des Antriebs stellt 4 programmierbare Eingänge bzw. Ausgänge zur Verfügung, die mit Funktionen belegt werden können.

8.14.1 MC_ReadDigitalInput_IFx

Aufgabe:

Lesen des aktuellen Eingangsabbilds des Antriebs.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Enable	BOOL	FALSE: Ausführung des Baustein beenden. TRUE: Ausführung des Baustein starten.
	InputNumber	INT	Nummer des Eingangs, der gelesen werden soll: 0: IO0 1: IO1 2: IO2 3: IO3 Wertebereich: 0..3, Initialwert: 0.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: Der zu lesende Wert steht zur Verfügung.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.
	Value	BOOL	TRUE: gelesener Eingang (InputNumber) hat 24V-Pegel. FALSE: gelesener Eingang (InputNumber) hat 0V-Pegel.
	Inputs	WORD	Gesamtes Eingangsabbild (unabhängig von InputNumber): IO0 = Bit 0, IO1 = Bit 1, IO2 = Bit 2, IO3 = Bit 3. Wertebereich: 00h..0Fh, Initialwert: 00h.

8.14.2 MC_ReadDigitalOutput_IFx

Aufgabe:

Lesen des aktuellen Ausgangsabbilds des Antriebs.

Aufruf:



Parameterbeschreibung:

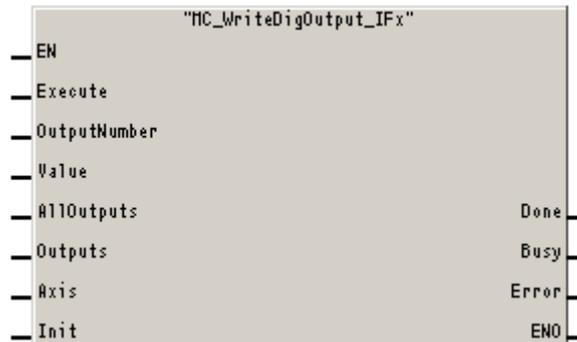
Par-typ	Parameter	Datentyp	Bedeutung
IN	Enable	BOOL	FALSE: Ausführung des Baustein beenden. TRUE: Ausführung des Baustein starten.
	OutputNumber	INT	Nummer des Ausgangs, der gelesen werden soll: 0: IO0 1: IO1 2: IO2 3: IO3 Wertebereich: 0..3, Initialwert: 0.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: Der zu lesende Wert steht zur Verfügung.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.
	Value	BOOL	TRUE: gelesener Ausgang (OutputNumber) hat 24V-Pegel. FALSE: gelesener Ausgang (OutputNumber) hat 0V-Pegel.
	Outputs	WORD	Gesamtes Ausgangsabbild (unabhängig von OutputNumber): IO0 = Bit 0, IO1 = Bit 1, IO2 = Bit 2, IO3 = Bit 3 Wertebereich: 00h..0Fh, Initialwert: 00h.

8.14.3 MC_WriteDigitalOutput_IFx

Aufgabe:

Schreiben des Ausgangsabbilds des Antriebs.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
	OutputNumber	INT	Nummer des Ausgangs, der geschrieben werden soll: 0: IO 0 1: IO 1 2: IO 2 3: IO 3 Wertebereich: 0..3, Initialwert: 0.
	Value	BOOL	FALSE: Ausgang zurücksetzen (0V-Pegel) TRUE: Ausgang setzen (24V-Pegel)
	AllOutputs	BOOL	FALSE: Einen Ausgang schreiben (OutputNumber, Value gültig) TRUE: Alle Ausgänge schreiben (Outputs gültig)
	Outputs	WORD	Gesamtes Ausgangsabbild (unabhängig von OutputNumber): IO0 = Bit 0, IO1 = Bit 1, IO2 = Bit 2, IO3 = Bit 3 Wertebereich: 00h..0Fh, Initialwert: 00h.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

Hinweis:

Dieser Baustein kann nur für IOs verwendet werden, welche als Ausgänge konfiguriert wurden (siehe Handbuch bzw. Beschreibung im nächsten Abschnitt).

8.14.4 MC_ConfigureIO_IFx

Aufgabe:

Konfigurieren und Freischalten der Ein-/Ausgänge.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
	IONumber	INT	Nummer des IOs, der konfiguriert werden soll: 0: IO 0 1: IO 1 2: IO 2 3: IO 3 Wertebereich: 0..3, Initialwert: 0.
	Configuration	INT	Konfiguration: 0: Eingang; frei nutzbar 1: LIMP; Eingang für positiven Endschalter (nur für IO0 möglich) 2: LIMN; Eingang für negativen Endschalter (nur für IO1 möglich) 3: STOP; Stop-Eingang 4: REF; Eingang für Referenzschalter 5: Eingang programmierbar (siehe MC_ControlIO_IFx) 128: Ausgang; frei nutzbar 129: Ausgang; Indexpuls (nur für IO0 möglich) 130: Ausgang programmierbar (siehe MC_ControlIO_IFx)
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

Hinweis:

Die Funktionalität der Eingänge LIMP, LIMN, STOP und REF wird automatisch mit der entsprechenden Konfiguration aktiviert.

8.14.5 MC_ControlIO_IFx

Aufgabe:

Programmierung der programmierbaren Ein-und Ausgänge ([MC_ConfigureIO_IFx](#)).

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
	IONumber	INT	Nummer des IOs, der programmiert werden soll: 0: IO 0 1: IO 1 2: IO 2 3: IO 3 Wertebereich: 0..3, Initialwert: 0.
	Index	INT	Index des zu steuernden Parameters. Programmierbarer Eingang: Index des zu schreibenden Parameter. Programmierbarer Ausgang: Index des zu lesenden Parameter. Wertebereich: 0..32767, Initialwert: 0.
	Subindex	INT	Subindex des zu steuernden Parameters. Programmierbarer Eingang: Subindex des zu schreibenden Parameter. Programmierbarer Ausgang: Subindex des zu lesenden Parameter. Wertebereich: 0..32767, Initialwert: 0.
	BitMask	DWORD	Bitmaske für den Parameterwert. Bitmaske zur logischen UND-Verknüpfung mit dem Lesewert vor der Weiterverarbeitung. Sonderfall: bei der Bitmaske 0 bleibt der Lesewert unverändert (wie bei 16#FFFFFFFF). Wertebereich: 0..16#FFFFFFFF, Initialwert: 0.

IN	Switch	INT	Flankenerkennung bzw. Vergleichsoperator. Programmierbarer Eingang: Auswahl der zu erkennenden Flanken 0: keine reaktion auf Pegeländerung 1: Reaktion auf steigende Flanke (Value 1) 2: Reaktion auf fallende Flanke (Value 2) 3: Reaktion auf beide Flanken (Value 1 und Value 2) Programmierbarer Ausgang: Auswahl der Bedingung für Vergleich 0: Lesewert = Vergleichswert (Value 1) 1: Lesewert <> Vergleichswert (Value 1) 2: Lesewert < Vergleichswert (Value 1) 3: Lesewert > Vergleichswert (Value 1) Wertebereich: 0..3, Initialwert: 0.
	Value1	DINT	Schreibwert bei steigender Flanke bzw. Vergleichswert Programmierbarer Eingang: Schreibwert bei steigender Flanke Programmierbarer Ausgang: Vergleichswert für Bedingung Wertebereich: -2147483648..2147483647, Initialwert: 0.
	Value2	DINT	Schreibwert bei fallender Flanke Programmierbarer Eingang: Schreibwert bei fallender Flanke Programmierbarer Ausgang: Keine Bedeutung Wertebereich: -2147483648..2147483647, Initialwert: 0.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

Funktionsweise:

Konfiguration "Eingang programmierbar"

Der Antrieb beobachtet den Eingang *IONo* ständig und führt in Abhängigkeit der Bitmaske und der Flankenerkennung selbständig Parameterzugriffe durch.

Der Parameterzugriff läuft immer nach folgendem Schema ab:

1. Positive bzw. negative Flanke (*Switch*) am Eingang (*IONo*) erkannt.
2. Parameter lesen (*Index, Subindex*).
3. UND-Verknüpfung des Lesewerts mit der Bitmaske (*BitMask*).
4. ODER-Verknüpfung des Ergebnisses mit dem Schreibwert bei positiver (*Value1*) bzw. negativer Flanke (*Value2*).
5. Ergebnis auf Parameter schreiben (*Index, Subindex*).

Konfiguration "Ausgang programmierbar"

Der Antrieb führt zyklisch Lesezugriffe des Parameters durch und setzt in Abhängigkeit der Bitmaske und des Vergleichsoperators den Ausgang *IONo*.

Der Parameterzugriff läuft immer nach folgendem Schema ab:

1. Parameter lesen (*Index, Subindex*)
2. UND-Verknüpfung des Lesewerts mit der Bitmaske (*BitMask*)
3. Ergebnis mit Vergleichswert vergleichen (*Value1*)
4. Je nach Ergebnis Ausgang (*IONo*) HIGH oder LOW setzen (*Switch*)

8.15 Fehlerbehandlung

Zur Fehlerbehandlung hat jeder Baustein einen Ausgang *Error*, der gesetzt wird, wenn ein [synchroner](#) oder [asynchroner](#) Fehler auftritt. Zur genaueren Analyse der Fehlerursache wird der Baustein MC_ReadAxisError_IFx aufgerufen. Mit MC_Reset_IFx wird die Fehlerzelle wieder gelöscht und damit frei für zukünftige Fehlermeldungen.

8.15.1 MC_ReadAxisError_IFx

Aufgabe:

Lesen der [Fehlerinformation](#) eines Antriebs.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Enable	BOOL	FALSE: Ausführung des Baustein beenden. TRUE: Ausführung des Baustein starten.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: Der zu lesende Wert steht zur Verfügung.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.
	ErrorID	INT	0: kein Fehler in der Fehlerzelle >0: Fehlernummer (siehe Liste der Fehlernummern). Wertebereich: 0..32767, Initialwert: 0.

8.15.2 MC_Reset_IFx

Aufgabe:

Fehlerquittierung.

Aufruf:



Parameterbeschreibung:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Ausgangsparameter nach Bausteinende löschen. TRUE: Start der Ausführung bei steigender Flanke.
IN_OUT	Axis	STRUCT	Achs-Referenz [Achs-DB.Achs-Referenz].
	Init	BOOL	Initialisierungsbit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Die Ausführung des Bausteins wurde fehlerfrei beendet.
	Busy	BOOL	TRUE: Der Baustein wird ausgeführt.
	Error	BOOL	TRUE: Während der Ausführung ist ein Fehler aufgetreten.

Funktionsweise:

Die Fehlerzelle wird gelöscht und damit frei für zukünftige Fehlermeldungen, sofern die Fehlerursache beseitigt wurde.

Wurde der Motor durch die automatische Fehlerreaktion gestoppt, dann wird er wieder freigegeben, wenn die Fehlerursache zum Zeitpunkt der Fehlerquittierung beseitigt ist.

Hinweis

Es wird immer nur der zuerst aufgetretene Fehler in die freie Fehlerzelle eingetragen, um Rückschlüsse auf die eigentliche Fehlerursache ziehen zu können. Ältere Fehlernummern werden nicht überschrieben (auch wenn die Fehlerursache bereits behoben ist) und dadurch auch keine neuen Fehler eingetragen, solange die Fehlerzelle belegt ist.

9 Glossar

Anwenderparameter

Parameter, welche vom Anwender verändert werden können und nach Abschalten des Antriebs im Speicher erhalten bleiben. Diese Parameter sind im Handbuch mit **per** gekennzeichnet.

[MC_ResetParameters_IFx](#)

[MC_StoreParameters_IFx](#)

Asynchroner Fehler

Asynchrone Fehler treten vom Programmablauf entkoppelt auf, wie z.B. die Bewegung auf einen Endschalter oder die Übertemperatur des Motors.

[Fehlerbehandlung](#)

Auflösung

Die Positionieraufösungen bezogen auf die Motorabtriebswelle (ohne Getriebe) betragen:

- Beim IFA: 16384 Inkremente/Umdrehung
- Beim IFE: 12 Inkremente/Umdrehung
- Beim IFS 20000 Inkremente/Umdrehung

Bewegungsprofilgenerator

Aus den Parametern für Beschleunigung, Verzögerung, Sollgeschwindigkeit, Soll- und Istposition errechnet der Bewegungsprofilgenerator ein Positions-/Zeitdiagramm, das zu jedem Zeitpunkt der Bewegung die Sollposition des Motors angibt. Dieses Profil wird von der Antriebssteuerung während der Bewegung abgearbeitet.

Fehlerklasse

Das Gerät reagiert in Abhängigkeit von der Schwere des Fehlers:

Klasse	Reaktion	Bedeutung
0	Warnung	Nur Meldung, keine Unterbrechung des Fahrbetriebs.
1	Quick-Stop	Motor stoppt, Endstufe und Regelung bleiben aktiv.
2	Abschalten	Motor stoppt, Endstufe schaltet bei Motorstillstand ab.
3	Fataler Fehler	Endstufe schaltet sofort ab
4	Unkontrollierter Betrieb	Endstufe schaltet sofort ab, Gerät muss abgeschaltet werden.

Fehlerzelle

Die Fehlerzelle enthält den Fehlercode und die Fehlerklasse eines aufgetretenen Fehlers. Ein neu aufgetretener Fehler wird eingetragen wenn die Fehlerzelle frei war (d.h. gleich Null). Ist die Fehlerzelle belegt (d.h. ungleich Null), dann wird die frühere Fehlermeldung nicht überschrieben, sondern die neue Fehlermeldung wird ignoriert. Die Fehlerzelle wird durch die Ausführung des Baustein MC_Reset_IFx gelöscht, sofern die Fehlerursache behoben ist.

Geräteparameterliste oder Objektliste

Verzeichnis aller Parameter im Gerät, auf die schreibend oder lesend zugegriffen werden kann. Die Parameter sind in der Gerätedokumentation beschrieben.

[MC_WriteParameter_IFx](#)

[MC_ReadParameter_IFx](#)

Inc, Inc/s

bedeutet "Inkrement" bzw. "Inkrement pro Sekunde" (engl. increments).

Bezogen auf den Motor bedeutet es die Auflösung der Endstufe, mit der der Motor positioniert werden kann (ohne Berücksichtigung vorgebauter Getriebe):

Auflösung der Antriebe:

- IFA: 16384 Inkremente/Umdrehung
- IFE: 12 Inkremente/Umdrehung
- IFS: 20000 Inkremente/Umdrehung

Die Geschwindigkeit ergibt sich aus der Anzahl Inkremente pro Sekunde [Inc/s].

Synchroner Fehler

Synchrone Fehler treten beim Schreiben von Parametern oder Starten von Funktionen auf und sind an eine Aktion gekoppelt, wie z.B. das Schreiben eines unzulässigen Parameterwerts oder das Starten einer Bewegung bei ausgeschaltetem Motorstrom.

[Fehlerbehandlung](#)

10 Liste der Fehlernummern

ErrorID hex	ErrorID dez	Fehler-klasse	Antriebsfehlermeldungen
0100h	256	2	Unterspannung 1 Leistungsversorgung
0101h	257	3	Unterspannung 2 Leistungsversorgung
0102h	258	3	Überspannung Leistungsversorgung
0105h	261	3	Überlast Motor
010Ch	268	2	Übertemperatur Endstufe
0110h	272	3	Motor blockiert oder ausgerastet
0111h	273	3	Schleppfehler
0112h	274	4	Motor-Lagesensor defekt
0115h	277	1	Protokollfehler Feldbus
0116h	278	2	Feldbus: Nodeguarding/Watchdog oder Clear
0117h	279	3	Frequenz am Puls-/Richtungseingang zu hoch
0118h	280	3	Kurzschluss dig. Ausgänge
0119h	281	3	Sicherheitsfunktion "Power Removal" ausgelöst (PWRR_A und PWRR_B)
011Ah	282	4	PWRR_A und PWRR_B haben >1s unterschiedliche Pegel
011Ch	284	4	Hardwarefehler EEPROM
011Dh	285	4	Hochlauf-Fehler
011Eh	286	4	Interner Systemfehler
011Fh	287	4	Watchdog
0120h	288	0	Warnung Positionsüberlauf Profilgenerator
0121h	289	0	Warnung Übertemperatur IGBTs
0128h	296	0	Warnung E/A-Timing
0130h	304	0	Parameter existiert nicht, ungültiger Index
0131h	305	0	Parameter existiert nicht, ungültiger Subindex
0132h	306	0	Kommunikationsprotokoll: unbekannter Dienst
0133h	307	0	Schreiben des Parameters nicht zulässig
0134h	308	0	Parameterwert außerhalb zulässigem Wertebereich
0135h	309	0	Segmentdienst nicht initialisiert
0136h	310	0	Fehler bei Aufzeichnungsfunktion
0137h	311	0	Zustand nicht Operation Enable
0138h	312	0	Bearbeitung in aktuellem Betriebszustand der Zustandsmaschine nicht möglich
0139h	313	0	Sollpositionsgenerierung unterbrochen
013Ah	314	0	Umschaltung bei laufender Achsbetriebsart nicht möglich
013Bh	315	0	Kommando bei laufender Bearbeitung nicht zulässig (xxxx_end=0)
013Ch	316	0	Fehler im Auswahlparameter
013Dh	317	0	Positionsüberlauf vorhanden/aufgetreten
013Eh	318	0	Istposition ist noch nicht definiert
013Fh	319	4	EEPROM nicht initialisiert
0140h	320	4	EEPROM nicht kompatibel zur akt. Software
0141h	321	4	Lesefehler EEPROM
0142h	322	4	Schreibfehler EEPROM
0143h	323	4	Prüfsummenfehler im EEPROM
0144h	324	0	Nicht berechenbarer Wert

0145h	325	0	Funktion nur im Stillstand erlaubt
0146h	326	0	Referenzfahrt ist aktiv
0147h	327	0	Kommando bei laufender Bearbeitung nicht zulässig (xxx_end=0)
0148h	328	1	RS485-Schnittstelle: Overrun-Fehler
0149h	329	1	RS485-Schnittstelle: Framing-Fehler
014Ah	330	1	RS485-Schnittstelle: Parity-Fehler
014Bh	331	1	RS485-Schnittstelle: Empfangsfehler
014Ch	332	1	RS485-Schnittstelle: Puffer-Ueberlauf
014Dh	333	1	RS485-Schnittstelle: Protokollfehler
014Eh	334	1	Nodeguarding, Schnittstelle wird nicht mehr bedient
014Fh	335	0	Zustand "Quick Stop" aktiviert
0150h	336	1	Unzulässiger Endschalter ist aktiv
0151h	337	1	Schalter wurde überfahren, Ausfahren nicht möglich
0152h	338	1	Schaltkante innerhalb Ausfahrweg nicht gefunden
0153h	339	1	Indexpuls nicht gefunden
0154h	340	1	Reproduzierbarkeit der Indexpulsfahrt unsicher, Indexpuls ist zu nahe an Schalter
0155h	341	1	Schalter nach Ausfahren noch immer aktiv, Ursache evtl. Prellen des Schalters
0156h	342	1	Eingang ist nicht als LIMP/LIMN/REF parametrier
0157h	343	1	Unterbrechung / "Quick Stop" durch LIMP
0158h	344	1	Unterbrechung / "Quick Stop" durch LIMN
0159h	345	1	Unterbrechung / "Quick Stop" durch REF
015Ah	346	1	Unterbrechung / "Quick Stop" durch STOP
015Bh	347	1	Endschalter ist nicht freigegeben
015Ch	348	0	Bearbeitung in aktueller Achsbetriebsart nicht erlaubt
015Dh	349	0	Parameter bei diesem Gerät nicht verfügbar
015Eh	350	0	Funktion bei diesem Gerät nicht verfügbar
015Fh	351	0	Zugriff verweigert
0160h	352	4	Fertigungsdaten im EEPROM nicht kompatibel zu akt. Software
0161h	353	4	Indexpuls-Sensor nicht abgeglichen
0162h	354	0	Antrieb ist nicht referenziert
0163h	355	0	CAN-Schnittstelle: COB-ID nicht korrekt
0164h	356	0	CAN-Schnittstelle: Anfrage fehlerhaft
0165h	357	0	CAN-Schnittstelle: Overrun-Fehler
0166h	358	0	CAN-Schnittstelle: Telegramm konnte nicht gespeichert werden
0167h	359	0	CAN-Schnittstelle: allgemeiner Fehler CAN Stack
0168h	360	0	Feldbus: Datentyp und Parameterlänge stimmen nicht überein
0169h	361	0	Blockiererkennung ist ausgeschaltet
016Ah	362	0	Verbindungsaufnahme zum DSP Bootloader fehlgeschlagen
016Bh	363	0	Kommunikation mit DSP Bootloader fehlerhaft
016Ch	364	0	Fehler bei der Speicherinitialisierung des SPC3
016Dh	365	0	Fehler bei der Berechnung der Länge der Input/Output-Daten
016Eh	366	0	Eingestellte Profibusadresse ist außerhalb vom erlaubten Bereich
016Fh	367	0	Unerlaubte Verwendung des DIP-Schalters S1.1
0170h	368	0	DSP Software nicht mit Profibus Software kompatibel
0171h	369	0	Prüfsumme der Profibus-DP Schnittstellensoftware nicht korrekt
0172h	370	0	Oszilloskop-Funktion: keine weiteren Daten verfügbar

Step 7 Motion Bibliothek für IFS, IFE und IFA

0173h	371	0	Oszilloskop-Funktion: Triggervariable wurde nicht definiert
0174h	372	0	Oszilloskop-Funktion unvollständig parametrisiert
0175h	373	1	Interne Kommunikation
0177h	375	1	Unterbrechung / "Quick Stop" durch Softwareendschalter positive Drehrichtung
0178h	376	1	Unterbrechung / "Quick Stop" durch Softwareendschalter negative Drehrichtung
Bibliotheksfehlermeldungen			
16#FF00	65280		Wrong_AxisRef
16#FF01	65281		Initialization_Failed
16#FF02	65282		Wrong_Data_Length
16#FF03	65283		Timeout
16#FF04	65284		Axis_Busy
16#FF05	65285		Invalid_Parameter_Number
16#FF06	65286		Unknown_State
16#FF07	65287		Capture_Busy
16#FF08	65288		Trigger_Event_Lost
16#FF09	65289		Axis_Not_In_Standstill
16#FF0A	65290		Unknown_Device_Type
16#FF0B	65291		Wrong_Data_Struct
16#FF20	65312		C/S specifier not valid or unknown
16#FF21	65313		attempt to write a read only object
16#FF22	65314		object does not exist in the object dictionary
16#FF23	65315		data type does not match, length of service parameter does not match
16#FF24	65316		sub-index does not exist
16#FF25	65317		value range of parameter exceeded (only for write access)
16#FFFF	65535		Unknown_Error

11 Parameterliste für UP- und Downloadfunktion

[MC UploadParameter IFx](#)

[MC DownloadParameter IFx](#)

Parametername	Profibus Adresse	Antrieb
timeout	1:11	IFA, IFE, IFS
name1	11:1	IFA, IFE, IFS
name2	11:2	IFA, IFE, IFS
I_still	14:1	IFS
I_acc	14:2	IFS
I_const	14:3	IFS
I_stop	14:4	IFS
monitorM	14:7	IFS
I_max	15:3	IFA, IFE
I_maxStop	15:4	IFA
I_maxBlk	15:5	IFE
p_maxDiff	15:7	IFE
KPn	15:8	IFA, IFE
TNn	15:9	IFA, IFE
KPp	15:10	IFA, IFE
KFPp	15:11	IFA, IFE
T_block	15:12	IFE
p_win	15:15	IFA
p_winTime	15:16	IFA
p_MaxDif2	15:17	IFA
pscDamp	15:20	IFA
pscDelay	15:21	IFA
serBaud	22:1	IFA, IFE, IFS
serAdr	22:2	IFA, IFE, IFS
serFormat	22:3	IFA, IFE, IFS
SafeState	24:5	IFA, IFE, IFS
invertDir	28:6	IFA, IFE, IFS
WarnOvrn	28:11	IFA, IFE, IFS
SignEnabl	28:13	IFA, IFE, IFS
SignLevel	28:14	IFA, IFE, IFS
dec_Stop	28:21	IFA, IFE, IFS
Flt_pDif	28:24	IFA, IFE
v_target0	29:23	IFA, IFE, IFS
acc	29:26	IFA, IFE, IFS
IO0_def	34:1	IFA, IFE, IFS
IO1_def	34:2	IFA, IFE, IFS
IO2_def	34:3	IFA, IFE, IFS
IO3_def	34:4	IFA, IFE, IFS
progDelay	34:7	IFA, IFE, IFS

IO0_Index	800:1	IFA, IFE, IFS
IO0_Subindex	800:2	IFA, IFE, IFS
IO0_BitMask	800:3	IFA, IFE, IFS
IO0_Switch	800:4	IFA, IFE, IFS
IO0_Value1	800:5	IFA, IFE, IFS
IO0_Value2	800:6	IFA, IFE, IFS
IO1_Index	801:1	IFA, IFE, IFS
IO1_Subindex	801:2	IFA, IFE, IFS
IO1_BitMask	801:3	IFA, IFE, IFS
IO1_Switch	801:4	IFA, IFE, IFS
IO1_Value1	801:5	IFA, IFE, IFS
IO1_Value2	801:6	IFA, IFE, IFS
IO2_Index	802:1	IFA, IFE, IFS
IO2_Subindex	802:2	IFA, IFE, IFS
IO2_BitMask	802:3	IFA, IFE, IFS
IO2_Switch	802:4	IFA, IFE, IFS
IO2_Value1	802:5	IFA, IFE, IFS
IO2_Value2	802:6	IFA, IFE, IFS
IO3_Index	803:1	IFA, IFE, IFS
IO3_Subindex	803:2	IFA, IFE, IFS
IO3_BitMask	803:3	IFA, IFE, IFS
IO3_Switch	803:4	IFA, IFE, IFS
IO3_Value1	803:5	IFA, IFE, IFS
IO3_Value2	803:6	IFA, IFE, IFS